

Oracle® E-Business Suite Concepts

Patching and Management Tools

Introduction

Powerful yet simple-to-use management tools are critical to the usefulness of any applications offering, and one fundamental management need is the ability to apply patches quickly and easily. This chapter surveys the relevant tools provided with Oracle E-Business Suite.

The most important new feature in Oracle E-Business Suite Release 12.2 is the ability to patch a running system, instead of having to take the system down for a significant period of time while the patches are applied. An outline of patching principles and nomenclature is followed by a description of the online patching environment and its associated tools.

This discussion of patching is complemented by the second part of the chapter, which examines the main features of the *Oracle Applications Manager* tool.

Patching Principles and Nomenclature

Patching Principles

As described in more detail later in this chapter, all Oracle E-Business Suite Release 12.2 patches are applied online: that is, to a running system with users logged on and working in their normal way. Use of online patching is mandatory in Release 12.2.

Note: For full details of the commands and options used in online patching, see the Patching section of *Oracle E-Business Suite Maintenance Guide*.

Patch Nomenclature

Oracle E-Business Suite Release 12.2 includes numerous products that are designed to serve a wide variety of needs in a consistent and complementary way. All have abbreviated names or two, or more usually three, letters. These short names are used in special patch names (see below) and other places. In general, the short names relate to the full product name. Common examples include Applications Technology (ATG), Financials (FIN), Human Resources (HR), and Supply Chain Management (SCM).

You can search a list of product abbreviations and full names within Oracle Applications Manager, using the *License Manager* tool. From the License Manager home page, choose Reports > Licensed Products. From the List of Products, you can view details of licensed, shared, or not licensed products, or search for a particular product abbreviation or name.

Additional Information: To learn more about the License Manager tool, see *Oracle E-Business Suite Maintenance Guide*.

Common Oracle E-Business Suite patching terms include the following.

Note: The specific examples shown in this section are intended solely to illustrate the principles involved, and do not represent current release levels.

- A *release update pack* increments the third section of the release number, for example to 12.2.1 or 12.2.2, and consists of a cumulative aggregation of product family release update packs created across Oracle E-Business Suite since the initial (base) release.
- A *product family release update pack* is an aggregation of patches created for a specific product family after the initial release, and designated by the release number followed by the product short name and a letter. For example, in 'R12.ATG_PFB.DELTA.3', '12' is the major release number, 'ATG_PFB' means 'Applications Technology Product Family pack B', and 'Delta.3' means this is the third *rollup patch* on top of Applications Technology Product Family pack B.
- A *codelevel* is essentially a rollup patch with a name that looks like a version number. Higher codelevels include all fixes from lower codelevels, plus additional fixes. For example, R12.AD.B was the second release of AD in Release 12. R12.AD.B.1 was a higher codelevel, which included all code in R12.AD.B plus more fixes; R12.AD.B.2 was a higher codelevel again, which included all code in R12.AD.B.1, plus further fixes.

Online Patching

Traditionally, Oracle E-Business Suite has needed to be shut down to apply patches and bug fixes of various kinds. Scheduling the sometimes lengthy downtimes needed can prove difficult to combine with requirements for the highest possible availability of business-critical systems. This is exacerbated by the increasing move to consolidate systems into a single instance that may support worldwide operations, and which consequently does not have a natural downtime window.

The introduction of *online patching* in Oracle E-Business Suite Release 12.2 greatly reduces the need for such downtime. All Oracle E-Business Suite patches in Release 12.2 are applied online.

Key features of online patching include:

- Oracle E-Business Suite patching operations are carried out while the applications are in use and users are online. This is accomplished by means of a copy of the application components that are stored in the database and file system.
- Oracle E-Business Suite patching is performed using the new *adtop* (AD Online Patching) utility.
- A short period of downtime is required, but this amounts to little more than a restart of the application tier services: the time the applications are unavailable is measured in minutes rather than hours, and this can be set to be at the most convenient time, for example outside normal business hours.
- Maintenance Mode is not needed or used in Release 12.2.

The patching model used before Release 12.2 was designed to minimize downtime by performing its operations in the shortest time possible, using whatever resources are needed. In contrast, the online patching model is designed to minimize downtime by allowing patching operations to be performed while users remain on the system.

Overview

In traditional patching, application of a patch is a single logical operation. In online patching, it can be thought of as a series of related steps:

1. A copy is made of the code of the running system that resides in both the file system and database.
2. Patches are applied to the copy while users continue to access the running system.
3. The copy becomes the new running system.
4. The original running system (now obsolete) is recycled for use in future patching cycles.

These steps constitute the *online patching cycle*. To implement this mechanism, various changes have been made to the Oracle E-Business Suite infrastructure.

Any method used to create a copy of the running application must take into account that an Oracle E-Business Suite application comprises both code and data, stored in the database and the file system. A key concept is the *edition* as a copy of the application code: the running application executes on the *run edition*, while any patching activity takes place in the *patch edition*. The implementation mechanism uses the Oracle Database *Edition-Based Redefinition* (EBR) feature to cater for database objects, and a dual file system to cater for objects stored in the file system.

Two complete file systems are always present in an Oracle E-Business Suite Release 12.2 system. As noted above, the run file system is the one currently in use by the running application, while the patch file system is the either being patched or awaiting the start of the next online patching cycle. In other words, the two file systems swap roles with every online patching cycle.

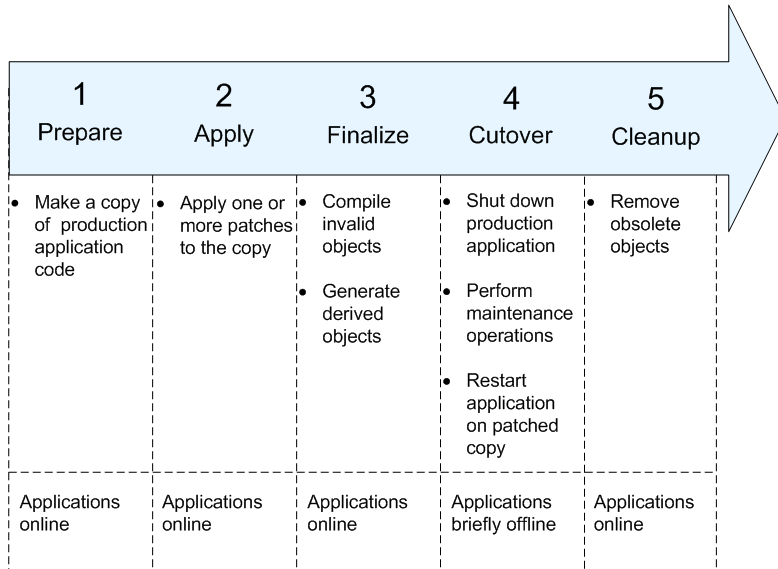
In the database, there is also a run edition and patch edition, but they do not swap roles back and forth as in the file system: the patch edition only comes into existence during a patching cycle, and becomes the new run edition at the end of the cycle. The former database run edition (the old edition) and the obsolete objects it contains are discarded at the end of an online patching cycle, and the space reclaimed during the cleanup phase.

These concepts will all be described in more detail after a discussion of the way they are used in the online patching cycle.

The Online Patching Cycle

The online patching cycle is divided into five phases:

Online Patching Phases



Patching Cycle Overview

The key actions in the various stages of the online patching cycle can be summarized as follows:

- Prepare**
 - Synchronizes patch edition and run edition on the file system.
 - Creates a new patch edition in the database.
- Apply**
 - Executes patch drivers to update patch edition.
 - Patches applied: can be one or many, including customizations.
- Finalize**
 - Compiles invalid objects.
 - Generates derived objects.
- Cutover**
 - Configures patch edition file system to be the new run edition file system.
 - Configures patch edition of database to be the new run edition.
 - Restarts application tier services.
- Cleanup**
 - Delete obsolete code and seed data to recover space.

Each of the phases will now be discussed in more detail.

Prepare

The following actions are taken in this phase.

On the *file system*:

- The patch file system is synchronized with the run file system.
- The patch edition files become an exact copy of the run edition files.

Note: By default, synchronization is *incremental*: that is, only files that were changed in the last patch application are copied.

In the *database*:

- A patch edition is created in the database.
- All code objects in the patch edition begin as pointers to code objects in the run edition. Code objects in the patch edition begin as lightweight "stub objects" that point to the actual object definitions, which are inherited from earlier editions. Stub objects consume minimal space, so the database patch edition is initially very small in size.
- As patches are applied to the patch edition, code objects are *actualized* (have a new definition created) in that edition.

Note: Storage objects (such as transaction tables) are *not* copied.

Apply

The following actions are taken in this phase:

1. Patches are applied to the patch edition. During this process, any changed stub objects will become actual code objects in the patch edition.

2. The changes introduced by the patches are made in the isolation of the patch edition.

- Changes to *code* objects are only visible in the patch edition of the file system and database.
- Changes to *tables* are stored in new columns or rows that are only visible to the patch edition.

Note: At this point, users still remain connected to the application and performing their work.

Finalize

This phase is used to perform the final operations that can be executed while the application is online:

1. Invalid objects are compiled.
2. Derived objects are generated.
3. Any actions that must be performed at cutover are pre-computed and stored for quick execution at that time.

Cutover

This phase involves:

1. Shutdown of application tier services.
2. Execution of any required cutover actions to maintain non-editioned objects and data.
3. Configuration of the patch edition of the file system as the new run edition.
4. Configuration of the patch edition of the database as the new run edition.
5. Restart of application tier services on the new run edition.

Although the cutover phase does require a short period of application tier services downtime, the online patching cycle can be paused for as long as required prior to running this phase. You could, for example, add such a pause to ensure that the downtime period will be outside business hours.

Note: The database remains open throughout the entire online patching cycle, including cutover.

Cleanup

The following database actions are taken in this phase, which occurs after users have been brought back online to the newly-patched application:

1. Old code objects that are no longer visible in the run edition are dropped.
2. Old data (rows or columns) that is no longer visible in the run edition is deleted or dropped.
3. Old database editions that no longer contain actual objects are dropped.

Database Implementation

Creating a copy of the database part of the running system has been accomplished by taking advantage of the Oracle Database *Edition-Based Redefinition* (EBR) feature. This allows an application to efficiently store multiple copies of its application definition in the same database, and thereby enables online upgrade of the database tier.

The term *edition* refers to the isolation mechanism that allows pre-upgrade and post-upgrade schemas to co-exist. The simplest way to think of an edition is as a separate (isolated) copy of all database code objects that are changed by a patch.

The three types of database edition are:

- **Run:** Online users connect to this edition, which is always the default database edition. The run edition will always exist.
- **Patch :** Patching tools connect to this edition. A child of the run edition, the patch edition exists *only* while a patching cycle is in progress.
- **Old:** When a patch edition is promoted to be the run edition, the previous run edition is now regarded as an old edition. There may be zero or more old editions at a given time. They are discarded when a full cleanup (described later) is performed. You cannot connect to an old edition.

File System Implementation

Creating and using two file systems allows one (*run file system*) to be part of the running system, while the other (*patch file system*) is either being patched or waiting to be patched during the next patching cycle.

Note: The file systems are designated *File System 1* and *File System 2* in Rapid Install. These are abbreviated to *fs1* and *fs2* respectively.

The two file systems are sometimes referred to as a *dual file system*, and swap roles at the end of each patching cycle. That is, the file system that has just been patched is put into use as part of the running system (becoming the new run file system), and the previous run file system takes over the the role of patch file system (in readiness for commencement of the next patching cycle).

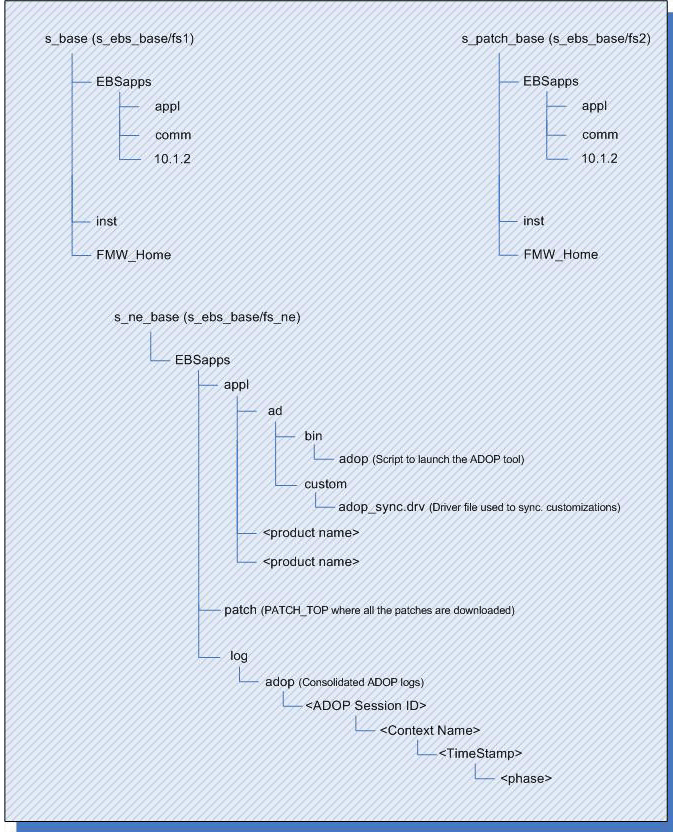
Important: The existence of the dual file system has significant implications for general (non-patching) maintenance activities. For information on understanding how to decide which file system to run AD tools from, refer to: *Choosing the Correct File System For Maintenance Tasks, Oracle E-Business Suite Maintenance Guide* (Chapter 7, page 7-1).

However, two file systems are not sufficient to meet all the practical needs of an online patching environment for Oracle E-Business Suite. A third file system, described in the next section, is also required.

Non-Editioned File System

The dual file system approach caters for application code, but applications also use the file system to read and write business data. In Release 12.2, application data files are stored in a third area, the *non-editioned file system* (*fs_ne*), which is used to store data that is needed across all file systems. Non-editioned files are not copied or moved during patching: their location remains constant across online patching cycles.

The Three File Systems: fs1, fs2, and fs_ne



The non-editioned file system is therefore completely separate from file system 1 and file system 2.

Context Variables for Online Patching File Systems

Several AutoConfig context variables support the file systems used in online patching. For example, the `s_ne_base` context variable stores the location of the non-editioned file system, and AutoConfig sets the environment variable `$NE_BASE` to the corresponding value. AutoConfig also sets the environment variables `$RUN_BASE` and `$PATCH_BASE`, so you can easily tell which is currently the run file system and which the patch file system.

Files Stored in the Non-Edited File System

The non-editioned file system is designed to store files that contain transactional data and reference data. Examples include: import, export files, general log files, and concurrent manager log and out files. These are all examples of files that are not modified during an online patching cycle.

More specialized examples include:

- Batch upload and download files.
- Files used to transfer transactional data from processes external to Oracle E-Business Suite (for example, where a third party order entry system delivers orders via order import files).
- Files containing transactions that are needed across all file systems.

Note: The non-editioned file system is not designed to store shared files, because initially identical files can become non-identical during patching life cycles. Nor is it designed to store code, which is editioned (one copy can be in the run file system while the other is in the patch file system).

Concurrent Processing and the Non-Edited File System

As noted above, concurrent processing in particular makes significant use of the non-editioned file system, with all concurrent manager log and out files being stored there. This has some additional implications, as described below.

Before Release 12.2, there were two choices for the location of log and out files. The first was `$APPL_TOP`, and the second was `$APPLCSF` (which pointed to `$APPL_TOP/admin/$TWO_TASK`). The location used was determined by whether the `$APPLCSF` environment variable was set.

With Release 12.2, the `$APPL_TOP` contains only code and configuration files. Therefore, it is mandatory for the `$APPLCSF` environment variable to be set. Its default value points to the non-editioned file system, specifically to `<s_ne_base>/inst/<EBS_Instance_ID>/logs/appl/conc`: for example, `/u01/R122_EBS/fs_ne/inst/VIS_app101/logs/appl/conc`.

In addition, a new environment variable, `$APPLDLM`, has been introduced to provide the option of organizing storage of log and out files on a product-specific basis.

In Release 12.2, `$APPLDLM` environment variable can have multiple possible values. The two main ones are 'single' and 'product'.

If `$APPLDLM` is set to 'single', concurrent processing and AD Administration log files will both be stored under `$APPLCSF/log`. Concurrent processing out files will be stored under `$APPLCSF/out`. This is the default setting.

If `$APPLDLM` is set to 'product', the concurrent processing log and out files will be written to product-specific directories under `$APPLCSF`. This is in contrast to previous releases, where the directories were under `$APPL_TOP`. For example, concurrent manager log files will be located under `$APPLCSF/<product>/log`, AD Administration log files will be located under `$APPLCSF/log` (i.e. the same location as with the 'single' setting), and PO reports will be located under `$APPLCSF/po/out`.

Beginning with Release 12.2.6, Oracle E-Business Suite provides additional storage strategies for management of large numbers of concurrent processing log and output files. For example, concurrent request log and out files can be organized by date or by user name. These storage strategies are called *schemes*. For more information, refer to the *Oracle E-Business Suite Setup Guide*.

In summary, the default behavior has been designed to be exactly the same as in previous releases: that is to say, if you did not set `$APPLCSF` before, and you do not set it or `$APPLDLM` now, you will not notice any difference in concurrent processing log file management or storage.

Edited and Non-Edited Objects

EBR automatically manages versioning of objects that support editioning, such as code objects. However, not all objects can be editioned: most notably, this includes transactional data, of which there is only ever one copy.

Oracle E-Business Suite Release 12.2 introduces a new concept, the *logical view* of the data model. This is implemented via *synonyms* and *editioning views*, which isolate the running application from changes to the data model that may be introduced by a patch. That is to say, data model changes are guaranteed to not affect the running application. Data model changes are implemented as new columns on the table, which are exposed to the patch edition of the application via the editioning view. New tables can be introduced by an extension of the same principle.

Crossedition Triggers

A new type of object, the *crossedition trigger*, is used to synchronize data as part of the online patching process. These triggers provide the logic to synchronize and transform data between the run edition and patch edition storage columns. The result is that new transactions entered into the system are patched in place.

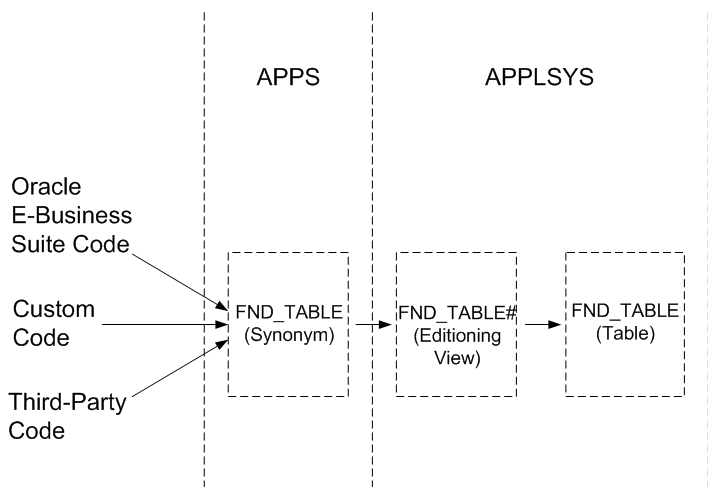
More specifically, crossedition triggers allow the run edition to signal that a data update is required. They fire in response to an insert into, delete from, or update of, FND_TABLE. For example, suppose that a patch updates a column called DESCRIPTION from upper case to mixed case. Editioning views project different views of the table to the run and patch editions, so the running application still sees the column data as upper case, while the patched application sees the column data as mixed case. The updated column is maintained by a crossedition trigger.

In summary, crossedition triggers are used to upgrade both existing data and ongoing changes that occur while the run edition remains in use.

Editing Views and the Application Data Model

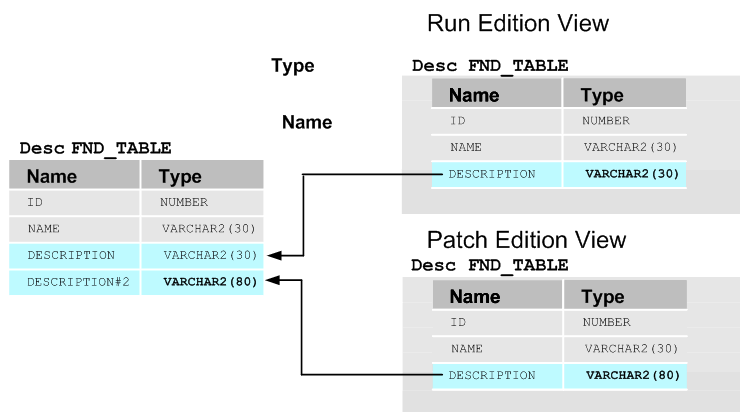
Editing views can be thought of as providing a *cover layer*, or logical representation of the data, on top of the physical representation. All code (Oracle E-Business Suite, custom, or third-party) must access Oracle E-Business Suite data via the cover layer: accessing the data model via the physical layer may result in obsolete data being returned.

Logical View of the Data Model



Note: The implementation of EBR for Oracle E-Business Suite requires all code to access the data model via the APPS synonym, which points to the editioning view (logical model).

Database Editioning Views



Important: Any code accessing the physical model risks accessing obsolete columns.

Preparing for Online Patching

Important: Upgrading to Oracle E-Business Suite Release 12.2 is *not* done with online patching enabled. Online patching can only be used after you complete the upgrade to Oracle E-Business Suite Release 12.2.

The Release 12.2 upgrade process installs a logical view of the data model that all code must use to access Oracle E-Business Suite, and automatically upgrades seed data tables to support editioned data storage. Several steps can be taken to ease the transition to a Release 12.2 online patching environment. These are divided into pre-upgrade and post-upgrade categories, and mentioned briefly below.

Pre-Upgrade

Ahead to the actual upgrade to Oracle E-Business Suite Release 12.2, it is highly advisable to prepare for the move to an online patching environment by running the three *online patching analysis reports*. Available for all supported pre-12.2 releases, these highlight all cases in Oracle E-Business Suite code or custom code where a non-editioned object references an editioned object.

The reports are:

- **Summary Report:** ADZDPSUM.sql provides the following information:
 1. Schemas that *will* be editioned upon online patching enablement.
 2. Schemas that *will not* be editioned upon online patching enablement.
 3. Unregistered schemas that depend on Oracle E-Business Suite objects.
- **Manual Fixes Report:** ADZDPMAN.sql lists any manual fixes required.
- **Auto Fixes Report:** ADZDPAUT.sql lists any automatic fixes required.

The reports are run as follows:

```
sqlplus system/manager@DB @$AD_TOP/sql/ADZDPSUM.sql sqlplus system/manager@DB @$AD_TOP/sql/ADZDPMAN.sql sqlplus system/manager@DB @$AD_TOP/sql/ADZDPAUT.sql
```

Note: For further details of the online patching analysis reports, see *Enabling Online Patching* in the *Oracle E-Business Suite Online Patching* chapter of *Oracle E-Business Suite Upgrade Guide: Release 12.0 and 12.2*.

Any compliance issues in Oracle E-Business Suite code are automatically fixed by the upgrade to Release 12.2. With regard to custom code, Oracle recommends running and studying the report before commencing the upgrade, to determine the amount of effort that will be needed for compliance. Most non-compliance issues in custom code can be fixed in the 12.2 upgrade simply by registering the custom schema with Oracle E-Business Suite. Only a small number of issues may require manual intervention.

Post-Upgrade

Having completed the actual upgrade, it is important to bear in mind several key aspects of the online patching environment, including identification of suitable new coding standards to ensure compliance of any new customizations. Also, patch editions should not be treated as a testing environment: patches should still be tested on a test system rather than the production system. New test scenarios may need to be considered, given that users can now be online during patch application.

For full details of the upgrade process, including online patching analysis reports and other pre-upgrade checks and reports, see the applicable upgrade guide: either *Oracle E-Business Suite Upgrade Guide Release 11i to 12.2* or *Oracle E-Business Suite Upgrade Guide Release 12.0 and 12.1 to 12.2*.

Online Patching Tools

Patching is performed by running the `adop` (AD online patching) utility. You run `adop` instead of the `adpatch` utility that was used in releases prior to 12.2.

The `adop` tool orchestrates the entire patching cycle, and can be executed by specifying phases individually or collectively. For example, `adop phase=prepare`, `adop phase=apply`, and `adop phase=cutover` is the equivalent of `adop phase=prepare,apply,cutover`. In both of these examples, only a single session id would be created.

Note: For full details of `adop` features and options, refer to the Patching part of *Oracle E-Business Suite Maintenance Guide*.

Hotpatch Mode

The *hotpatch* mode of online patching provides the capability to apply changes to the run edition while the run edition is in active use. However, this option can only be used to apply patches that provide explicit support for it. The readme of a patch will state if this is the case.

If hotpatch mode is used with patches that do not support it, the following issues may occur.

- **Possible runtime transaction failure because of invalid objects** - Although some objects may automatically revalidate, a hotpatch operation may not yet have finished making all the necessary object changes. This will result in associated user transactions continuing to fail until the hotpatch is complete.
- **Possible runtime transaction failure due to loss of PL/SQL package state** - Many PL/SQL packages contain global variables which the database considers to be "package state". If a database session has accessed a package containing package state at any time before a hotpatch operation, and then accesses the same package after the package has been invalidated, the session will get the error "ORA-04068: existing state of packages has been discarded" and the transaction will fail.
- **Application code and database objects may be temporarily inconsistent** - If a patch loads new seed data, each uploaded change will be visible to the running application as it is committed. However, while the seed data is being loaded, only some of the necessary changes may have been committed at any given point. This may lead to data integrity problems if the seed data is accessed during the loading operation.
- **Tables that are patched will be temporarily inconsistent** - The process of patching a table involves creating new columns, regenerating the editioning view, populating new columns via cross-edition triggers, and recreating indexes to reference the new columns. Until this sequence of operations is complete, data in the table may appear to be missing or incorrect, or not have correct indexing. Full execution of the table patching steps are spread out over several `adop` phases, so the table may be in an inconsistent state for the entire duration of the hotpatch operation. It is therefore not safe to access any table from the edition that is being patched.
- **Code and data cached in application tier server memory may be inconsistent with changes made by the hotpatch operation** - To provide optimum performance, application tier servers load code objects, seed data and transaction data into memory caches. After the hotpatch operation is complete, and until the application tier servers are restarted, this cached information may be out of date and lead to various types of failure or corruption.
- **Runtime processing may hold long-term locks on objects, causing the hotpatch operation to fail** - Some database objects in active use by the running application cannot be changed by a hotpatch operation without an exclusive lock first having been obtained on the object (code or data). Such runtime processing includes long-running concurrent programs, or background services that may hold locks on code objects indefinitely. Consequently, the hotpatch operation may fail because it cannot obtain the exclusive lock required to apply some change.
- **Other reasons**
 - In addition to the above issues, it is also possible that a patch will not work in hotpatch mode for unspecified reasons.

In general, Oracle E-Business Suite Development does not test patches in hotpatch mode. Any failures, corruptions, or other errors that result from unauthorized use of hotpatch mode cannot be supported. Consequently, you should only apply a patch in hotpatch mode if support for doing so is explicitly stated in the patch readme.

Online Patching and Seed Data

Oracle E-Business Suite *seed data* is data stored in database tables that affects the behavior of the applications, and is patched as needed by Oracle E-Business Suite Development. In an online patch environment, patches cannot be allowed to modify the seed data that is visible to the running application.

Online patches are prevented from modifying runtime seed data by the use of *editioned data storage*. This involves creation of a (patch) copy of the seed data, which is stored in the same table. The patches that are applied interact only with this copy, while the run edition only interacts with a private copy (which is eventually deleted as part of the cleanup phase).

The running application uses the run edition copy of seed data, while patches may update the patch edition copy of seed data in isolation. The two copies are isolated, except that seed data changes made by the running application are synchronized to the patch edition copy.

In terms of editions and seed data, the *run edition*:

- Always operates on a private copy of the seed data
- Is never modified by patch application

In contrast, the *patch edition*:

- Runs the seed data loader
- Prepares the relevant table for patching
- Copies all table rows
- Loads seed data changes into the (patch) copy

Updates to seed data in the run edition are automatically propagated to the patch edition by the use of crossedition triggers.

Environment Variables

The following table shows examples of the environment variables that are used in online patching. These are all set automatically.

Environment Variable	Example Value on Run File System	Example Value on Patch File System
FILE_EDITION	run	patch
TWO_TASK	vision	vision_patch

APPS_JDBC_URL	jdbc:oracle:thin:@(DESCRIPTION= (ADDRESS_LIST= (LOAD_BALANCE=YES) (FAILOVER=YES) (ADDRESS=PROTOCOL=tcp) (HOST=testsys1.example.com (PORT=1521))) (CONNECT_DATA= (SERVICE_NAME=vision)))	jdbc:oracle:thin:@(DESCRIPTION= (ADDRESS_LIST= (LOAD_BALANCE=YES) (FAILOVER=YES) (ADDRESS= (PROTOCOL=tcp) (HOST=testsys1.example.com) (PORT=1521))) (CONNECT_DATA= (SERVICE_NAME=ebs_patch) (INSTANCE_NAME=vision)))
AD_APPS_JDBC_URL	(DESCRIPTION=(ADDRESS_LIST= (LOAD_BALANCE=YES) (FAILOVER=YES) (ADDRESS=(PROTOCOL=tcp) (HOST=testsys1.example.com) (PORT=1521))) (CONNECT_DATA= (SERVICE_NAME=vision)))	(DESCRIPTION=(ADDRESS_LIST= (LOAD_BALANCE=YES) (FAILOVER=YES) (ADDRESS=(PROTOCOL=tcp) (HOST=testsys1.example.com) (PORT=1521))) (CONNECT_DATA= (SERVICE_NAME=ebs_patch) (INSTANCE_NAME=vision)))
RUN_BASE	/u01	/u02
PATCH_BASE	/u02	/u01

Note: These variables and values are all created automatically. No user action is required to set them, and they should not be changed manually.

Distributed AD

Many deployments utilize large database servers and multiple, smaller application (middle) tier systems. With the increasing deployment of low cost Linux-based systems, this configuration is becoming more common.

AD has always utilized a job system, where multiple workers are assigned jobs. Information for the job system is stored in the database, and workers receive their assignments based on the contents of the relevant tables. The *Distributed AD* feature offers improved scalability, performance, and resource utilization, by allowing workers of the same AD session to be started on multiple application tier nodes, utilizing available resources to complete their assigned jobs more efficiently.

Requirements for Distributed AD

Because the AD workers create and update file system objects as well as database objects, a shared application tier file system (shared APPL_TOP in earlier releases) must be employed to ensure the files are created in a single, centralized location.

Using Distributed AD

On one of your shared application tier nodes, you start your adop or AD Administration session, specifying the number of local workers and the total number of workers.

While using adop or AD Administration, you can start a normal AD Controller session from any of the nodes in the shared APPL_TOP environment to perform any standard AD Controller operations, using both local and non-local workers. This is possible because the job system can be invoked multiple times during adop and AD Administration runs. Each time an individual invocation of the job system completes, distributed AD Controller sessions will wait until either the job system is invoked again (at which point it will once again start the local workers) or until the AD utility session ends (at which point distributed AD Controller will exit).

Additional Information: See *Oracle E-Business Suite Maintenance Guide* for further details of Distributed AD and AD Controller.

AD Controller Log Files

The log file created by AD Controller is created wherever the AD Controller session is started. This is to prevent file locking issues on certain platforms. It is therefore recommended that the AD Controller log file should include the node name from which the AD Controller session is invoked.

Oracle Applications Manager Features

Oracle Applications Manager (OAM) is a powerful, easy to use tool that enables you to manage and monitor an Oracle E-Business Suite system from an HTML-based central control console.

Among other Oracle E-Business Suite system management tasks, Oracle Applications Manager can help you to:

- Configure and administer your system
- Diagnose and correct problems
- Manage patches
- Monitor and tune performance
- Monitor system security

Oracle Applications Manager is built into Oracle E-Business Suite, and can be used in conjunction with a number of optional tools and features described later in this chapter: including *Oracle Application Management Pack for Oracle E-Business Suite*.

The *Applications Dashboard* provides an overview of the key features of an Oracle E-Business Suite installation, including summaries of current status, performance, critical activities, diagnostics, business flows, and security. You can then view additional information on any of these areas.

Oracle Applications Manager Dashboard

Applications Dashboard

Navigate to [Overview](#) | [Performance](#) | [Critical Activities](#) | [Business Flows](#) | [Security](#) | [Software Updates](#)

Applications System Status

Data Retrieved: 20-Jul-2019 02:06:59

Host	Platform	Host Status	Admin	Database	Concurrent Processing
RWS	Linux x86-64 (64-bit)				

Configuration Changes (last 24 hours)

Data Retrieved: 20-Jul-2019 02:06:59

Patches Applied 0
 Site Level Profile Options 0
 Applications Context Files Edited 0

System Alerts

Data Retrieved: 20-Jul-2019 02:06:59

New Alerts 7
 New Occurrences 7
 Open Alerts 0
 Open Occurrences 0

Web Components Status

Data Retrieved: 20-Jul-2019 08:00:00

Servlet Agent Up
 JSP Agent Up
 TCF Up

User Initiated Alerts

Data Retrieved: 20-Jul-2019 02:06:59

New Alerts 0
 New Occurrences 0
 Open Alerts 0
 Open Occurrences 0

TIP The information shown above (with the exception of Web Components Status section) is retrieved from the system periodically. To retrieve up-to-the-minute data, please use the refresh icon for the desired section. Please see Help for more details.

A more detailed view of the Oracle E-Business Suite system is offered by the *Site Map*, which provides easy access to the numerous features and options that exist in the areas of administration, monitoring, maintenance, and diagnostics and repair.

Administration

Oracle Applications Manager provides a comprehensive system configuration editor, for use in conjunction with the AutoConfig configuration management tool. Previous configuration settings can easily be compared with the current settings, allowing changed settings to be identified and rolled back as necessary.

Site Map - Administration

[Applications Dashboard](#) | [Site Map](#)

Site Map

[Administration](#) | [Monitoring](#) | [Maintenance](#) | [Diagnostics and Repair](#)

System Configuration

Hosts
 AutoConfig
 License Manager
 Fusion Middleware Control
 Weblogic Administration Console

Application Services

Generic Services
 Request Processing Managers
 Transaction Managers
 Parallel Concurrent Programming Setup

Workflow

Home
 Work Item Metrics
 Agent Activity
 Background Engines
 Notification Mailer
 Service Components
 Purge

Concurrent Requests

Submit New
 Pending
 Running
 Completed (Last Hour)

Service Fulfillment Manager

Service Fulfillment Manager

Others

Applications Manager Log
 Knowledge Base

TIP Only the items to which you have access are clickable.

Additionally, Oracle Applications Manager helps you track configuration changes, such as recently altered site-level profile option settings. It can also help you detect potential configuration problems, such as database initialization parameters that do not meet Oracle requirements or recommendations. Oracle Applications Manager can be used to configure and control critical application tier services such as concurrent processing, Forms listeners, and Web servers. For example, you can monitor and administer concurrent requests, and easily analyze the concurrent request workload to make informed configuration changes for concurrent processing.

Oracle Applications Manager can also be used to control Oracle Workflow system services, such as background engines, notification mailers, agent listeners, and queue propagation. You can monitor and analyze Oracle Workflow system activity, suspend and resume processes, retry activities that end in error, and purge obsolete Workflow data.

The *License Manager* component of Oracle Applications Manager enables you to manage all aspects of licensing your products and related features. For example, you can obtain reports of currently licensed products, country-specific functionalities, and languages, as well as an overall licensing summary for your system. You can also license additional products, country-specific functionalities, and languages.

You can extend Oracle Applications Manager using your own custom SQL scripts. Not only can SQL scripts be organized and accessed from within the OAM console, but drilldowns can be enabled from script output to standard Oracle Applications Manager interfaces. For example, if you obtain concurrent request IDs from an SQL script, you can then drill down into the standard Oracle Applications Manager interface to obtain details of the concurrent requests.

Monitoring


Oracle Applications Manager provides extensive monitoring features for Oracle E-Business Suite. Charts and graphs provide high-level summaries, and you can drill down for more detailed information as required.

The Monitoring tab provides key performance metrics related to online, batch, and workflow activity. Additionally, system administrators can now subscribe to receive automated alerts when concurrent requests run longer or wait longer than specified thresholds.

Site Map - Monitoring

Site Map

Administration	Monitoring	Maintenance	Diagnostics and Repair
Availability	Performance	Current Activity	
<ul style="list-style-type: none"> Hosts Database Web Components Internal Concurrent Manager Request Processing Managers Transaction Managers Forms Workflow Business Flows Fusion Middleware Control Weblogic Administration Console 	<ul style="list-style-type: none"> SQL Activity Forms Sessions Forms Runtime Processes Concurrent Processing Reports Concurrent Processing Charts Concurrent Request Runaways Workflow 	<ul style="list-style-type: none"> System Alerts Database Sessions User Monitoring Invalid Objects Forms Runaway Processes Forms Sessions Forms Runtime Processes Application Services Activity Monitors Concurrent Requests Critical Activities Logs 	
System Configuration	Usage	Custom Reporting Utilities	
<ul style="list-style-type: none"> Overview Database Init.ora Setting Applications Context Site-level Profiles 	<ul style="list-style-type: none"> Applications Usage Reports Concurrent Processing Reports 	<ul style="list-style-type: none"> SQL Extensions 	

 **TIP** Only the items to which you have access are clickable.

Monitoring support is provided for application tier services such as concurrent managers and Web services. For currently running processes such as forms or concurrent requests, you can examine both application tier and database session details, right down to the currently executing SQL.

OAM can alert you when certain errors occur within the E-Business Suite system, and will provide detailed context information to help in diagnosing those errors. Information from system components, including performance details, availability, configuration, diagnostic data, and security concerns can all be presented in the context of affected business flows, so that you can better understand any impact on your business.


Maintenance

You can easily determine which patches have been applied to a system, including the individual patches included in mini-packs, maintenance packs, and merged patches. Oracle Applications Manager also enables you to examine the patched files on a system, and identify all the patches that altered a given file. For each patch applied, you can see the actions taken by each patch driver.

Site Map - Maintenance

Site Map

Administration	Monitoring	Maintenance	Diagnostics and Repair
Patching and Utilities		Critical Activities	
<ul style="list-style-type: none"> Applied Patches File History Patch Wizard Timing Reports Register Flagged Files Codelevels Summary 		<ul style="list-style-type: none"> Setup Monitor 	

 **TIP** Only the items to which you have access are clickable.

The *Patch Wizard* tool built into Oracle Applications Manager can recommend patches for your Oracle E-Business Suite system. The wizard takes patch data downloaded from Oracle, analyzes that data against the Oracle E-Business Suite system, and recommends patches based on criteria you specify. The wizard can also analyze individual patches, identify any prerequisites missing on the system, and show the impact that the patches would have on the system in terms of affected applications, files, and other areas. Additionally, the wizard can download multiple patches from Oracle and merge them into a single patch.

Administrators can use the *Manage Downtime* feature to schedule downtimes. When a downtime is scheduled, a message on the Home Page notifies users of the period when the system will not be available. The user has to log in, or return to the Home Page to see this message. If the downtime is in progress, an appropriate warning page will replace the login page.

During the scheduled downtime period, administrators with suitable database privileges are able to log into OAM to monitor the progress of AutoPatch and other Applications DBA (AD) utilities: access to other parts of Oracle E-Business Suite is not allowed.


Diagnostics and Repair

Oracle Applications Manager for Oracle E-Business Suite Release 12 provides troubleshooting wizards that automate complex recovery steps. Tools such as Concurrent Manager Recovery help you diagnose problems and take corrective action from the same screen. The Service Infrastructure Troubleshooting Wizard ensures that the Generic Service Management (GSM) infrastructure is running normally, and if not, guides the administrator through the steps needed to bring GSM back up.

Site Map - Diagnostics and Repair

Site Map

Administration	Monitoring	Maintenance	Diagnostics and Repair
Diagnostics		Troubleshooting Wizards	
<ul style="list-style-type: none"> Execute Diagnostic Tests View Diagnostics Reports View Scheduled Diagnostics Tests Diagnostics Configuration Debug WorkBench 		<ul style="list-style-type: none"> Concurrent Manager Recovery Service Infrastructure GCS and Forms Monitoring CP Signature Dashboard Collection Signature 	

 **TIP** Only the items to which you have access are clickable.

The Applications Dashboard provides a gateway into Oracle Diagnostics for Oracle E-Business Suite, offering graphical views of diagnostic test executions and failures, summarizing test results, and providing access to detailed test reports. Keyword searches can be performed on the test repository to locate relevant tests, which can then be grouped and executed directly from the same interface.

Additionally, system administrators can schedule and run diagnostics tests as batch programs, and output from all diagnostic interfaces can be captured and packaged for shipment to Oracle Support quickly and easily. Other features include provision for customers to download the latest diagnostics pack supplied by Oracle Support, and create their own diagnostic tests as required.

Finally, the system alerting infrastructure has been enhanced with a new flood control mechanism to prevent multiple instances of a given alert from overwhelming the alert system. Additionally, alerts have been enhanced to allow summary and context information to be retrieved in XML format via standard business event integration mechanisms.

Support Cart

The OAM *Support Cart* automates the collection of detailed node-specific information that can be sent to Oracle Support to assist with problem diagnosis.

Oracle Applications Manager Support Cart

[Applications Dashboard](#) | [Site Map](#)

Support Cart Contents

Description	Applications Signature	Other Information Collected
TAR Number	<input type="text"/>	
Details	<div style="border: 1px solid #ccc; padding: 5px;"><p>Please use this section to provide the following information:</p><ul style="list-style-type: none">(1) Problem description(2) Steps to reproduce the issue(3) System, Platform, Environment specific information and(4) Error messages.<p><input checked="" type="checkbox"/> TIP</p></div>	
<input type="button" value="Update"/>	<input type="button" value="Reset"/>	

The information collected includes details of products installed, patches applied, database version, current database parameters, and system topology. Other pages from OAM can be added to the Support Cart if desired.

Extending Oracle Applications Manager Functionality

Oracle Applications Manager can be used in conjunction with *Oracle Enterprise Manager Grid Control* via the supporting *Oracle Application Management Suite for Oracle E-Business Suite*, to create a comprehensive, top-down approach to system monitoring and administration.

Oracle Application Management Suite for Oracle E-Business Suite extends Oracle Enterprise Manager Grid Control to help monitor and manage Oracle E-Business Suite systems. Features include simplified administration, hot cloning (where the database does not need to be shut down), and management of advanced topologies.

Generic Service Management

Traditionally, application tier processes such as Forms listeners, HTTP servers, and concurrent managers had to be started and monitored individually by system administrators, a time-consuming and potentially error-prone exercise. With the transition to web-based applications, the number of application tier processes required for Oracle E-Business Suite products has increased significantly, with many Oracle E-Business Suite products making use of multiple application tier services to support one or more processes.

Service processes are similar to concurrent manager and transaction manager processes, and must be kept running on an application tier for the proper functioning of their associated products. Management of the services is complicated by the fact that they may be distributed across multiple host machines.

The *Generic Service Management (GSM)* feature simplifies management of these generic service processes, by providing a fault-tolerant framework with a central management console built into Oracle Applications Manager. With Generic Service Management, the Internal Concurrent Manager (ICM) manages the various service processes across multiple hosts. On each host, a *Service Manager* acts on behalf of the ICM, allowing the ICM to monitor and control service processes on that host.

System administrators can configure, monitor, and control services through Oracle Applications Manager, which communicates with the ICM. Generic Service Management provides a fault-tolerant system: if a service process exits unexpectedly, the ICM will automatically attempt to restart the process. If a host fails, the ICM may start the relevant service processes on a secondary host. The ICM itself is monitored and kept alive by Internal Monitor processes located on various hosts.

This application tier service management infrastructure has several benefits:

- The service processes no longer need to be manually and individually started and monitored by Oracle E-Business Suite system administrators
- Administrators can configure and control the services through Oracle Applications Manager
- As with concurrent manager processes, system administrators can use work shifts to determine the number of processes that will be active for a service on a given node for a given time period
- Services can take advantage of the process distribution and fault tolerance capabilities that have been developed for concurrent processing

Page 8 of 18
< (https://docs.oracle.com/cd/E26401_01/doc.122/e22949/T120505T120514.htm)



[About Oracle](http://www.oracle.com/corporate/index.html) | [Contact Us](http://www.oracle.com/us/corporate/contact/index.html) | [Legal Notices](http://www.oracle.com/us/legal/index.html) | [Terms of Use](#)

<http://www.oracle.com/us/legal/terms/index.html> | [Your Privacy Rights](http://www.oracle.com/us/legal/privacy/index.html) | [Cookie Preferences](#) | [Ad Choices](#)

[privacy-policy.html#12](#)

Copyright © 2025, Oracle and/or its affiliates. <http://www.oracle.com/pls/topic/lookup?ctx=cpy&id=en>