Oracle® Fusion Applications

Patching Guide

Release 12 (11.12.x.0.0)
E71563-04
January 2017

Oracle Fusion Applications Patching Guide, Release 12 (11.12.x.0.0)

E71563-04

Primary Authors: Edith Aceves, Keila Chavez

Contributors: Emilio Jasso, Subash Chadalavada, Lori Coleman, Rick Lotero, Shashi Handalgere, Prashant Salgaocar, Venkatesh Sangam, Praveena Vajja

Contents

Preface

Documentation for installers and system administrators describing the patching framework, tools and processes to update and maintain Oracle Fusion Applications software between major releases.

Audience

This guide is intended for system administrators, database administrators, and Identity Management (IDM) experts who are responsible for performing Oracle Fusion Applications patching tasks.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle's Accessibility Program.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit Global Customer Support Accessibility

If you are hearing impaired, visit Global Customer Support Accessibility.

Related Documents

For more information, see the following documents:

- Oracle® Fusion Applications Administrator's Guide Release 12 (11.12.x.0.0)
- Oracle® Fusion Applications Installation Guide 11g Release 12 (11.12.x.0.0)
- Oracle® Fusion Applications Upgrade Guide 11g Release 12 (11.12.x.0.0)

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

News

Oracle Fusion Applications Release 12 (11.12.x.0.0) includes the following new and changed patching features:

- For Release 12 (11.12.x.0.0) the guide has been entirely reorganized and overhauled to make it easier to use for novice, intermediate, and expert users.

New and Changed Features for Release 12 (11.12.x.0.0)

Oracle Fusion Applications Release 12 (11.12.x.0.0) includes the following new and changed features of the Oracle Fusion Applications patching process and other significant changes that are described in this guide, and provides pointers to additional information:

- The introductory section provides a novice approach, considering the knowledge of a new participant on the patching environment. See the Introduction to Patching Oracle Fusion Applications (page 1-1) section.
- The patch planning section ensures a successful and comprehensive patching strategy. See the Planning for Patching (page 3-1) section.
- The functionality of Technical Patch Bundles (P4FA) together with FASPOT utility automates the installation of several individual patches in single bundle. See the Applying P4FA Patches (page 4-1) section.

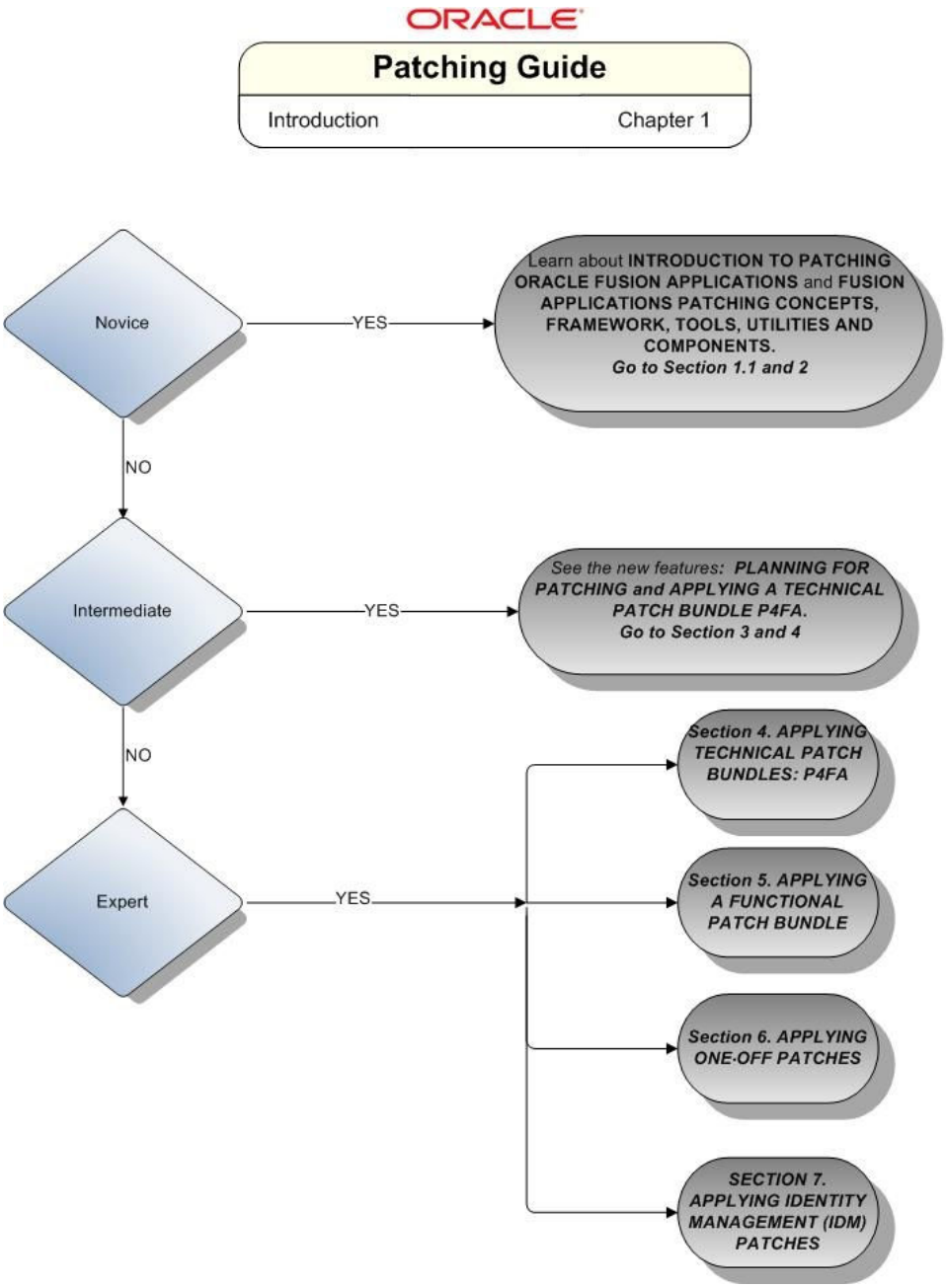
In this user guide, the nomenclature “11.12.x.0.0”, where “x” is a number, is used to indicate the release and patch releases for which the guide is applicable. When using this document be sure to replace “x” with the number of the release that is being used.

1

Introduction to Patching Oracle Fusion Applications

This Patching Guide is intended to be a detailed and helpful tool for understanding the patching environment regardless of the user's level of knowledge.
In order to understand better what section to consult according to your expertise in patching, see the decision tree below:

Figure 1-1 Decision Tree according to your expertise



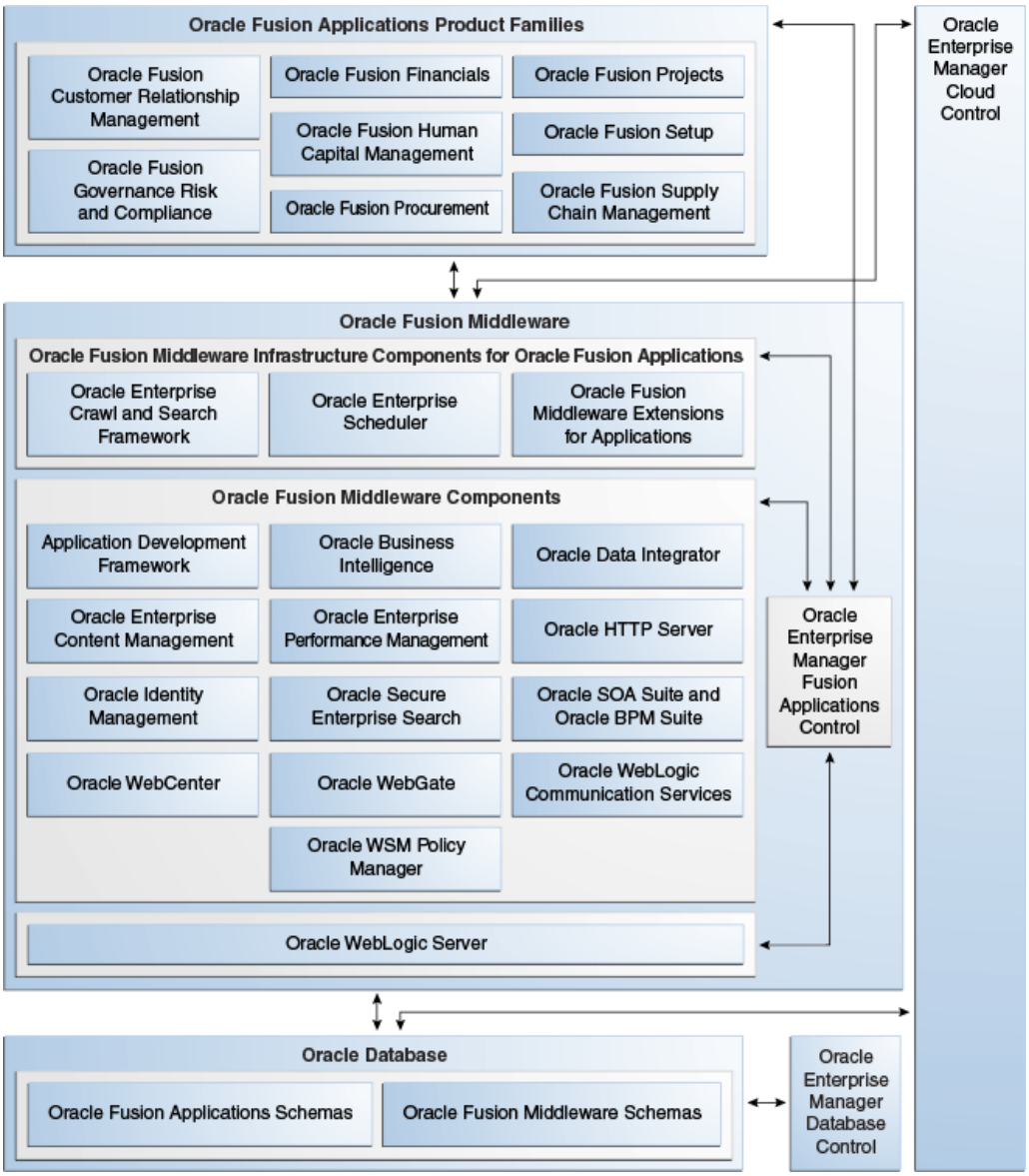
1.1 Introduction to Patching

Patching is the process of applying modifications to the Oracle Fusion Applications environment. Patching is required to close security vulnerabilities, fix bugs, improve performance, and add new features. Continuous patch application minimizes incidents where bugs are encountered, improves security and ensures the instance of Oracle Fusion Applications is performing efficiently.

1.2 Tech Stack View and Interdependencies

Oracle Fusion Applications is a layered architecture that comprises development tools, distributed applications, middleware, databases, among other infrastructure components, all included in a standards-based platform. As shown in the diagram below, since the systems are integrated with complex interdependencies and linked through WebLogic Servers, deploying bundle patches to one product family can impact other tech stack components. For example, applying a patch to Oracle Fusion Supply Chain Management may affect a component within Oracle Fusion Human Capital Management. Further explain the types of product families and that there are, in fact, interdependencies:

Figure 1-2 Oracle Fusion Application Product Families

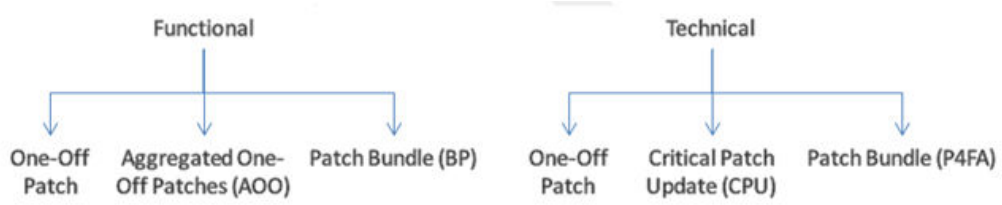


1.3 Patch Categories

The two categories of patches for Fusion Applications is as follows:

- **Functional patches** deliver new or fix existing functional capabilities within Fusion Applications. Functional patches are delivered in the following three formats:
  - One-Off Patch
  - Aggregated One-Off Patches (AOO) and
  - Patch Bundle (BP)
- **Technical Patches** deliver new or fix existing technical capabilities in the underlying Fusion Middleware Components upon which Fusion Applications exists. The Technical patches are delivered in three formats are as follows:
  - One-Off Patch,
  - Critical Patch Update (CPU)
  - Patch Bundle (P4FA)

Figure 1-3 Oracle Fusion Application Patch Categories Types



1.4 Planning

The planning section has been added to organize the patching strategies for the Oracle Fusion Applications environment in a more logical way.

See the Plan for Patching (page 3-1) section to understand the types of patches and their release schedules, asses the effect for each deployment, create a tracking sheet to document the plan and outcomes, coordinate people, back up databases and file systems, and estimate schedules and down times.

1.4.1 Key Roles and Expertise

The patching team must have technical and organizational knowledge and typically includes both administrators and regular team members. See the Patch Plan: Skills and Knowledge Required (page 3-1) section to know more about the skills and knowledge required to complete the patching progress.

1.4.2 Patch Time

The patching team is in charge of scheduling a downtime time to maximize the uptake of the patch bundles. Multiple patch bundles for both technical and functional

side are published periodically for each release. The best practice for scheduling the uptake for these patches is to establish a schedule that minimizes the downtime while maximizing the uptake of key patches for the organization. See the Time of Patch Bundles Releases (page 3-5) section to determine the best frequency to install the patches.

1.5 Find Patches

Patches can be found and downloaded from My Oracle Support. For specific instructions related to finding Technical Patch Bundles, see the Download a Technical Patch Bundle (P4FA) (page 4-2) section. For information about Functional Patch Bundles, see the Download the Latest Patch Bundle and Its Prerequisites (page 5-2) section.

1.6 Patching Tools

MANDATORY: Ensure that the framework is up to date. Tools themselves have to be patched as described in the Ensure The Patching Tools are Up-to-Date (page 3-2) section.

The tools used to apply patches depend on what type of patch is being applied. Tables 1–1 and 1–2 show the tools used for each type of patch, as follows:

Table 1-1 Technical Patching Tools

TECHNICAL PATCHES	TOOL
Technical One-Off	OPatch
Critical Patch Updates (CPU)	OPatch
Technical Patch Bundle (P4FA)	FASPOT as described in the Apply Technical Patch Bundles: P4FA (page 4-1) section

Table 1-2 Functional Patching Tools

FUNCTIONAL PATCHES	TOOL
Functional One-Off	Fusion Applications Patch Manager as described in the Apply One-Off Patches (page 6-1) section
Functional Patch Bundle (BP)	Fusion Applications Patch Manager as described in the Apply a Functional Patch Bundle (page 5-1) section
Aggregated One-Off	Fusion Applications Patch Manager as described in the Apply a Functional Patch Bundle (page 5-1) section

2

Patching Concepts, Framework, Tools, Utilities, and Components

Patching requires a full understanding of several concepts, and this section describes three Fusion Applications patching concepts

2.1 Frameworks Available for Oracle Fusion Applications

Oracle Fusion Applications is updated by the following patching frameworks:

2.1.1 Technical Patches

Technical patches are geared toward delivering new or fixing existing technical capabilities in the underlying Fusion Middleware components that Fusion Applications is built on, as described in the following table:

Table 2-1 Types of Oracle Fusion Applications Technical Patches

Patch Type	Description	Frequency
One-Off Patch	Contain a single bug fix	As needed for critical fixes
Critical Patch Updates (CPUs)	Address security vulnerabilities	On the Tuesday closest to the 17th day of January, April, July, and October
Patch Bundle (P4FA)	Collection of tech stack fixes (bundles or one-offs) for Fusion Applications customers	Monthly

2.1.2 Functional Patches

Functional patches deliver new functional features or fix existing functional capabilities within Fusion Applications. Functional patches are delivered in three formats and all are applied using Patch Manager, as described in the following table:

Table 2-2 Types of Oracle Fusion Applications Functional Patches

Patch Type	Description	Frequency
Functional Patch Bundles	Collection of one-off patches for a product family, which are cumulative and distributed for a specific baseline release. They include previously released Aggregated One-Off Patches (AOOs) and One-Off Patches	Monthly
Aggregated One-Off Patches (AOO)	Collection of One-Off Patches for a product family	Weekly, but not all product families release AOOs weekly
Functional One-Off Patches	Contain a single bug fix	As needed for critical fixes

2.1.3 Oracle Identity Management Patches

The primary purpose of the Oracle Identity Management patching framework for Oracle Fusion Applications is to simplify and expedite the maintenance of the code and functionality shipped as part of Oracle Identity Management for the Oracle Fusion Applications suite of products.

2.2 About Patch Manager and Functional Patching Topology

During the provisioning of Oracle Fusion Applications, the configuration of the patching framework performs the following:

- The `FUSION_env.properties` file is populated in the admin directory with complete environment setup information required by the patching framework. This is the source of information that patching framework utilities use when setting up the environment.
- The patching framework configuration scripts are created to set the environment and call utilities. For example, it creates the `fapmgr.sh` script, which sets up the environment and then calls Oracle Fusion Applications Patch Manager (Patch Manager) when applying the patch.

The Patching topology and configuration topics include the following:

- Understand Patch Manager and Functional Patching (page 2-3)
- Oracle Fusion Applications Oracle Home (page 2-8)
- Patch Top Directory (page 2-9)
- Backup Copies of Patched Database Artifacts (page 2-9)
- Oracle Universal Installer (OUI) Inventory (page 2-10)
- Taxonomy URL (page 2-10)
- One-Off Patch Directory Structure (page 2-11)
- One-Off Patch Contents (page 2-12)

2.2.1 Understand Patch Manager and Functional Patching

The primary function of Oracle Fusion Applications Patch Manager (Patch Manager) is to apply Functional Patches. It also validates whether patches can be applied and generates patching reports. Patch Manager supports parallel processing of non-dependent tasks, thereby improving the performance during patch application.

Patch Manager provides a command-line interface to coordinate its patching functions. A single patch may include changes to both Oracle Fusion Middleware and database artifacts, and these Oracle Fusion Middleware artifacts may be deployed to Managed Servers running on different nodes. The artifacts are updated in the Oracle Fusion Applications Oracle home that is shared by the different nodes. To patch both types of artifacts, two patching tools are called by Patch Manager to manage the actions involved: *OPatch* for the Oracle Fusion Middleware artifacts and *Oracle Fusion Applications AutoPatch* (AutoPatch) for artifacts associated with the database.

The same set of patching-related software and database tables is used by both Patch Manager and Oracle Fusion Middleware Extensions for Applications (Applications Core). Patch Manager and Applications Core each reside in their own separate Oracle home and use their specific shell scripts to support their product-specific patching requirements. These scripts are uniquely defined to reference the appropriate Oracle home, set the patching configuration and environment, and then call the AutoPatch utility for database patching.

WARNING: There can be only one patching session active for Oracle Fusion Applications or Applications Core at a time.

For detailed information about how to run Patch Manager, see the Patch Manager Command Reference (page 9-1) section.

2.2.1.1 Patch Modes

The patch administrator is responsible for ensuring there are no active transactions or processes running during patching. The three patching modes supported by the Patch Manager are as follows:

- Online Mode (page 2-4)
- Offline Mode (page 2-4)
- Hot Patching Mode (page 2-4)

Each patching mode type affects users, servers, artifacts, and the database in different ways, as described in the following table:

Table 2-3 Effect of Oracle Fusion Applications Patching Modes

Patching Mode	Online	Offline	Hot Patch
Users	No Access	No Access	Active
Servers Bounced	Automatically	Manually	No Bounce
Artifacts Deployed	Automatically	Manually	Automatically
Database Use	Idle	Idle	Active

2.2.1.1.1 Online Mode

MANDATORY: To apply any patch in online mode, the Administration Servers must be running.

Before applying the patch, review which servers will be impacted by the patch. For more information, see the Online Patch Progress Report and Diagnostics Report (page 9-7) section.

In online mode, Patch Manager automates the post-apply steps, such as shutting down and starting the impacted Managed Servers, and deploying supported Oracle Fusion Middleware artifacts, such as SOA composites, Oracle Business Intelligence Publisher (Oracle BIP) artifacts, and Flexfields. When patching in online mode, Patch Manager provides messages about the steps to take after the patch is applied, to resolve any failures that occurred during the post-apply tasks. For more information, see the Oracle Fusion Applications Patch Manager Middleware Artifact Support (page 10-2) section.

To enable online patching mode, specify the `online` option and the `stoponerror` option when running Patch Manager. Patch Manager determines which domains the patch affects by referencing the taxonomy URL, either by an environment setting or by using the `taxonomyurl` option. For more information, see the Taxonomy URL (page 2-10) section.

In online mode, only those impacted servers that are running are stopped and started. No stop or start operations are performed on those servers that are not in a running state even if the patch impacts an application that is deployed on this server.

2.2.1.1.2 Offline Mode

In offline mode, server management is performed manually.

OPTIONAL: Before applying the patch, the Patch Impact report can be run to see which servers will be impacted by the patch.

In offline mode, all applications are unavailable to users, but only the servers impacted by the patch are shut down. The net effect is that the system is unavailable, but the system downtime is minimized if only certain servers are shut down and then started.

MANDATORY: All servers impacted by the patch must be shut down before applying patches in offline mode, and when applying a patch in offline mode, manually start and stop the impacted Managed Servers and manually deploy certain Oracle Fusion Middleware artifacts, such as SOA composites, Oracle BI Publisher artifacts, and Flexfields, after the patch applies.

OPTIONAL: To minimize downtime, servers can remain running during patching. The impacted server stop and start can occur after the patching session ends.

For information about deploying patch artifacts manually when applying a patch in offline mode, refer to the Manual Patching of Oracle Fusion Applications During Offline Patching (page 10-1) section.

2.2.1.1.3 Hot Patching Mode

Hot patching mode allows a patch to be applied in online mode concurrent with active users and transactions. Hot patching is supported for only database, JEE, BIP, MAA (Mobile Artifact), SOA, and C artifacts.

MANDATORY: The patch must be enabled for hot patching.

During hot patching, an application is updated by deploying a new version of the application with modified artifacts from the patch. Any requests to the application are directed to the new version once it is available. A user may experience a small performance impact due to the added load on the system by patching sessions and also due to initialization when an updated application is accessed for the first time.

The following phases occur during hot patching mode before the patch is actually applied:

- The system is placed in maintenance preparatory warning mode. In this mode users are informed of the upcoming maintenance window.

  MANDATORY: All users impacted by the applications must complete their activities before the maintenance window begins.

- During maintenance warning mode, the impacted ESS task schedulers are paused and the impacted SOA Event Delivery Network is paused.

- When the specified waiting period ends, background processes are checked to ensure there are no active tasks. If the `-forceterminateactivetasks` option is specified at the command line, all active tasks are terminated. Otherwise, a list of tasks is printed and the patching session exits with a failure.

See Apply Patches in Hot Patching Mode (page 5-6) for more information about hot patching mode.

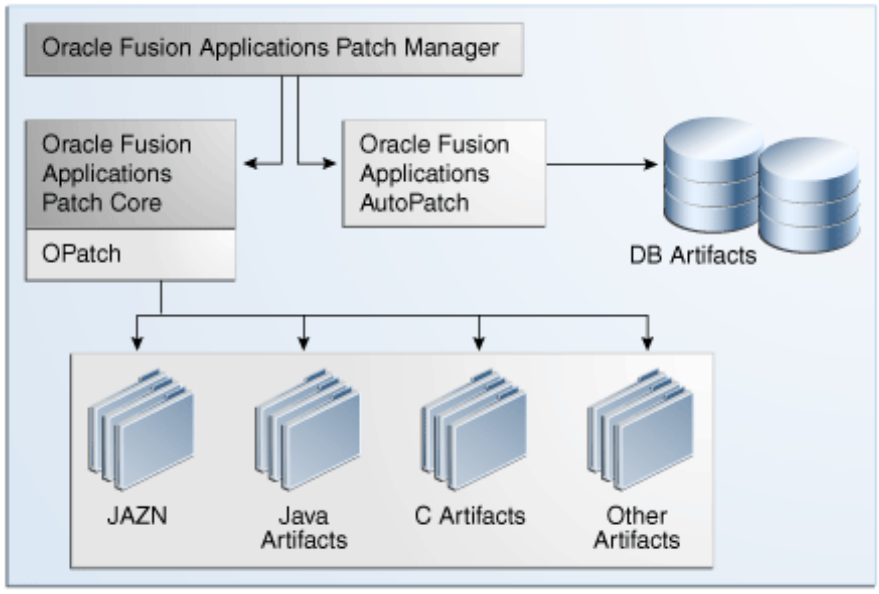
2.2.1.2 Coordinated Patching with Patch Manager

When a patch contains Database and Oracle Fusion Middleware changes, Patch Manager coordinates the application of both changes, applying database changes first, followed by Oracle Fusion Middleware changes. The patch is applied in a single operation, regardless of the type of artifacts that are updated.

Patch Manager examines patch metadata and determines which actions must be performed by OPatch and which must be performed by AutoPatch. If Patch Manager discovers that a patch contains only database changes, it assigns the patch directly to AutoPatch for processing. If the patch is related only to Oracle Fusion Middleware changes, Patch Manager orchestrates the application of the changes across domains and the Oracle home.

The following figure illustrates the patching process coordinated to Patch Manager:

Figure 2-1 Oracle Fusion Applications Coordinated Patching



The following high-level phases occur when applying a patch in online mode that contains an Oracle Application Development Framework (Oracle ADF) library and a seed data file:

- Patch Manager interprets the contents of the patch by reading the patch metadata.
- AutoPatch updates the seed data.
- OPatch applies the change to the Oracle ADF library in the form of a JAR file.
- Patch Manager coordinates with OPatch and forces an immediate shutdown and restart of the impacted Managed Servers so the change to the Oracle ADF library takes effect.
- Patch Manager consolidates and provides results and status for the overall patching tasks in the Log Summary and the Diagnostics report.

2.2.1.3 Patch Database Artifacts

A patch with database-related changes includes a patch driver file that provides instructions to AutoPatch about how to apply the patch. The patch driver file specifies the types of actions to be executed and the phases in which they are executed.
To achieve efficient processing time, the database tasks are performed by worker processes and the number of tasks performed is minimized by file version verification.. See the About Worker Processes (page 2-7) and the File Version Verification (page 2-7) sections to understabd how Autopatch works.

When a patch contains updates to database artifacts, such as application seed data, the database schema, PL/SQL objects, and SQL scripts; Patch Manager calls AutoPatch to coordinate the following tasks:

- **Worker calculation**: Calculates the default number of workers that are necessary. If patching is run on the same machine as the database server, the default number of workers is calculated as 0.5 times the number of Virtual CPUs (VCPUs) on the

database server. If patching is run on a machine different from the database server on a Linux platform, the default number of workers is calculated as the minimum of the VCPUs available on the database server and the patching machine. On non-Linux platforms, the default number of workers is equal to the number of VCPUs on the database server. Reduce the number of workers If the machine where the patch is applied has a lower number of VCPUs when compared to those on the database server, then reduce the number of workers.

OPTIONAL: To override the default number of workers when applying a patch, specify the number of workers by using the `workers` option.

The number of workers used for patching database artifacts also imposes a requirement on the open file descriptors configured for the system. Patching requires that the open file descriptors be set to a minimum of 8000 times the number of workers used for the patch session.

- **Patch validation** : Validates whether the database portion of the patch is compatible with the environment and can be applied. If the patch is not valid and the patching session fails, see the Monitor and Troubleshoot Patches (page 8-1) section. If the patch is valid, the following validations are performed:
  - Platform check: Compares the operating system platform for each Oracle Fusion Applications Oracle home against the platform metadata in the patch.
  - Prerequisite check: Validates that all patch prerequisites have been applied.
- **Patch Application**: Copies the database artifacts to the Oracle home and then makes changes in the Oracle Fusion Applications database using the updated files.
- **Invalid Object Compilation**: Compiles all invalid objects in the database.
- **Consolidation of log files**: Collects the patching results and location of log files for reporting purposes.

2.2.1.3.1 About Worker Processes

An AutoPatch `manager` process reads the patch driver file and determines the set of tasks to be performed. It then spawns processes called **workers** to execute the tasks. The manager and its workers communicate through a table in the database, which contains one row for each worker process. The manager assigns tasks to workers by updating the worker row in the table. Each worker process checks the table for updates to its row and carries out the task. When the task is complete, the worker updates the status in the table, and the manager then assigns another task to the worker.

2.2.1.3.2 File Version Verification

AutoPatch performs file version verification to ensure that only new actions run during patch applications.

CONDITIONAL: AutoPatch runs the action only if the version in the patch is newer than the last version run.

2.2.1.3.3 Compile Invalid Objects

Patch Manager uses the standard database-supplied compile utility, which compiles all invalid objects in the database, if no specific schema is supplied. If a schema is supplied it compiles all objects in the schema that are in an invalid state, including

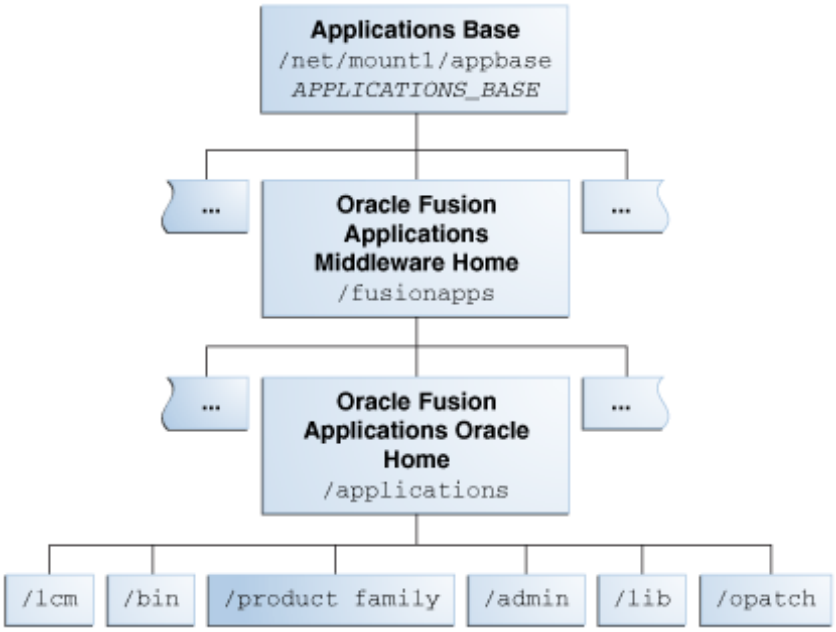
those invalid objects that were not affected by the patch. Dependencies between objects can be complex, such as when patching an object causes other objects to become invalid, even though those objects are not in the patch. The purpose of compiling invalid objects after a patch applies is to have a clean database where all objects are in a valid state.

2.2.2 Oracle Fusion Applications Oracle Home

During provisioning, the patching framework and the Oracle Fusion Applications software were installed into what is known as the *Oracle Fusion Applications Oracle home*. This Oracle home directory, `/net/mount1/appbase/fusionapps/applications`, is a subdirectory under the Oracle Fusion Applications Middleware home. The top level directory, `/net/mount1/appbase`, is referred to as the `APPLICATIONS_BASE`, and is where all Oracle Fusion Applications binaries reside. There is one and only one set of patching-related software and database tables for each Oracle home. Unless otherwise specified, the use of "Oracle home" and *`FA_ORACLE_HOME`* in this guide refers to the Oracle Fusion Applications Oracle home.

The following figure shows the related directory structure, beginning with `APPLICATIONS_BASE`:

Figure 2-2 Oracle Fusion Applications Directory Structure



The Oracle home contains the following subdirectories:

- **lcm**: Contains the patching framework software in the following subdirectories:
  - `.../ad/bin`: Patching framework software and files, including C artifacts and configuration scripts that set the environment and start the corresponding utility
  - `.../ad/java`: Java artifacts

  - `.../ad/db/sql`: Database artifacts and SQL files
  - `.../ad/lib`: Application libraries
  - `.../ad/template`: Configuration files or templates delivered and used by the patching framework during configuration activities
- **bin**: Contains applications artifacts called by Enterprise Scheduler Service jobs.
- ***product family***: Contains directories for artifacts specific to a product configuration.
- **admin**: Contains the patching framework environment properties file (`FUSION_env.properties`), Oracle Fusion Applications AutoPatch (AutoPatch) and the patching logs, reports, and administration files.

  MANDATORY: These files are required by Patch Manager.
- **lib**: Contains applications-specific libraries.
- **OPatch**: Contains the OPatch utility called by Patch Manager when patching middleware artifacts. This version of OPatch is used to apply patches to the middleware files and software artifacts that reside within the Oracle Fusion Applications Oracle home, and is delivered as part of the Oracle Fusion Applications software. There may be multiple versions of OPatch to support the enterprise software. However, if a newer version is required it will clearly stated in the README of patch set to be applied.

  CONDITIONAL: When applying patches to the Oracle homes, on hosts where Oracle Fusion Middleware Oracle homes co-exists with *`FA_ORACLE_HOME`*, OPatch from *`FA_ORACLE_HOME`* has to be used. For example, SOA (`fusionapps/soa`) and ATGPF (`fusionapps/atgpf`) Oracle homes exist on the same file system as *`FA_ORACLE_HOME`* (`fusionapps/applications`). When applying patches or executing any OPatch related commands for SOA and ATGPF, the OPatch from *`FA_ORACLE_HOME`* (`fusionapps/applications/OPatch/opatch`) must be used.

Oracle Fusion Middleware Oracle homes and Oracle Fusion Applications Oracle home are read only and customers are not expected to update or install any components manually to these home directories. These home directories can be updated only by Oracle Fusion Applications LifeCycle tools, such as Provisioning, Upgrade Orchestrator, and Patch Manager.

2.2.3 Patch Top Directory

The patch top directory is any directory selected for downloading patch ZIP files. This directory is also called `patch_top` or `PATCH_TOP`. For example, if patch 1234567.zip is downloaded into `/home/mypatches` and unzipped there, the patch top directory is `/home/mypatches/1234567`.

2.2.4 Backup Copies of Patched Database Artifacts

When a patch with a later version of an existing database artifact is applied on the Oracle Home, Patch Manager automatically backs up the existing database artifacts and restores them into a backup directory. The default location for the backup directory is `admin/pbackup` under the Oracle home. This location may be overridden by editing the `PATCH_BACKUP_DIR` parameter in the `FUSION_env.properties` file.

2.2.5 Oracle Universal Installer (OUI) Inventory

The Oracle Universal Installer (OUI) inventory stores information about all Oracle software products installed in all Oracle homes. Each product, such as Oracle Fusion Applications, maintains its own local inventory and Oracle home. The Local inventory files for Oracle Fusion Applications are located in the Oracle Fusion Applications Oracle home where they are read and updated by the patching framework.

The OUI inventory has the following hierarchical structure:

- Central Inventory Pointer File: The Central Inventory is located in the directory that the inventory pointer file specifies. Each Oracle software installation has its own Central Inventory pointer file that is unknown to another Oracle software installation. The following table shows the location of the default inventory pointer file for various platforms:

Table 2-4 Default Inventory Pointer File Locations

Platform	Default Inventory Pointer Location
Linux Linux.PPC64 AIX	`/etc/oraInst.loc`
Solaris.SPARC Solaris.X64 HPUX HPIA HP.TRU64 Linux.IA64 Linux.xSeries	`/var/opt/oracle/oraInst.loc`

- Central Inventory File: This file, `inventory.xml`, is present in the following location: *`central_inventory_location`*`/ContentsXML/inventory.xml`. It contains a list of Oracle homes installed on the node.
- Oracle Home Inventory: The Oracle home inventory or local inventory is present inside each Oracle home and contains only information relevant to a specific Oracle home. This file is located in the `$ORACLE_HOME/inventory` and contains the following files:
  - Components File
  - Home Properties File
  - Other Folders

Each Oracle home contains OUI components. In Oracle Fusion Applications, each product family is assigned an OUI component and other entities are also assigned a component. For example, the component `oracle.fusionapps.fin` is assigned to Oracle Fusion Financials. The patching framework uses this information to identify and determine the specific contents of the patch that are applicable to the Oracle home, as well as to perform patch validation, patch verification, and reporting.

2.2.6 Taxonomy URL

Patch Manager queries the taxonomy MBean URL (as defined by the environment property called `taxonomy_url`) to determine which domain is affected by a specific patch. For example, to determine where a Java EE application is running or where a Service-Oriented Architecture (SOA) composite is deployed. The URL points to an Administration Server of the domain where taxonomy MBeans are hosted. This variable is set during the provisioning process in the `FUSION_env.properties` file. This value can be overridden during patching by providing the `taxonomyurl` option when

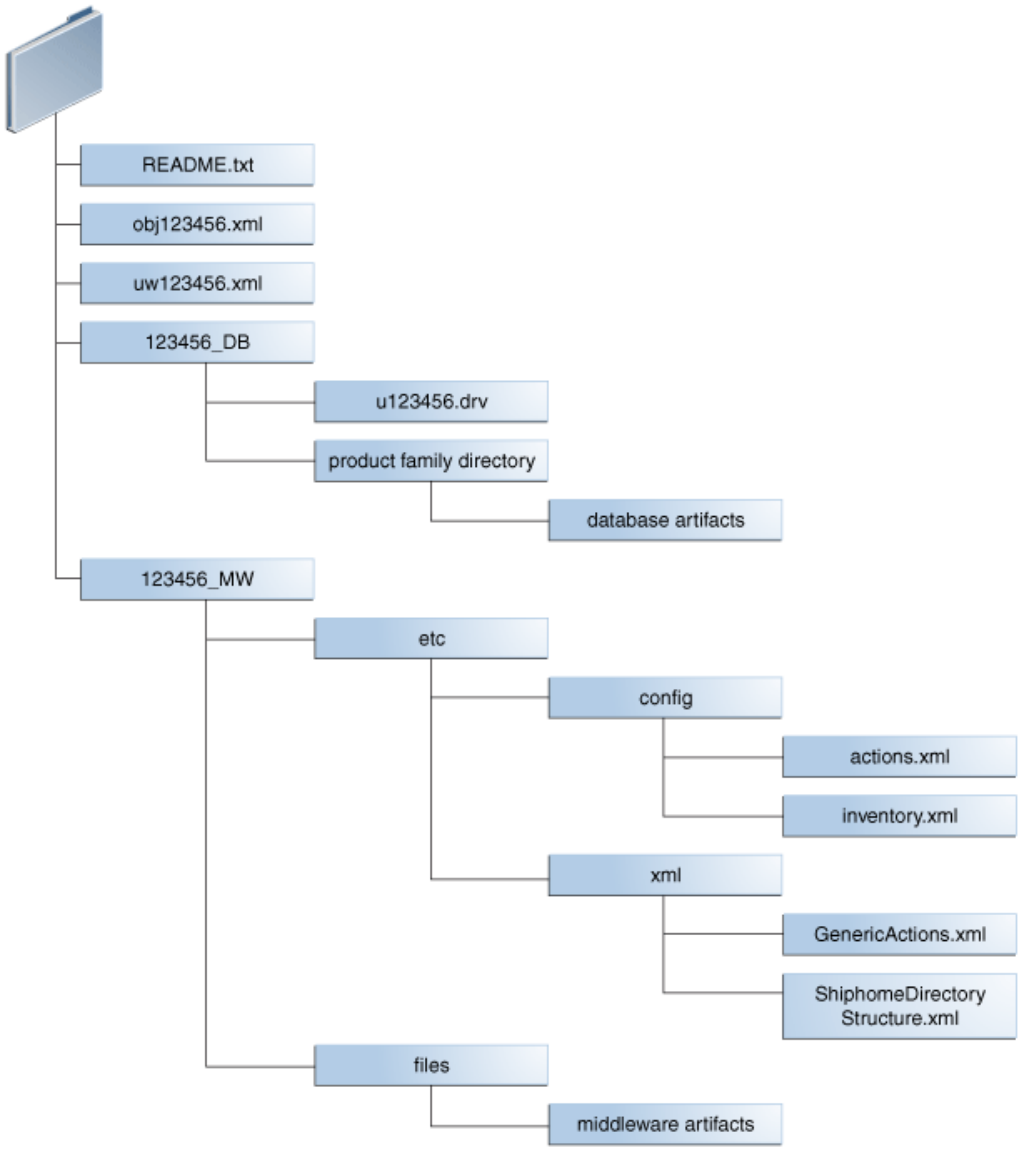
running Patch Manager. For example, if the server being referenced by the default `taxonomy_url` is down, enter an overriding URL from the command line.

2.2.7 One-Off Patch Directory Structure

Oracle Fusion Applications patches often include content for both middleware artifacts and database artifacts. The patching framework examines the high-level contents of each patch and calls the appropriate patching tool to process the patch content.

If the patch only contained database artifacts, the `12345_MW` directory would not exist. If the patch only contained middleware artifacts, the `123456_DB` directory would not exist. Using patch number 123456 as an example of a patch that contains both database and middleware artifacts, the unzipped patch directory, *`PATCH_TOP`*`/123456`, contains the files and subdirectories shown in the following figure:

Figure 2-3 Example Directory Structure of a One-Off Patch



2.2.8 One-Off Patch Contents

A sample of the Patch Content is described below, using patch number 123456 as an example of a patch that contains both database and middleware artifacts:

- `README.txt`: Provides general instructions to apply the patch and to perform manual steps, if required by the patch. If there are patches listed under "Other Patches" in the `README` file, they must be downloaded and applied before deploying the Oracle Fusion Applications patch.
- `obj123456.xml`: Contains information about each artifact included in the patch.

  An example of the contents of the `obj123456.xml` is described below:

```
<?xml version="1.0" encoding="UTF-8"?>

<PATCH_OBJECT_MANIFEST VERSION="1.0">
 <COMPONENT TYPE="MW">
   <OBJECT_INFO NAME="AdfPjgTopPublicUi.jar"
SUBDIR="prj/deploy/EARProjectsFinancials.ear/EARProjectsFinancials/WEB-INF/
lib"
SRCDIR="prj/deploy/EARProjectsFinancials.ear/EARProjectsFinancials/WEB-INF/
lib"
PRODUCTFAMILY="prj" PRODUCT="pjg" LBA="PjgTop"
APPNAME="EARProjectsFinancials.ear"
HEADERSTRING="$AppsHeader:fusionapps/prj/components/projectsFinancials/jlib/
AdfPjgTopPublicUi.jar st_fusionapps_pt/63 level:0 00.S $"

OUI_COMPONENT="oracle.fusionapps.prj.deploy" VERSION="63.0"
TRANSLATION_LEVEL="0" ACTION="COPY" ARTIFACT_TYPE="JEE" />
 </COMPONENT>
       <COMPONENT TYPE="DB">
               <OBJECT_INFO NAME="pjf_event_type_data.sql"
SUBDIR="prj/pjf/db/sql"
SRCDIR="prj/pjf/db/sql" PRODUCTFAMILY="prj" PRODUCT="pjf"
LBA="" APPNAME="" HEADERSTRING="$Header: fusionapps/prj/pjf/db/sql/
pjf_event_type_data.sql"
OUI_COMPONENT="oracle.fusionapps.prj.db" VERSION="st_fusionapp/1"
TRANSLATION_LEVEL="0" />
       </COMPONENT>
       <COMPONENT TYPE="DB">
               <OBJECT_INFO NAME="pjf_event_type_data.sql"
SUBDIR="prj/pjf/db/sql"
PRODUCTFAMILY="prj" PRODUCT="pjf" LBA="" APPNAME=""
HEADERSTRING="$Header: fusionapps/prj/pjf/db/sql/pjf_event_type_data.sql"
OUI_COMPONENT="oracle.fusionapps.prj.db" VERSION="st_fusionapps/1"
TRANSLATION_LEVEL="0" />       </COMPONENT>
</PATCH_OBJECT_MANIFEST>
```

- `uw123456.xml` contains high-level information about the patch and provides the following information:
  - Translation and platform attributes
  - Prerequisite patches
  - Additional bug fixes that are included in the patch
  - Compatibility information for the patch, such as product family and application name

– Type of patch content and attributes, such as the patch driver location and whether manual steps exist

An example of the contents of the `uw123456.xml` file is described below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--PATCHGEN_VERSION:     11.1.1.5.0-->
<!--OPACK_LABEL:          /net/sta.world.com/OPATCH_MAIN_GENERIC.rdd/opatch/
OPack-->
<!--OPACK_VERSION:        null-->
<!--VIEW_LABEL:           FUSIONAPPS_PT.2000.S-->
<!--PATCH_COMMAND:        ant stFullPatchTransaction -
Dtransaction=prj_adflib_db -Dinclude=ALL -Dbugid=123456 -->
<PatchManifest Version="1.0">
<PatchList PatchType="SNOWBALL" Translatable="Y" PartialTranslations="N"
HighAvailability="DERIVE" Merge="N" GUID="1004567" >
       <Patch Number="123456" Language="US" Platform="GENERIC"
GUID="1004567" BaseBug="123456" BaseProductFamily="UNKNOWN"
BaseProduct="UNKNOWN" BaseLBA=""
Description="" />
</PatchList>
<PreReqBugfixList>
</PreReqBugfixList>
<RequiredComponentList>
       <RequiredComponent ID="oracle.fusionapps.prj.deploy"
Version="11.1.1.5.0" />
       <RequiredComponent ID="oracle.fusionapps.prj.db"
Version="11.1.1.5.0" />
</RequiredComponentList>
<BugfixList>
       <Bugfix Number="123456" ProductFamily="" Product="" LBA=""
Description=""/>
</BugfixList>
<Impact>
       <ProductFamilyList>
              <ProductFamily Name="prj">
              <ProductList>
                     <Product Name="pjf">
                     </Product>
                     <Product Name="pjg">
                     <LBAList>
                            <LBA Name="PjgTop"/>
                     </LBAList>
                     </Product>
              </ProductList>
              </ProductFamily>
       </ProductFamilyList>
       <ApplicationList>
              <Application Name="EARProjectsFinancials.ear"/>
       </ApplicationList>
</Impact>
<ContentList>
                     </Product>
                     <Product Name="pjg">
                     <LBAList>
                            <LBA Name="PjgTop"/>
                     </LBAList>
                     </Product>
              </ProductList>
              </ProductFamily>
       </ProductFamilyList>
       <ApplicationList>
```

```
                <Application Name="EARProjectsFinancials.ear"/>
        </ApplicationList>
</Impact>
<ContentList>
        <Content Type="DB" PreApplySteps="N" PostApplySteps="N"
PatchDriver="u123456.drv"
PatchDriverLocation="123456_DB" DataModelChanges="N" SeedDataChanges="N"
PlSqlChanges="N" SQLChanges="Y" FlexChanges="N" LDAPChanges="N"
DataSecurityChanges="N" />
        <Content Type="MW" PreApplySteps="N" PostApplySteps="N"
PatchDriverLocation="123456_MW" />
</ContentList>
</PatchManifest>
```

- `123456_DB`: Contains files related to changes for the database artifacts included in this patch, bundled so that they can be accessed and applied using AutoPatch.

  The following files exist in the `123456_DB` directory:

  – `u123456.drv`: Contains instructions for AutoPatch to make changes to an Oracle Fusion Applications database and is referred to as the patch driver file.

  – Product family directory: Contains the patch content for database artifacts in a form that is readable by AutoPatch.

- `123456_MW`: Contains files related to middleware artifact changes included in this patch, bundled so that they can be accessed and applied using OPatch. The patch content resides under the `files` subdirectory in a form that is readable by OPatch. The patch metadata resides under the `etc` subdirectory.

  The middleware metadata files exist in the following subdirectories:

  – `/etc/config/actions.xml`

    An example of the contents of the `actions.xml` file is described below:

    ```
    <oneoff_actions>
       <oracle.fusionapps.prj.deploy version="11.1.1.5.0" opt_req="R">
           <copy name="AdfPjgTopPublicUi.jar" path="%ORACLE_HOME%/prj/
    deploy/EARProjectsFinancials.ear/EARProjectsFinancials/WEB-INF/lib"
    file_name="prj/deploy
    /EARProjectsFinancials.ear/EARProjectsFinancials/WEB-INF/lib/
    AdfPjgTopPublic
    Ui.jar" file_version="63.0"/>
       </oracle.fusionapps.prj.deploy>
    </oneoff_actions>
    ```

  – `/etc/config/automation.xml`

    An example of the contents of the `automation.xml` file is described below:

    ```
    <automation xmlns="http://oracle.com/schema/opatch/Automation"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://oracle.com/schema
    /opatch/Automation ../../xsd/automation.xsd" opatch-version="11.1.0.6.0"
    deployment-type="fapps" deployment-sub-type="fapps-artifacts">
        <post-patch-application>
            <deploy-action acts-on="SOAComposite">
                <deploy-artifact file-
    name="sca_FinGlCurrencyUserPreferredCurrencyComposite.jar"
     destination-path="%ORACLE_HOME%/fin/deploy" name="FinGlCurrencyUser
    PreferredCurrencyComposite" revision="7_5512345"/>
            </deploy-action>
    ```

```
    </post-patch-application>
</automation>
```

– `/etc/config/checksum.xml`

  An example of the contents of the `checksum.xml` file is described below:

```
</checksum_info>
    <file path="%ORACLE_HOME%/fscm/security/policies/system-jazn-
data.xml" checksum="-1"/>
</checksum_info>
```

– `/etc/config/inventory.xml`

  An example of the contents of the `inventory.xml` file is described below:

```
<oneoff_inventory>
   <opack_version version="11.1.0.6.0"/>
   <patch_id number="123456"/>
   <cannot_autorollback>false</cannot_autorollback>
   <date_of_patch year="2011" month="Feb" day="16" time="10:47:37 hrs"
zone="PST8PDT"/>
   <base_bugs>
       <bug number="123456" description="fusionapps patch"/>
   </base_bugs>
   <required_components>
       <component internal_name="oracle.fusionapps.prj.deploy"
version="11.1.1.5.0" opt_req="R"/>
   </required_components>
   <os_platforms>
       <platform name="Generic Platform 2" id="2000"/>
   </os_platforms>
   <executables></executables>
   <instance_shutdown>false</instance_shutdown>
   <instance_shutdown_message></instance_shutdown_message>
   <online_rac_installable>false</online_rac_installable>
   <run_as_root>false</run_as_root>
   <sql_migrate>false</sql_migrate>
   <wls_prereq_oneoffs></wls_prereq_oneoffs>
   <os_platforms>
       <platform name="Generic Platform 2" id="2000"/>
   </os_platforms>
   <executables></executables>
   <instance_shutdown>false</instance_shutdown>
   <instance_shutdown_message></instance_shutdown_message>
   <online_rac_installable>false</online_rac_installable>
   <run_as_root>false</run_as_root>
   <sql_migrate>false</sql_migrate>
   <wls_prereq_oneoffs></wls_prereq_oneoffs>
   <prereq_oneoffs></prereq_oneoffs>
   <coreq_oneoffs></coreq_oneoffs>
   <overlay_oneoffs></overlay_oneoffs>
   <patch_type value="snowball"/>
   <patch_language value="en"/>
   <product_family value="fusionapps"/>
   <patching_model value="snowball"/>
   <auto>false</auto>
   <translatable>true</translatable>
   <applicable_product/>
   <products></products>
   <update_components></update_components>
</oneoff_inventory>
```

2.3 Oracle Fusion Applications Patching and the Security Model

In Oracle Fusion Applications, credentials used for patching are stored securely based in the Lightweight Directory Access Protocol (LDAP) Credential Store Framework (CSF), where they can be retrieved when required and hidden when starting processes from the command line. Credentials are not stored in any format in the file system or in the database. Users are not prompted for passwords when using command-line utilities. A separate role is not used for patching purposes because all patch administrators log in as the same operating system user to apply patches.

MANDATORY: The patch administrator user must be an owner of the Oracle Fusion Applications Oracle home.

2.3.1 Obtain Credentials

Patch Manager obtains passwords from the CSF based on the following criteria:

- CSF APIs are used to obtain passwords from the CSF
- A combination of a `MAP` and a `KEY` returns the user name, and its corresponding password, in decrypted format

All credentials are securely stored in a wallet that is stored in LDAP. Patch Manager credentials are available under the `oracle.patching` `MAP` name and each credential is identified by a KEY.

2.3.2 Usage of CSF APIs

The patching framework uses CSF APIs to retrieve credentials. It does not pass the credentials at the command line when calling either AutoPatch or OPatch.

2.3.3 No Password Prompts in Interactive Mode

Security can be breached when prompted for a password while invoking patching from the command line. To avoid this situation, Patch Manager uses the Oracle Platform Security Services APIs to fetch passwords from the CSF.

2.3.4 Removal of Credential From Files

Patch Manager uses a defaults file to store the arguments and other information required for a given session, but does not read or write credentials to or from the defaults file. Additionally, Patch Manager does not read or write credentials from restart files or log files.

3
Plan for Patching

This section outlines the following key considerations when planning a patching strategy for an Oracle Fusion Applications environment:

3.1 Patch Plan: Skills and Knowledge Required

A technical team with the skills and knowledge required to complete patching may include the following:

- A Fusion Applications administrator
- A Database administrator
- A Identity Management (IDM) expert

At a minimum, the team responsible for downloading, evaluating, and applying patches must have the following technical knowledge:

- Oracle Fusion Application installation(s) knowledge
- WebLogic and SOA familiarity
- Database knowledge
- Identity Management knowledge
- Relevant operating system knowledge (Linux, etc.)

Additionally, it is highly recommended to form a Patch Advisory Forum comprised of technical subject matter experts and management. The Patch Advisory Forum would act as a gate keeper and would be the only body with the authority to approve patches. Patch requests are presented to the Patch Advisory Forum, which assesses the risk and prioritizes the application of patches. Critical input into the forum typically includes timing estimates, outage planning, and testing requirements obtained from an impact assessment of the patch environment. The Patch Advisory Forum may be different for each environment and would typically include stakeholder representation for the specific environment.

Example 1: When planning to patch a Test environment, then the Test Lead would be a key stakeholder. The patch would be scheduled at a time that will not interfere with ongoing testing activities, and to ensure that the correct teams are notified about the patching window appropriately.

Example 2: When planning to patch a Production environment, then key business stakeholders will be involved in the Patch Advisory Forum. For example, if the patch

plan would affect a payroll run, those stakeholders might request a patching delay until the payroll is completed.

3.2 Ensure The Patching Tools are Up-to-Date

Before installing Technical Patches, ensure that the environment has the latest version of the framework for installing the patches (OPatch). When downloading a Technical Patch Bundle (P4FA), the latest version of OPatch is included. To ensure that OPatch is up to date, perform the following steps:

1. Set the `ORACLE_HOME` to the directory that will be patched. For example:

   ```
   export ORACLE_HOME=/u01/app/idm/products/app/idm
   ```

2. Navigate to the OPatch directory and execute `cd $ORACLE_HOME/OPatch`.
3. Note that the output contains the version of OPatch. For example:

   ```
   OPatch Version: 11.1.0.8.0
   OPatch succeeded
   ```

If a different version of OPatch is required, see the Update OPatch (page 3-3) section.

3.2.1 Download the Latest Version of OPatch From My Oracle Support

MANDATORY: Check the patch Readme for prerequisite tips regarding the necessary OPatch version. Notice in the sample text below, the OPatch release number is given, as well as the bug placeholder number for downloading OPatch versions, and a link to documentation on My Oracle Support.

```
Ensure that the OPatch is 11g Release 11.1.0.8.3 or higher.


Review and download the latest version version available from patch#
6880880


For information about OPatch documentation, including any known issues,
see My Oracle Support Document 224346.1 OPatch documentation list:


https://support.oracle.com/CSP/main/article?
cmd=show&type=NOT&id=224346.1
```

3.2.2 Check the Current OPatch Version

Perform the following steps to validate the current OPatch version:

1. Set the `ORACLE_HOME` to the directory to be patched. For example:

   ```
   export ORACLE_HOME=/u01/app/idm/products/app/idm
   ```

2. Go to the OPatch directory and execute. For example:

   ```
   cd $ORACLE_HOME/OPatch
   ```

3. Check the output for text like the following:

```
OPatch Version: 11.1.0.8.0
OPatch succeeded.
```

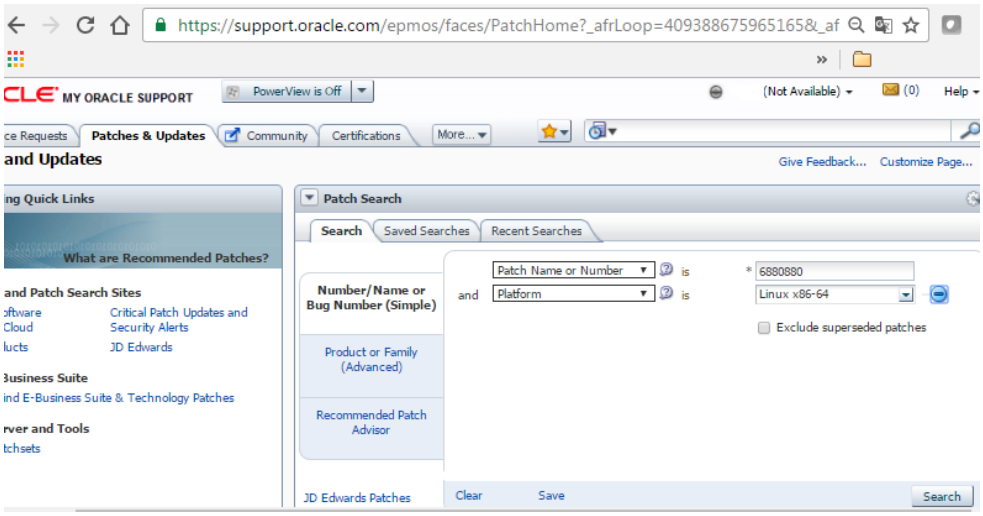
The same steps can be followed for database or applications `ORACLE_HOME` directories.

3.2.3 Update OPatch

Perform the following steps to update the OPatch tool installed on `$ORACLE_HOME` :

1. Go to My Oracle Support (support.oracle.com) and select the **Patches and Updates** tab.
2. Select search by “Patch Name or Number” and enter the number 6880880 in the search field, and choose the operating system as show in the following figure. The same number (6880880) works for all operating systems and OPatch versions.

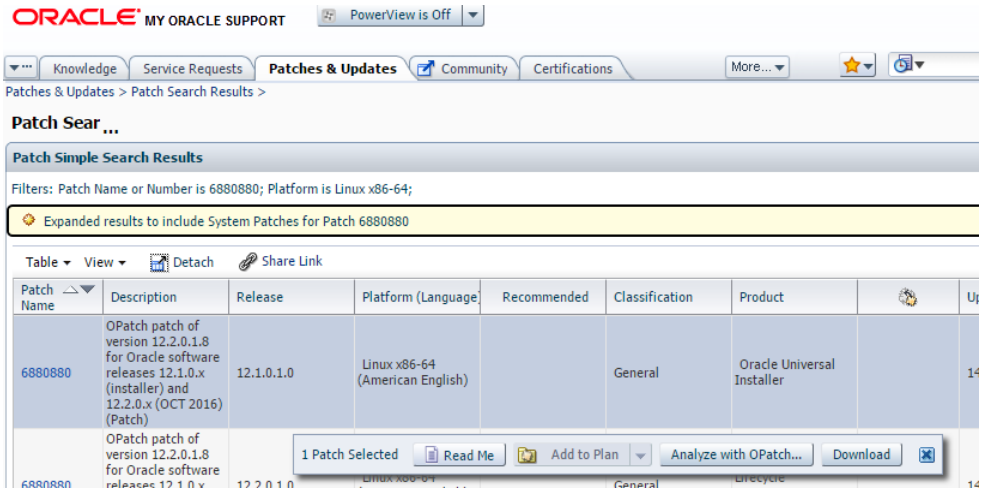
Figure 3-1 Patch Search window



3. Click **Search**.

   The Patch Search result appears as shown in the following figure:

Figure 3-2 Patch Search Results



4. Select the most recent version of OPatch for the relevant release to display Readme and Download options.
5. Click the Read Me to read additional installation instructions.
6. Click **Download** and follow the installation instructions in the Readme text.

3.3 Plan System Backups

MANDATORY: Before applying any patch, a cold (offline) backup of the database and the file system being patched must be performed to ensure data consistency and avoid synchronization problems.

To ensure that no changes are made in the WebLogic Server domains, it is recommended to lock the WebLogic Server configuration for all the domains in Oracle Fusion Applications environment. Best practice also suggests giving a unique name to the backup file, perhaps appending the date and time to the `.tar` file name.

3.4 Plan Impact and Maintenance

This section provides strategies for creating a patching approach that suits the enterprise's needs while adhering to best practices. This section contains the following topics:

3.4.1 Types of Patches For Which to Plan

As mentioned previously, Oracle Fusion Applications includes both technical patches (which affect the underlying Middleware components) and functional patches (which affect the Fusion Applications product families).

Patches are release-specific; patch bundles designated for 11.1.10 cannot be applied to 11.1.9, nor to 11.1.11 versions of Oracle Fusion Applications. Furthermore, functional patches are released per product family. For example, HCM, SCM, and CRM each have their own functional patch bundles for a given release.

Patch planning is focused on functional and technical patches. IDM patching is normally handled either automatically or manually during upgrades, and is discussed in the *Oracle Fusion Applications Upgrade Guide*. There is a rare circumstance in which a patching strategy may be applied to Identity Management, as described in the Apply Identity Management (IDM) Patches (page 7-1) section. Otherwise, IDM patching is not handled in this guide.

3.4.1.1 Time of Patch Bundle Releases

Since multiple patch bundles for technical and functional side are published for each release, it is recommended to establish a cadence that minimizes the amount of downtime taken to apply the various patches while maximizing the uptake of key patches for the organization.

For example, a particular HCM bundle may contain a critical fix but will not be released for another few weeks. The HCM AOO patch may be applied to a test environment to allow for completion of a critical project milestone while choosing to wait for the functional patch bundle for the production environment. Another facet of the decision could be that a particular organization may not allow for a maintenance window every month, but instead every other month. Given that both the functional and the technical patch bundles are cumulative, patching every other month still ensures a pro-active approach while minimizing the needed maintenance windows. A six-month patching cycle is an example of too long a cadence while a weekly cycle may prove too short a cadence. Determining the right frequency is a combination of the organization's maintenance windows and the need to consume a particular fix.

This concept is best summarized in the following table:

Table 3-1 When to Apply Oracle Fusion Applications Patches

Type	One-Offs	AOOs	CPU	Patch Bundles
Technical	Apply only to address critical issue, otherwise wait for patch bundle	Not applicable	Apply immediately upon release	Ideally apply monthly. Less frequently than every other month is not recommended

Table 3-1 (Cont.) When to Apply Oracle Fusion Applications Patches

Type	One-Offs	AOOs	CPU	Patch Bundles
Functional	Apply only to address critical issue, otherwise wait for update bundle	Apply when project timelines or urgency of fix do not alight with release of patch bundle, otherwise wait for update bundle	Not applicable	Ideally apply monthly. Less frequently than every other month is not recommended

3.4.1.2 Functional Patching Modes and How They Affect Outage Windows

All technical patches, whether one-off, CPU, or P4FA bundles, require that the system be taken offline for the patching process. This comprises a true “outage window.”

For functional patches, Patch Manager supports three different modes in which patches might be applied: offline, online, or hot patching. Some functional patches and patch bundles support one mode, others require another. The appropriate mode can be found using the Patch Manager *validate* option. Each mode has a different effect on the extent and timing of an outage window.

Each of the three modes is described in detail in the Patch Modes (page 2-3) section.

3.4.2 Impact Assessment Strategies

The following are some of the tools and strategies for assessing the effect that patching will have on a production environment:

- **READMEs**: The Patch READMEs describe the bug fixes and system areas addressed by the patch.
- **Test Environments**: Applying the patch on a test or patching environment and performing subsequent regression tests will help determine the timing and impact of a patch on a given Oracle Fusion Applications installation, and permit targeted planning for patching subsequent environments.
- **Reports**: The Patch Impact Report is available for functional patches. For more information, see the Patch Impact Report (page 3-6) section.

3.4.2.1 Patch Impact Report

The Patch Impact report compares the contents of the patch to be applied with the files that currently exist on the system. The report shows a complete picture of what file system changes will occur when the patch is applied. Plan the system downtime by viewing the servers that will be affected by the patch, along with any manual deployment actions that are required after the patch is applied. This report reads the patch metadata, local patch inventory, and the current view snapshot.

The Patch Impact report displays the impact information about a patch in the following section:

Bug Fixes

This section provides the following information about the bug fixes included in the patch:

- Bug Number: The number of the bug fix or patch
- Bug Description: The description of the bug fix or patch
- Exists in Oracle home: Whether the bug fix or patch was already applied (Yes or No)

Prerequisite Bug Fixes

This section provides the following information about patches that must be applied before the current patch can be applied:

- Bug Number: The number of the prerequisite bug fix or patch
- Bug Description: The description of the prerequisite bug fix or patch
- Exists in Oracle home: Whether the prerequisite bug fix or patch was already applied (Yes or No)

Prerequisite Bug Fixes Not in FA_ORACLE_HOME

This section provides the following information about patches that must be applied before the current patch can be applied. These patches are not applied to FA_ORACLE_HOME.

- Bug Number: The number of the prerequisite bug fix or patch
- Bug Description: The description of the prerequisite bug fix or patch
- Exists in Oracle home: Whether the prerequisite bug fix or patch was already applied (No)

Product Families Impacted

This section provides the following information about which product families are impacted by the patch:

- Product Family: The name of the product family (component) that is updated by the patch
- Product: The name of the product that is updated by the patch
- LBA: The logical business area that is updated by the patch

Servers Impacted

This section provides the following information about which servers will be impacted by the patch. Note that all artifacts in the patch are copied, but server life cycle actions occur only for those product families that have been deployed during the provisioning process.

- Artifact Type: The type and name of the artifact included in the patch
- Domain (Servers): The servers that are affected by the artifacts in the patch
- Expectation/Impact: The description of what servers must be running, what actions will be taken during the patch apply phase by Patch Manager, and what manual actions must be taken

Files Included in the Patch

This section provides the following information about the files that are included in the patch:

- File Name: The name of the file
- File Type: The type of the file
- File Version: The version of the file

For detailed information about the Patch Impact Report parameters, see the Patch Impact Report Parameter Details (page 3-8) section.

3.4.2.1.1 Patch Impact Report Parameter Details

The following table describes the parameters used by the Patch Impact report:

Table 3-2 Parameters Used by the Patch Impact Report

Parameter	Mandatory	Description
`patchtop`	Yes	Identifies the directory where the patch is unzipped
`outputfile`	No	Sends the report output to the specified file after this parameter. An existing file name cannot be used. If this parameter is not used, no output file is created
`logfile`	No	Overrides the default log file name and sends the processing information to the specified file, under the *APPLICATIONS_CONFIG/lcm/logs/<Fusion Applications Release Version>/FAPMGR* directory. If an existing file name is entered, the output is appended to the file. If this parameter is not used, the utility generates a log file under *APPLICATIONS_CONFIG/lcm/logs/<Fusion Applications Release Version>/FAPMGR* using this naming convention: `FAPatchManager_report-patchimpact_`*timestamp*`.log`
`loglevel`	No	Records messages in the log file at the specified level. See the Oracle Fusion Applications Patch Manager Logging (page 8-1) section
`reportwidth`	No	Sets the column width to either 80 columns or 132 columns by specifying `NORMAL` or `WIDE`. The default value is 80 columns, or `NORMAL`

3.4.2.2 Run the Patch Impact Report

The Patch Impact report can be run when applying only one patch or multiple patches downloaded in a patch plan. Before running the Patch Impact Report, ensure that the snapshot is current for the environment.

Use the following syntax to run the Patch Impact report for a single patch:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -patchimpact -patchtop
 path_to_unzipped_patch [optional parameters]
```

Use the following syntax to run the Patch Impact report for a single patch:

```
(UNIX)FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -patchimpact -
grouptoptop_directory_for_patches -patchingplanfull_path_to_patching_plan
[optional parameters]
```

3.4.3 Create a Patch Tracking Sheet

It is important to document the patching plan and outcomes, to enhance team coordination and knowledge transfer, and to be used for future reference in system maintenance. Every organization creates its own documentation standards. A Patch Tracking Sheet is one way of centralizing information about the patches applied to a Fusion Applications environment. In general, the tracking sheet can be generated in almost any word editor or spreadsheet software, or even as a hard copy on paper. The basic and most minimal information which should be tracked and retained as follows:

- Requestor of the patch
- Patch Number
- Description of the patch
- Date Applied
- Target Environment
- Time Taken to Apply
- Issues Experienced
- Special Instructions
- Comments
- Oracle Service Request # (if applicable)

3.4.4 Summary: Principles for Scheduling Maintenance

- **Stay Current**: Stay as current as possible with patching. Patches are cumulative, so if they are delayed, the system downtime for installation can expand considerably. On the other hand, patching disrupts normal system activities and patch releases can be frequent, so discernment is required. The best practice is to establish a cadence that minimizes the amount of downtime taken to apply the various patches while maximizing the uptake of key patches for the organization.
- **Patch in Order**: Technical patches (P4FA) are always installed before functional patches. Patches are release-specific, cumulative, and sequential.
- **Plan Outage Windows**: Technical patching always requires that users exit the system, tasks and jobs be stopped, and the servers be taken offline. This comprises a true outage window. Functional patches can sometimes be applied as hot patches, in which no servers are shut down, users may remain in the system, and there is a defined time frame to allow background processes and jobs to complete before the patch is launched. In these cases, the “outage” is minimal. Some functional patches inhabit a grey area of online patching, in which servers remain up, but no users, transactions, or tasks are allowed. Have a clear understanding of which patching mode will be in use for any given patching session.
- **Use a patching test environment**: At a minimum, test patches on a non-production system to determine the necessary timing, analyze the patching effects and perform regression testing before applying the patch to a production system. Some organizations, in addition to *test*, *dev*, and *prod* environments, also establish a dedicated *patch* environment.

- **Ensure communication**: Establish an advisory forum with stakeholders and schedule notifications with all affected parties.
- **Document the plan**: Use tracking sheets and other systems to ensure that the patching team is coordinated and that patches are documented for future reference.

3.5 Plan Test for Applied Patches

Patch application follows a progressive environment path. Although every company is different, the usual path consists of Development, Test , QA, and Prod.

Development: The goal during the Development stage is to perform unit testing of the patch. The following is an example of a Development stage procedure:

1. An issue is identified and a Service Request (SR) is filed.
2. Oracle Support recommends to apply a patch.
3. The users are able to reproduce the issue in the Development environment.
4. The patch is applied to the Development environment.
5. The issue cannot be reproduced in the Development environment anymore.
6. If the issue is still present, the patch is discarded and the SR is updated accordingly.

Test: The goal during the Test stage is to perform integration testing of the patch. Below is an example of a Test stage procedure:

1. After unit testing is successful, the patch is applied to the Test environment.
2. The users test the functionality of all other components affected by the patch payload, not only the issue at hand. Normally, there is documentation describing all the tests needed and the expected output.

QA: The goal during the QA stage is to provide quality assurance testing also known as user acceptance testing. Quality assurance or user acceptance testing is a final test to verify whether the patch is approved to be applied to production. Below is an example of a QA stage procedure:

1. After integration testing is successful, the patch is applied to the QA environment.
2. The key users will execute and sign-off on a standardized set of tests to be sure the patch is not breaking the functionality of the system. This set of tests is the same or an updated version of the tests used when the system was deployed initially. This test set is also known as regression test and can be applied manually or automatically.

4

Apply Technical Patch Bundles: P4FA

This section describes how to apply a Technical patch bundle (P4FA) to the Oracle Fusion Applications environment using FASPOT.

The concepts and tasks related to applying P4FA are as follows:

4.1 Anatomy of a Technical Patch Bundle (P4FA)

Patches for Fusion Applications (P4FA) are collections of one-off fixes and tech stack updates (e.g. Fusion MiddleWare, Database, Weblogic Server, etc.) compiled and certified for Fusion Applications where an installation of them improves system stability and performance substantially.

MANDATORY: Every Fusion Application release has its own set of P4FA patches and it is mandatory to apply them. P4FA patches are cumulative and administrators must always apply the most recent patch after becoming available on Oracle Support.

To ensure the successful and efficient application of a patch bundle, it is important to understand the directory structure. The procedure for understanding the directory structure is as follows:

- After downloading the latest patch from My Oracle Support and unzipping it, the first component in the directory is the Patch Number, this changes with every new release of the patch. A new number gets assigned to the latest version of the P4FA Patch.
- The second component under the actual Patch Number is the README File which contains the instructions to a successful application of a Patch Bundle.
- Further down in the directory, `/patch number/patches4fa/dist/` contains the following sub-components:
  - **Prepatch** provides some patches that need to be applied before applying the actual P4FA patch. Prepatches can be normally taken care of by FASPOT, unless they are Database patches, in which case, it remains a manual task for DBAs, per example:
    * The **Exadata** patches are applied for an specific type of hardware and they can be found on the`/Exadata/` subdirectory.

    - The **Database** patches can be identified within the subdirectory as `/RDBMS/database version/` , and to facilitate the manual process of applying the patches, start the application with the PSU patches, and then continue applying the rest of the One-Offs.
  - **FASPOT** is a utility used to automate the application of Oracle Fusion Middleware Patches.
  - **Release Directory** is the number that contains the rest of the patches normally applied by FASPOT.

The following directories and subdirectories should exist in the P4FA patch structure:

```
|- Patch Number
    |- README.txt
        |- /patches4fa/dist
            |- /Prepatch
                    |- /RDBMS/database version
                        |- /PSU
                        |- /Exadata
                    |- /generic/one_off_patches
            |- /FASPOT
            |- /Release Directory
```

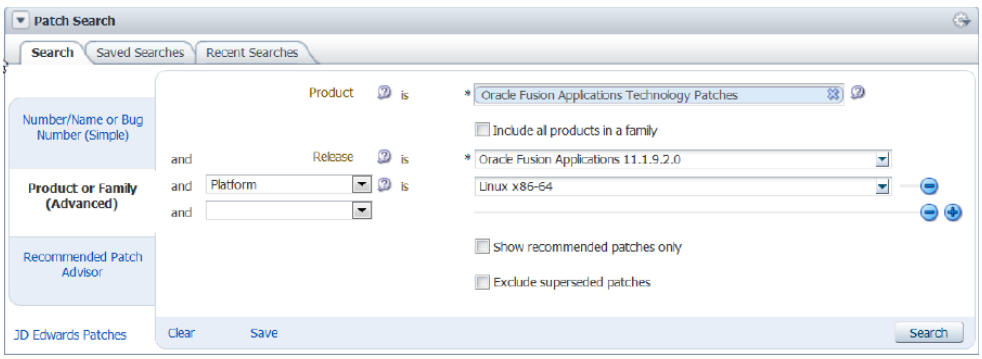
4.2 Download a Technical Patch Bundle (P4FA)

Download and unzip P4FA patches on a shared file system that is accessible by all hosts for Fusion Applications, IDM and OHS. To find patches in My Oracle Support (MOS), search using the standard naming conventions for Technical Patch Bundles. Technical Patch Bundles follow the naming convention of "P4FA System Patch *Release.yymmdd*". For example, a technical patch bundle released for Release 9 on March 31, 2015 is named "P4FA System Patch 11.1.9.150315".

Perform the following steps to find the latest technical patch bundle (P4FA):

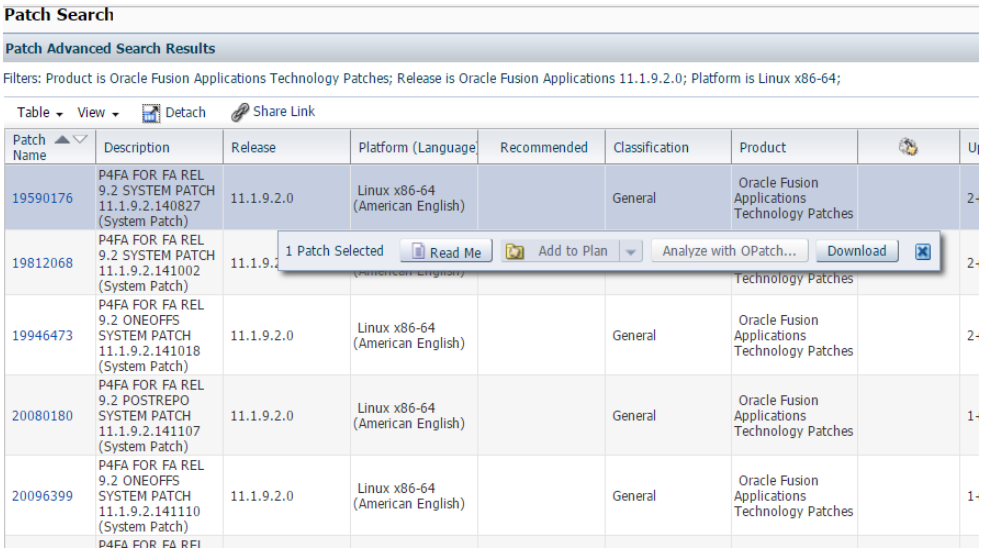
1. Log in to My Oracle Support (MOS) and navigate to “Patches and Updates”.
2. On the Patch Search panel choose “Product or Family (Advanced)” and type “Oracle Fusion Applications Technology Patches” in the drop down list labeled "Product".
3. Select the appropriate release for the P4FA patch.
4. Select the appropriate platform for the P4FA patch.
5. Click **Search**.

Figure 4-1 Patch Search Tab



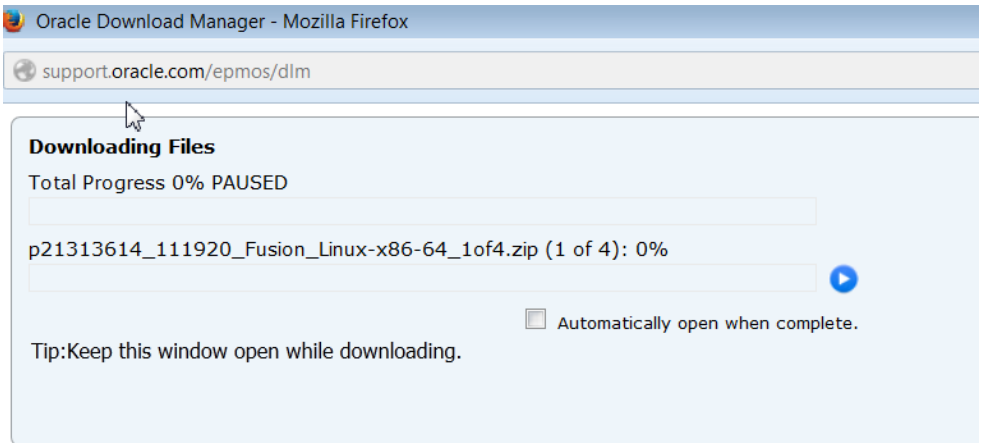
6. The search results may include a CPU technical patch. In the Search Results screen, find the patches with the P4FA naming convention and choose the most recent one.

Figure 4-2 Patch Advanced Search Results



7. Click **Download**.

Figure 4-3 Download Manager for Patch Bundle P4FA



8. Wait for successful completion of the download.
9. Unzip the downloaded zip files into the *PATCH_DOWNLOAD_DIR* directory.

4.3 What is FASPOT?

A P4FA contains several individual patches, which can result in complex and time consuming installations. FASPOT automates the P4FA patch installation, and its scripts are bundled with the P4FA patches. FASPOT is used to simplify the Fusion Applications tech stack patching process. Since P4FA patches are cumulative,

subsequent patches will also include updates from previous patches although they only need to be applied once.

FASPOT orchestrates the installation of the P4FA using *ant* and provides several targets. FASPOT requires preparation tasks, as described in the Prepare to Apply a P4FA With FASPOT (page 4-4) section. After the preparation steps are complete, the patching installation process is performed by an *ant* script.

MANDATORY: FASPOT must run once on every server that hosts a component of the Fusion Applications tech stack.

FASPOT applies the patches to the following categories:

- Fusion Middleware Components (including atgpf, ODI, oracle_common, SOA, webtier, BI, ECM, SES, WebCenter, wls)
- Identity and Access Management (including OID & OHS)

The installation of RDBMS patches are not within the scope of FASPOT and remains a manual task for DBAs. These are out of scope for FASPOT. Number of combinations and variants of existing DB topologies would make it too complex to be automated by scripts. Usually DBA's use their own mechanisms and best practices to perform DB life cycle management.

4.4 Overview of Installing a P4FA With FASPOT

The order of tasks during the installation of P4FA using FASPOT is critical because certain services must be either down or running depending on which patches are being applied. An overview of these tasks are as follows:

1. Download and extract the P4FA patch, preferably on a shared file system accessible by all hosts for FA, IDM and OHS.
2. Complete the preparation steps as described in the Prepare to Apply a P4FA With FASPOT (page 4-4) section.
3. Prepare the staging.
4. Complete RDBMS P4FA patching manually for IDM and FA databases.
5. Start both IDM and FA databases and their services.
6. Run the IDM and FMW patch-apply targets.
7. Start up IDM and FA completely.
8. Run the IDM and FMW post-install targets to deploy various artifacts.
9. Validate the results by performing functional tests.

For more information on how to download and extract the these patches, know the preparation steps, prepare the staging area, and run de IDM and FMW path-apply targets, review the information below in this section.

4.5 Prepare to Apply a P4FA With FASPOT

Ensure the FA and IDM databases have been patched, including the related post installation steps. Then perform the following preparation steps before applying a P4FA patch:

1. Verify disk space availability to support a full installation of P4FA.
2. Unzip the *PATCH_DOWNLOAD_DIR/* FASPOT directory. This will now be referred to as *FASPOT_HOME_DIR*.
3. Verify the Contents of *FASPOT_HOME_DIR*. The following files and directories should exist in *FASPOT_HOME_DIR*:

```
|-   README.txt
|-   build.xml
|-   config
|- …
|-   env
|-   faspot.properties
|-   …
|-   faspot.sh
|-   patch_log_dir
|- …
|-   setEnv.sh
|-   tools
|- …
```

   The meaning of files and directories mentioned in this guide is as follows:

   - **build.xml**: The script that contains all targets and actions to perform while patching in a Fusion Applications environment.
   - **README.txt**: The documentation for each patch.
   - **faspot.sh**: The script that handles the ant calls for different targets, and is called to execute the ant targets.
   - **setEnv.sh**: The script to set the environment for the FASPOT utility execution.
   - **env**: The directory that contains property files that must be updated before script execution.
   - **env/faspot.properties**: Contains details about installed components and the patch download location.
   - **env/faspot.properties.template**: The sample template file used to populate the values and generate *faspot.properties* .
   - **config**: Contains product specific configuration files and templates that are generated during runtime.
   - **tools**: Contains ant libraries that are used for setting the *ANT_HOME* in setEnv.sh.
   - **relnumoneoffs**: Includes a subdirectory for each type of tech stack. For example, pfcore, weblogic, odi, and idm.
   - **prepatch**: Includes a subdirectory for each type of tech stack. For example, atgpf, ecm, odi, and soa.
4. Create the two directories used by FASPOT as follows:
   - *PATCH_WORK_DIR*: This directory should be an external directory to be used during runtime by all pods and hosts. It will contain all of the extracted P4FA patches organized in the format required by the FASPOT script.
   - *PATCH_LOG_DIR*: This directory will be used to store logs generated during execution of the FASPOT script.
5. Edit the `faspot.properties` file as follows:

The `faspot.properties` file contains information about the Fusion Applications installation such as host names, directory names, user names, and passwords. This file should be edited for each specific P4FA patch application.

a. Copy the properties template file `FASPOT_HOME_DIR/env/faspot.properties.template` to `FASPOT_HOME_DIR/nv/faspot.properties` so the new file can be edited. Before editing it, make a copy of `FASPOT_HOME_DIR/env/faspot.properties`

b. Edit both variables for the P4FA download directory and the patch staging directory, as shown in the following example:

```
PATCH_DOWNLOAD_DIR=P4FA_PATCH_EXTRACTION_DIR/patches4fa
PATCH_WORK_DIR=any_directory_to_run_patches_from
```

c. Update the remaining properties in `env/faspot.properties`. As a general rule, replace all placeholder `%<value>%`with values that are specific to the environment. The following example of the `faspot.properties` file shows sample values for each property:

```
#############GENERAL#############################################
#PATCH_DOWNLOAD_DIR should contain the following folders
### 1)reloneoffs
################################################################
PATCH_DOWNLOAD_DIR=/u01/fastage/p4fa/19290105/patches4fa/dist
PATCH_WORK_DIR=/u01/fastage/p4fa/19290105/patches4fa/workdir
################################################################
############FA(FMW COPONENTS)#################################
#atgpf - ATGPF
################################################################
ATGPF_HOST_NAME=fusionhost.mycompany.com
ATGPF_ORACLE_HOME=/u01/app/fa/fusionapps/atgpf
ATGPF_JDK_LOC=/u01/app/fa/fusionapps/jdk6
ATGPF_DEFAULTS_FILE=/u01/app/fa/config/atgpf/admin/defaults.txt
ATGPF_ADPATCH_WORKERS=4
ATGPF_FA_DB_SID=slc01hye
ATGPF_FA_DB_USER=fusion
ATGPF_FA_DB_PASSWD=Welcome1
ATGPF_FUSION_ORA_ESS_USER=fusion_ora_ess
ATGPF_FUSION_ORA_ESS_PASSWORD=Welcome1
ATGPF_FUSION_RUNTIME_USER=fusion_runtime
ATGPF_FUSION_RUNTIME_PASSWORD=Welcome1
ATGPF_SYSTEM_PASSWORD=Welcome1
ATGPF_SYS_PASSWORD=Welcome1
ATGPF_MIDDLEWARE_HOME=/u01/app/fa/fusionapps
ATGPF_WEBLOGIC_HOME=/u01/app/fa/fusionapps/wlserver_10.3
################################################################
#bi - BI
################################################################
#P4FA patching is done from Primoridal host.
#Enter PRIMORDIAL_HOST_NAME for BI_HOST_NAME
BI_HOST_NAME=fusionhost.mycompany.com
BI_ORACLE_HOME=/u01/app/fa/fusionapps/bi
BI_JDK_LOC=/u01/app/fa/fusionapps/jdk6prepare-patch-stage
BI_WL_ADMIN_USER=FAAdmin
```

```
BI_WL_ADMIN_PASSWD=Welcome1
BI_WL_ADMIN_URL=t3://fusionhost.mycompany.com:10201
```

MANDATORY: The previous example does not include all of the sections in the `faspot.properties` file which must be edited. The remaining sections that must be edited are as follows:

- ECM
- ODI
- OSN
- ORACLE COMMON – FA(FMW)
- SES
- SOA
- PFCORE
- WEBCENTER
- WEBTIER-ADMIN
- WEBTIER-APPSOHS
- WEBLOGIC
- DATABASE
- OHS – IDM

4.6 Prepare the Patch Staging Area

To prepare the patch staging area, perform the following steps:

1. Create the staging area by running the `prepare-patch-stage` target, which extracts the P4FA patches from `PATCH_DOWNLOAD_DIR` into the directory that is specified by the `PATCH_WORK_DIR` property in the `faspot.properties` file. To run the `prepare-patch-stage`, as follows:

   ```
   cd FASPOT_HOME_DIR
   sh ./faspot.sh -Dlogfile=prepare-patch-stage prepare-patch-stage >
   prepare-patch-stage-stdout.log 2>&1
   ```

2. Prepare the local environment by creating the required configuration files under the `FASPOT_HOME_DIR/config` directory. All required instance information needed by other patch application targets is derived from these configuration files. Run the following command from the `FASPOT_HOME_DIR` directory:

   ```
   cd FASPOT_HOME_DIR
   sh ./faspot.sh -Dlogfile=prepare-local-env prepare-local-env >
   prepare-local-env-stdout.log 2>&1
   ```

   If a failure occurs while running `prepare-local-env` that requires an update to `faspot.properties`, run `prepare-local-env` again. The instance specific configuration files under `FASPOT_HOME_DIR/config` are generated by the `prepare-local-env` target, so any subsequent changes in `faspot.properties` have no impact on the execution of FASPOT targets until `prepare-local-env` runs successfully after the updates to `faspot.properties`.

3. Confirm the primordial host and APPOSH hosts are ready for the P4FA application by running the environment checking target. This target checks the existence of prerequisite patches, the directory paths that are specified in the properties file, connectivity to the databases, and any locks that may prevent patching. Run the following command from the *FASPOT_HOME_DIR* directory:

   ```
   cd FASPOT_HOME_DIR
   sh ./faspot.sh -Dlogfile=fmw-prereq-check fmw-prereq-check > fmw-prereq-check-stdout.log 2>&1
   ```

4. Verify the log file after target execution for any failures. In the case of any wrong values provided in `faspot.properties`, change the values and rerun `prepare-local-env` as explained in Step 1. Repeat the `fmw-prereq-check` target execution until all failures are resolved.

5. Confirm the AUTHOHS, OAM, and OID hosts are ready for the P4FA application by running the environment checking target. Run the following command from the *FASPOT_HOME_DIR* directory:

   ```
   cd FASPOT_HOME_DIR
   sh ./faspot.sh -Dlogfile=idm-prereq-check idm-prereq-check > idm-prereq-check-stdout.log 2>&1
   ```

6. Verify the log file after target execution for any failures. In the case of any wrong values provided in `faspot.properties`, change the values and rerun `prepare-local-env` as explained in Step 1. Repeat the `idm-prereq-check` target execution until all failures are resolved.

4.7 Apply Patches with FASPOT

Ensure all preparation steps were successful before applying patches with FASPOT. The generic syntax for applying patches is as follows:

```
cd FASPOT_HOME_DIR
sh ./faspot.sh -Dlogfile=<target-name> <target-name> > <target-name>-stdout.log 2>&1
```

The following are the existing standard targets, which must be applied in the correct order and on the correct host:

1. On the IDM host, run `idm-patch-apply`.
2. On the FA host, run `fmw-patch-apply`.
3. On the FA host, run `fmw-patch-postinstall`.
4. On the IDM host, run `idm-patch-postinstall`.

4.7.1 Run the IDM and FMW `patch-apply` Targets

Perform the following in order to apply the P4FA patches:

1. Run `idm-patch-apply` on the IDM host by performing the following:

   ```
   cd FASPOT_HOME_DIR
   sh ./faspot.sh -Dlogfile=idm-patch-apply idm-patch-apply > idm-patch-apply-stdout.log 2>&1
   ```

2. Run `fmw-patch-apply` on the FA host by performing the following:

```
cd FASPOT_HOME_DIR
sh ./faspot.sh -Dlogfile=idm-patch-postinstall patch-postinstall >
patch-postinstall-stdout.log 2>&1
```

3. Start the IDM and Fusion Applications Databases

4.7.2 Perform the IDM and FMW Post-Install Tasks

Perform the following tasks to apply the P4FA patches:

1. Run `fmw-patch-postinstall` on the FA host by performing the following:

```
cd FASPOT_HOME_DIR
sh ./faspot.sh -Dlogfile=idm-patch-postinstall patch-postinstall >
patch-postinstall-stdout.log 2>&1
```

2. Run `idm-patch-postinstall` on the IDM host by performing the following:

```
cd FASPOT_HOME_DIR
sh ./faspot.sh -Dlogfile=idm-patch-postinstall patch-postinstall >
patch-postinstall-stdout.log 2>&1
```

4.7.3 Validate Results with Functional Tests

It is necessary to perform various functional tests in order to ensure that the patches have been applied correctly and that the Fusion Applications environment is performing as intended. Validation tests vary by specific business needs, but typical tests include such tasks as logging in to the home page, submitting an ESS job to verify the scheduler is working, and checking the status of various services.

4.8 Verify Whether FASPOT Was Successful

- If FASPOT was successful, then the process ended.
- If FASPOT was not successful, then see the Troubleshoot Patching Sessions for FASPOT (page 8-27) section.

5

Apply Functional Patch Bundle

This section describes how to apply patch bundles to the Oracle Fusion Applications environment in an efficient manner.
The following topics are discussed:

If the process of applying a Functional Patch Bundle is understood, see the Download the Latest Patch Bundle and Its Prerequisites (page 5-2) section.

5.1 Anatomy of a Functional Patch Bundle

To ensure the successful and efficient application of a patch bundle, review all requirements described in the README file. A patch bundle README file typically describes the following information:

- Prerequisites for the patch bundle, which may include patches that must be applied or steps to be followed.

  MANDATORY: Ensure that the latest P4FA has been applied before applying a patch bundle, even if it is not mentioned in the patch bundle README file.
- The actual steps to apply the patch bundle.
- Required post-installation patches for the patch bundle.

The structure of the README file may vary, depending on the product family.

For example, the Oracle Fusion Financials Patch Bundle README file contains the following sections:

- Prerequisites
- How to validate and apply this update
- Post-installation
- Functional documentation, including a reference to the functional README document and the Patch Payload

In contrast, the Oracle Fusion Procurement Patch Bundle README file contains the following sections:

- Prerequisite Patches

- Corequisite Patches
- Pre-Installation Steps
- Post-Installation Steps
- Included Bug Fixes, including the Patch Payload for each bug fix

Furthermore, the Oracle Fusion Human Capital Management (HCM) Patch Bundle README file contains the following sections:

- Prerequisite Patches, which refers to Pre-Installation Steps
- Corequisite Patches
- Pre-Installation Steps

  MANDATORY: The HCM bundle always requires that the Business Intelligence (BI) bundles be applied as a prerequisite.
- Post-Installation Steps

5.2 Release Cadence of Functional Patch Bundles

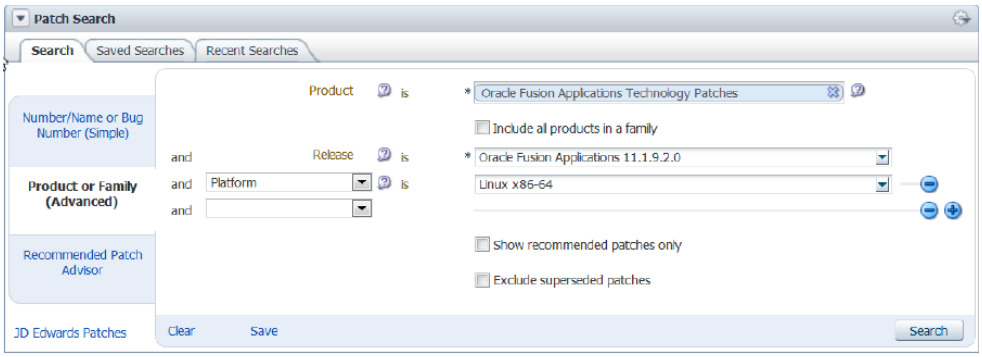
Fusion Applications patch bundles are typically released on a monthly basis, by each functional product family, for a specific release. HCM releases patch bundles for Release 12 (11.12.x.0.0), as does Financials and CRM, for example. All patch bundles are cumulative and Oracle recommends that the bundles are installed as soon as possible.

5.3 Download the Latest Patch Bundle and Its Prerequisites

Perform the following steps to download the latest patch bundle and any prerequisites required by the patch bundle:

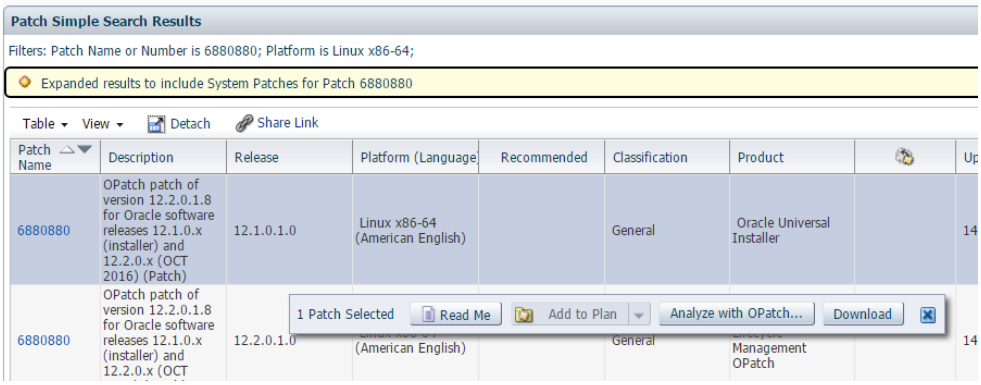
1. Create a directory for storing the downloaded patches, such as `/tmp/finbundle`, in this example.
2. Log in to My Oracle Support.
3. Navigate to "Patches and Updates".
4. On the Patch Search panel, select "Product or Family (Advanced)".
5. Enter "Oracle Fusion *NAME*" where *NAME* is the name of the product family, such as Oracle Fusion *Financials Operations.* Optionally choose to exclude superseded patches.
6. Click **Search.**

Figure 5-1 Patch Bundle search window



7. In the Search Results screen, find the most recent patch bundle with the naming convention such as "FA FIN PATCH BUNDLE 7 11.1.9.2.150610". The search results may also include AOO functional patches.

Figure 5-2 Patch Bundle Search Results



8. Click the patch number, which is in the Patch Name column, and then select **Read Me** to open the patch bundle Readme file. Check if any prerequisite patches are required. Information about downloading prerequisite patches will be in the README file for the patch bundle.
9. Click Download to download the patch bundle zip file to the directory created for this bundle, for example, `/tmp/finbundle`.
10. Download the prerequisite patches and any additional patches which are required as indicated in the respective README files.

5.4 Apply All Patches Related to a Functional Patch Bundle

Create a patch plan to apply all of the prerequisite patches, the patch bundle and any post-installation patches. Using a patch plan significantly reduces the time required to apply the patch bundle because it allows the patching framework to minimize the number of environment bounces during the application of the patch bundle, its prerequisites and post-installation patches.

In addition to applying patch bundles, the steps in this section can also be followed for applying multiple one-off patches in a single patching session, outside of a patch bundle.

Perform the following steps to create and apply a patch plan:

1. Set the PATH environment variable to:

   ```
   /u01/APPLTOP/dbclient/perl/bin:$PATH
   ```

2. Set the PERL5LIB environment variable to:

   APPLICATIONS_BASE/dbclient/perl/lib/5.8.3:*APPLICATIONS_BASE*/dbclient/perl/lib/site_perl/5.8.3/:*APPLICATIONS_BASE*/dbclient/perl/lib/site_perl

3. Use the following command syntax to create the patch plan file:

   $*APPLICATIONS_BASE*/dbclient/perl/bin/perl *APPLICATIONS_BASE*/fusionapps/applications/lcm/ad/bin/adGenerateFAPatchPlan.pl -grouptop *download_location*

   The *download_location* is the location where the patches were downloaded.

   An excerpt from a sample patch plan follows:

   ```
   - <fapatchexecplan>
   <generated_date>20130531</generated_date>
   <fapatchutilversion>1.1</fapatchutilversion>
   - <group_list>
   - <group>
   - <patch>
    <id>33001</id>
    <description />
   <artifact_type>BIP</artifact_type>
   <language>US</language>
    </patch>
    </group>
   - <group>
   - <patch>
   <id>9912345</id>
    <description />
    <artifact_type>SOA</artifact_type>
   <language>US</language>
   </patch>
   </group>
    </group_list>
    </fapatchexecplan>
   ```

   Run the `fapmgr validate` command to find any potential issues that could result in failure while applying the patches. Resolve all validation failures during this step, to prevent failures during patch application. This validation step is strongly recommended, especially for patches that deliver artifacts whose deployment is automated by Patch Manager, such as SOA composites.

   Patch validation runs again by default when the patches are applied. An example of the validate command follows:

```
FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh validate -grouptop
path_to_unzipped_patches-patchingplan path_to_patch_plan_xml_file
```

4. Use the following checklist before patching the target environment:
   a. Ensure all users are off the system.
   b. Set the environment variable for the *APPLICATIONS_BASE* and *FA_ORACLE_HOME* directories.
   c. Run Health Checker to perform the Patching Readiness Health Checks and the General System Health Checks, as shown in the following example:

      ```
      FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest
      FA_ORACLE_HOME/lcm/hc/config/PatchingReadinessHealthChecks.xml [-
      DlogLevel=log_level]

      FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest
      FA_ORACLE_HOME/lcm/hc/config/GeneralSystemHealthChecks.xml [-
      DlogLevel=log_level]
      ```

   d. Manually shut down the Oracle Enterprise Scheduler Service (ESS) servers, especially when a patch contains a PL/SQL package, by performing the following steps:
      i. Stop the Oracle Enterprise Scheduler request processor and dispatcher to prevent new requests from being processed.
      ii. Cancel any in-progress requests.
      iii. Shutdown the Oracle Enterprise Scheduler WebLogic Server Managed server.
   e. **CONDITIONAL**: If a patch contains BI Publisher reports, ensure that the versions of any customized BI Publisher reports are available. If a patch includes an update to a catalog object that was delivered with an Oracle Fusion application, the patch will overwrite any customizations applied to the original report.
   f. **CONDITIONAL**: If a patch contains BI Publisher artifacts, the BI OPMN control process, which is similar to a node manager, must be running for online mode validation to succeed.
5. Apply the patch plan using the fapmgr `apply` command as shown in the following example:

   ```
   FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh apply -grouptop
   path_to_unzipped_patches -patchingplan path_to_patch_plan_xml_file -
   online -stoponerror [-workers number_of_database_workers]
   ```

   Monitor the progress of the patching session and verify its successful completion by reviewing the Log Summary from a browser. If the patching session completes successfully, proceed to the next step.
6. Review the Diagnostics report to see if any manual steps are required for the artifacts included in the patch.
7. Review the Diagnostics report to determine if Managed Servers require restart. If any servers must be restarted manually, the report provides the target domain and the names of the cluster and Managed Servers that must be restarted.

   CONDITIONAL: If the patch contains JAZN artifacts, resolve any conflicts by reviewing the log files.

8. Run Health Checker to perform the Post Patching Health Checks and the General System Health Checks, as shown in the following example:

   ```
   FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_HOME/lcm/hc/
   config/PostPatchingHealthChecks.xml [-DlogLevel=log_level

   FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_HOME/lcm/hc/
   config/GeneralSystemHealthChecks.xml [-DlogLevel=log_level
   ```

For additional information about the `fapmgr` commands, see the Patch Manager Command Reference (page 9-1) section.

5.5 Apply Optional Language Packs

This step is only mandatory if Language Patches are necessary in the Fusion Applications environment.

If the environment contains installed languages other than US English, download and apply the same patches described in the previous section for the languages. Use the patching framework to build a patch plan and apply the patches.

Create a unique patch download directory, such as `/tmp/finbundle_lang`, and a separate patch plan for each language.

5.6 Apply Patches in Hot Patching Mode

Perform the following steps to apply a patch in hot patching mode:

1. Confirm that the patch can be applied in hot patching mode by running the validate command, as follows:

   ```
   fapmgr.sh validate -patchtop patchtop_directory -hotpatch
   ```

2. If the patch is valid for hot patching mode, run the apply command, as shown in the following example:

   ```
   fapmgr.sh apply -patchtop patchtop_directory -hotpatch [-maintenanceendtime
   YYYYMMDDHHmm] [-forceterminateactivetasks]
   ```

The command line parameters for hot patching are the following:

- `hotpatch`: Use hotpatch mode for applying or validating a patch
- `maintenancewaitperiod`: The number of minutes until impacted jobs will be paused
- `maintenanceendperiod`: The estimated time and date that a hotpatch session will complete
- `forceterminateactivetasks`: Force all active tasks to terminate after the maintenance wait period

5.7 Verify Whether Patch Manager Was Successful

Verify whether Patch Manager was successful by reviewing the Log Summary from a browser, as follows:

- If Patch Manager was successful: End of process

- If Patch Manager was not successful: See the Diagnostic and Troubleshoot Functional Patching Sessions (page 8-6) section.

6
Apply One-Off Patches

This section describes how to apply Functional One-Off Patches to your Oracle Fusion Applications environment.
The topics related to applying One-Off Patches are as follows:

- Prepare to Apply a Functional One-Off Patch (page 6-1)
- Apply Functional One-Off Patches (page 6-1)

6.1 Prepare to Apply a Functional One-Off Patch

MANDATORY: These steps must be performed prior to applying individual patches using Patch Manager.

1. To confirm whether an issue may be resolved by a patch for Oracle Fusion Applications, research the issue on My Oracle Support
2. After finding a patch that may resolve the issue, confirm whether the patch was previously applied to the system. Run the Patch Status report to see if specific patches were applied, as described in the Patch Status Report (page 9-5) section.

6.2 Apply a Functional One-Off Patch

The end-to-end process of obtaining and applying individual patches using Patch Manager is described as follows:

1. Upon determining that a new patch is required, download the patch from My Oracle Support. Unzip the patch zip file in the *PATCH_TOP* directory.
2. Review the README file that accompanies the patch. This file contains important information and instructions that must be followed. If a patch contains pre-installation or post-installation manual steps, they are described in the patch README file. If there are patches listed under "Other Patches" in the README file, download and apply them before continuing with the application of the Oracle Fusion Applications patch.
3. Run the Patch Impact report to see the artifacts and managed servers impacted by this patch as described in the Patch Impact Report (page 3-6) section. An example of the report command is as follows:

   ```
   FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -patchimpact -patchtop
   path_to_unzipped_patch
   ```

4. Run the `fapmgr validate` command to find any potential issues that could result in failure while applying the patch. Resolve all validation failures during this step, to prevent failures during patch application. During patch application, the patch validation runs again by default. An example of the `validate` command is as follows:

   ```
   FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh validate -patchtop
   path_to_unzipped_patch
   ```

5. Prevent locks on patched objects and other data issues during patching of database artifacts by performing the following:
   a. Ensure all users are off the system.
   b. Set the environment variable for your *APPLICATIONS_BASE* and *FA_ORACLE_HOME* directories.
   c. Run Health Checker to perform the Patching Readiness Health Checks and the General System Health Checks. Examples of the Health Checker commands are as follows:

      ```
      FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_HOME/lcm/hc/
      config/PatchingReadinessHealthChecks.xml [-DlogLevel=log_level]
      FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_HOME/lcm/hc/
      config/GeneralSystemHealthChecks.xml [-DlogLevel=log_level]
      ```

   d. Manually shut down the Oracle Enterprise Scheduler Service (ESS) servers by performing the following steps:
      i. Stop the ESS request processor and dispatcher to prevent new requests from being processed.
      ii. Cancel any in-progress requests.
      iii. Shut down the ESS WebLogic Server Managed server.
   e. If the patch contains BI Publisher reports, ensure to back up any customized versions of BI Publisher reports. If a patch includes an update to a catalog object that was delivered with an Oracle Fusion application, the patch will overwrite any customizations applied to the original report.

      If a patch contains BI Publisher artifacts, the BI OPMN control process, which is similar to a node manager, must be running for online mode validation to succeed.
6. Apply the patch using the`fapmgr apply` command. An example of the apply command is as follows:

   ```
   FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh apply -patchtop path_to_unzipped_patch [-
   workers number_of_database_workers]
   ```

7. Monitor the progress of the patching session and verify its successful completion by reviewing the Log Summary from a browser.
8. Review the Diagnostics report to determine if any manual steps are required for the artifacts included in the patch and where the artifacts were copied in *FA_ORACLE_HOME*.
9. Review the Diagnostics report to determine if Managed Servers require restart. If any servers must be restarted manually, the report provides the target domain and the names of the cluster and Managed Servers that must be restarted.

   If the patch contains JAZN artifacts, resolve any conflicts by reviewing the log files.
10. Run Health Checker to perform the Post Patching Health Checks. An example of commands is as follows:

    ```
    FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_HOME/lcm/hc/config/
    PostPatchingHealthChecks.xml [-DlogLevel=log_level
    FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_HOME/lcm/hc/config/
    GeneralSystemHealthChecks.xml [-DlogLevel=log_level
    ```

7

Apply Identity Management (IDM) Patches

This section describes how apply Oracle Identity Management (IDM) in Patching Framework.
The primary purpose of the Oracle Identity Management (IDM) Patching Framework for Oracle Fusion Applications is to simplify and expedite the maintenance of the code and functionality shipped as part of Oracle Identity Management for the Oracle Fusion Applications suite of products.

IDM patching can be either manual or automated depending on a variety of factors.

7.1 Overview of the Oracle Identity Management Patching Framework

The Oracle Identity Management patching framework coordinates the application of multiple patches to an Oracle Identity Management deployment and includes the following features:

- Patches all products within the Oracle Identity Management domain, including dependencies
- Runs across multiple machines
- Uses shared or local storage
- Runs during both initial provisioning and on an ongoing basis
- Runs in a defined, tier-wise order, minimizing downtime based on the patches being applied
- Stops and starts affected servers, as required and when appropriate
- Includes the ability to execute post-patch artifact changes
- Includes comprehensive state-sharing and reporting

Oracle Identity Manager includes patches for the following products that are installed in the Oracle Identity Management domain:

- Oracle Internet Directory
- Oracle Virtual Directory
- Oracle Directory Services Manager
- Oracle Identity Federation
- Oracle Access Manager
- Oracle Identity Manager
- Oracle HTTP Server
- Oracle HTTP Server WebGate
- Oracle SOA Suite

- Oracle WebLogic Server

7.1.1 Understand Oracle Identity Management Patching Framework Concepts

The Oracle Identity Management Patching Framework is composed of the Oracle Identity Management Patch Manager and the Oracle Identity Management Patcher tools. These tools work to apply patches to the Oracle Identity Management environment, using complete information about the deployment topology and verifying what services are running on which hosts. Based on the topology and the patches available, a patch session is created that defines and executes a patch plan. The Oracle Identity Management Patch Manager is used to generate the patch plan.

The patch plan is then executed by the Oracle Identity Management Patcher by:

- Stopping and starting servers
- Applying patches, as required, in an optimal manner

7.1.2 About Oracle Identity Management Patch Manager

The Oracle Identity Management Patch Manager is a tool that generates the patch plan and controls the patch session.

The Oracle Identity Management Patch Manager generates the patch plan as follows:

1. A *PATCH_TOP* directory containing patches, classified by each product subdirectory is provided to the tool.
2. The *PATCH_TOP* directory is scanned and initial validations are performed.
3. The deployment topology is read and analyzed.
4. The information obtained in Step 2 and Step 3 is combined, and a patch plan is generated using the `OPlan` utility. The patch plan is generated in HTML and plain-text formats, as well as binary format used for execution.

The topology data used by the tools is located in the topology store, which is an XML file located at `$LCM_CONFIG/topology/topology.xml`. This file contains most of the environment information used by the tools to apply patches. Additionally, the `provisioning.plan` file, located at `$IDM_TOP/provisioning/plan`, is also used for some tasks.

7.2 Verify the `patchtop-contents.properties` File

The downloaded patches must be organized in the following directory structure:

- A top-level *PATCH_TOP* directory containing different subdirectories for storing product-specific patches.
- Mapping between the products and the relative paths of the subdirectories under the *PATCH_TOP* stored in `patchtop-contents.properties`.

The relative paths of the subdirectories should be populated correctly in the patchtop-contents.properties file under the `$IDM_LCM_TOP/patch/config/` directory to ensure that the Oracle Identity Management Patching Framework can find the patches.

CONDITIONAL: There is a default structure already supported by the `patchtop-content.properties` file. If you do not want to follow the existing directory structure for storing the patches, then ensure that the `patchtop-content.properties` file is updated with the relative paths created under the *PATCH_TOP* so that the patching framework can find the product-specific patches correctly.

The following example provides details of the `patchtop-contents.properties` file:

```
#key: name of Fusion Middleware/Application patch component
#value: list of PATCH_TOP subdirectories containing the patches of the component
separated by commas.
common=oracle_common/patch
dir=idm/patch/oid, idm/patch/ovd, pltsec/patch
oam=iamsuite/patch/oam, idm/patch
odsm=idm/patch/odsm
ohs=webtier/patch
ohswg=webgate/patch
oif=idm/patch/oif, oif/patch
soa=soa/patch
wls=smart_update/weblogic
```

The targets shown on the left side cannot be modified, but the values on the right side can be updated. These values are the relative paths from the *PATCH_TOP*. If these paths are deleted from the file, the Oracle Identity Management Patching Framework assumes the default path location.

7.3 Verify the `env.properties` File

The `env.properties` file, located at `$IDM_LCM_TOP/patch/config/env.properties`, contains all environment variables required by the Oracle Identity Management Patching Framework. These properties are populated by the provisioning flow. Before running the Oracle Identity Management Patch Manager and Oracle Identity Management Patcher tools, ensure that the environment variables described in the following table are set:

Table 7-1 Environment Variables

Name	Value	Mandatory	Description
JAVA_HOME	JDK absolute path	Yes	The path pointing to the JDK location.
IDM_TOP	IDM_TOP absolute path	Yes	The absolute path of the IDM_TOP where IDM products are installed and configurations are stored.
LCM_CONFIG	IDMLCM absolute path	Yes	Absolute path where the IDMLCM configuration is stored.
ANT_HOME	Ant Home	No, but recommended	Absolute path pointing to the root directory of an Apache Ant distribution. This is required only to apply artifact changes for some products. If this environment variable is not set, impacted artifact changes may not complete.

Table 7-1 (Cont.) Environment Variables

Name	Value	Mandatory	Description
RETURN_MESSAGE_BUFFER_SIZE	This buffer size includes standard output and error messages stored in log files Default value is 8KB	No	The size of return message that is stored for each command executed. Affects the size of output printed to console and logs. Available units are as follows: • B (byte) • KB (kilobyte) • MB (megabyte) • GB (gigabyte)
COMMAND_TIMEOUT	A number and unit default value is 3600s (1 hour)	No	Timeout value followed by unit. If command execution takes longer, it is terminated. Permissible units are as follows: • ms (milliseconds) • s (seconds) • m (minutes) • h (hours) • d (days)

The `env.properties` file is populated during the provisioning flow. However, in case of multiple `IDM_TOP`s are using a single Oracle Identity Management provisioning and patching tools install, then the values must be deleted of the `IDM_TOP` and `LCM_CONFIG` variables from the `env.properties` file and set the correct values.

There is also an option to set the environment variables through the command line using the commands listed. However, ensure that the existing values are deleted from the `env.properties` file before setting the values. In case of use a POSIX-compliant shell, use the following command:

```
export JAVA_HOME=<JDK absolute path>
```

7.4 Use the Oracle Identity Management Patching Framework

The Oracle Identity Management Patching Framework consists of the Oracle Identity Management Patch Manager and Oracle Identity Management Patcher tools. The following sections describe how to create and apply the patch plan:

- Create an IDM Patch Plan (page 7-6)
- Apply the Patches (page 9-2)

7.4.1 Create an IDM Patch Plan

Perform the following steps to create the patch plan using Oracle Identity Management Patch Manager:

- Run Oracle Identity Management Patch Manager (page 7-5)
- Understand the Patch Plan (page 7-6)

- Create a Patch Plan (page 7-6)

7.4.1.1 Run Oracle Identity Management Patch Manager

To run the Oracle Identity Management Patch Manager, use the command line utility, `idmpatchmgr`, located in the `$IDM_LCM_TOP/patch/bin` directory. Its shell script sets the environment and calls the utility. For UNIX, the shell script is `idmpatchmgr.sh`. `idmpatchmgr` and can be run with various commands and options. Oracle Identity Management Patch Manager maintains a stateful session to track the patch process coordination with the Oracle Identity Management Patcher tool.

MANDATORY: The Oracle Identity Management Patch Manager must be run on the primordial host to create the patch plan as described in the Create the Patch Plan (page 7-6) section. A new patching session cannot be created until the existing session is completed or is aborted.

Oracle Identity Management Patch Manager maintains a session file in the `$LCM_CONFIG/patch/session/` directory. The session file has the current state of the Oracle Identity Management Patch Manager patch session. At any given point in time there will be only one or zero active patch sessions existing on the primordial host.

The patch session displays one of the statuses as described in the following table. The status `COMPLETE` and `INCOMPLETE` are the terminal states; whereas `FAILED` and `ABORTING` are recoverable states.

Table 7-2 Patch Session Status

State	Description
`ACTIVE`	In-progress state
`FAILED`	Halted state in response to a step failing execution
`ABORTING`	Halted state in response to the administrator issuing an abort command
`COMPLETE`	Terminal state where all steps are executed
`INCOMPLETE`	Terminal state if a session is aborted, either in response to a step execution failure or otherwise

Run the Oracle Identity Management Patch Manager, use the command line utility, `idmpatchmgr`, where instructions in brackets are optional . Example of the Oracle Identity Management Patch Manager command is a follows:

```
(UNIX) $IDM_LCM_TOP/patch/bin/idmpatchmgr.sh <command> [-options]
```

Where <command> is any IDM Patch Manager command, and the [options] are any options desired for the given command. The following table describes all the IDM Patch Manager commands:

Table 7-3 Oracle Identity Management Patch Manager Commands

Command	Description
apply	Starts a patch session where selected patches will be deployed.
rollback	Starts a patch session where selected patches will be removed.

Table 7-3 (Cont.) Oracle Identity Management Patch Manager Commands

Command	Description
abort	Ends a patch session without completing all planned steps.
progress	Displays the status for an ongoing patch session.

To view additional information for any `idmpatchmgr` command, use the following syntax:

```
(UNIX) $IDM_LCM_TOP/patch/bin/idmpatchmgr.sh command -help
```

7.4.1.2 Create the Patch Plan

To create a patch plan containing instructions for applying patches to an Oracle Identity Management environment, run the `idmpatchmgr apply` command. This plan can be executed by running the Oracle Identity Management Patcher tool.

MANDATORY: To create the patch plan, run the Oracle Identity Management Patch Manager on the primordial host.

Syntax

```
(UNIX) $IDM_LCM_TOP/patch/bin/idmpatchmgr.sh apply -patchtop patch_top
```

For more information about the way the patch plan is generated, see the Understand the Patch Plan (page 7-6) section.

7.4.1.3 Understand the Patch Plan

The patch plan is automatically generated by the Oracle Identity Management Patch Manager. To do so, Oracle Identity Management Patch Manager performs the following:

- The `apply` command validates the given *`PATCH_TOP`* location and validates the existence of the patch session with `ACTIVE` or `FAILED` status.
- If no patch session exists, the patch scanner is internally invoked to validate and generate a composite bundle patch from the provided *`PATCH_TOP`*. This bundle patch is internally used in the plan generation. The composite bundle patch is created in the location: `$LCM_CONFIG/patch/patches`.
- A patch plan is generated with instructions for applying patches using the topology store information and composite bundle patch.
- The `apply` command generates the patch plan in the following location in HTML and plain text formats:

  ```
  $LCM_CONFIG/patch/status/current-sessionID/manager/log/
  PatchInstructions.html

  $LCM_CONFIG/patch/status/current-sessionID/manager/log/
  PatchInstructions.text
  ```

  The patch plan in HTML and plain text formats provides useful information regarding the Oracle Identity Management environment, commands executed by the Oracle Identity Management Patcher, total number of steps, steps that require

downtime and so on. This enables you to better understand the Oracle Identity Management Patching Framework execution flow.

- At the time of plan generation, a new patch session is created in `ACTIVE` status, with all steps with status `PLANNED`. The patch session is stored in the `$LCM_CONFIG/patch/session/session` file. The step information is stored in the `$LCM_CONFIG/patch/session/step` file.
- The log files are generated in the following locations:

  Before the session is created:

  `$LCM_CONFIG/patch/status/log/idmpatchmgr.log`

  After the session is created:

  `$LCM_CONFIG/patch/status/`*`currentSessionID`*`/manager/log/idmpatchmgr-session.log`

The following table lists the option available for the `apply` command:

Table 7-4 `apply` Command Option

Option	Description
`-patchtop`	Displays the path to the location of the patches.

7.4.2 Apply Oracle Identity Management Patches

The following section describe the concept of applying Oracle Identity Management Patcher is based on an understanding of the Oracle Identity Management Patches utility and consists of applying the patches and applying artifact changes. This section contains the following topics:

- Understand the Oracle Identity Management Patcher (page 7-7)
- Apply the Patches (page 9-2)
- Apply Artifact Changes (page 7-8)

7.4.2.1 Understand the Oracle Identity Management Patcher

The Oracle Identity Management Patcher is the tool used to apply Oracle Identity Management (IDM) patches to an Oracle Fusion Applications environment.

To apply patches, use the `run` command. This command performs the following tasks:

- Validates the existence of a patch session and the availability of one or more steps with status `PLANNED` for the host where the tool is running.
- If there are one or more steps with status `PLANNED` for any other host prior to the above steps, then Oracle Identity Management Patcher reports that the execution is not possible until execution is complete for the other host.
- Creates the following log file named `status` with the details:

  `$LCM_CONFIG/patch/status/`*`currentSessionID`*`/hosts/`*`currentHostName`*`/status`

- When Oracle Identity Management Patcher starts executing the patching steps, the status log file is updated with `key =` *step-id* and `value = RUNNING`. After

setting the status, it extracts the command from the execution step and invokes the command using the step executor. On successful execution of the command, the status log file will be updated with `key = step-id` and `value = COMPLETED`. The execution continues to the next step from the execution plan for the current host.

- If there are no steps to be executed for the current host, it halts the execution and updates the administrator on the next steps to be executed.
- The `run` command also updates the session status. When reusing the `run` command, Oracle Identity Management Patch Manager displays the results.
- On failure, the status log file is updated with `key = step-id` and `value = FAILED` and execution is stopped.
- The `run` command generates log files in the following locations:

  Before the session is created:

  ```
  $LCM_CONFIG/patch/status/log/idmpatchmgr.log

  $LCM_CONFIG/patch/status/log/idmpatch.log
  ```

  After the session is created:

  ```
  $LCM_CONFIG/patch/status/currentSessionID/manager/log/idmpatchmgr-session.log

  $LCM_CONFIG/patch/status/currentSessionID/hosts/hostname/log/idmpatch-session.log
  ```

For information about how to use theOracle Identity Management Patcher `run` command, see the Apply the Patches (page 9-2) section.

7.4.2.2 Apply the Patches

To run the Oracle Identity Management Patcher, use the command line utility, `idmpatch`, located in the `$IDM_LCM_TOP/patch/bin` directory. Its shell script sets the environment and calls the utility. The following command shows the basic syntax for the `idmpatch` utility:

```
(UNIX) $IDM_LCM_TOP/patch/bin/idmpatch.sh run
```

OPTIONAL: To run only the prerequisites, use the `prereq` option . This will not stop and start the services or apply and rollback patches. The syntax to run the `idmpatch` is as follows:

```
(UNIX)$IDM_LCM_TOP/patch/bin/idmpatch.sh run -prereq
```

7.4.2.3 Apply Artifact Changes

Oracle Identity Management Patch Manager supports the application of post-patch artifact changes, such as adding an entry within a configuration properties file or invoking a product MBean. While most patches do not include them, Oracle Identity Management Patch Manager automatically executes the changes after all binary patch application for a single product is completed for those patches that do.

For example, if three patches [1, 2, 3] are applied to Oracle Access Manager within a patch session, and 1 contains an artifact change, the order of operations is [binary 1, binary 2, binary 3, artifact 1].

8

Monitor and Troubleshoot Patches

This section describes how to monitor and troubleshoot Oracle Fusion Applications patching and AD Administration processing sessions.
The topics related to Monitor and Troubleshoot Patches are as follows:

8.1 About Oracle Fusion Applications Patch Manager Logging

Oracle Fusion Applications Patch Manager (Patch Manager) creates log files for the actions it performs. These logging capabilities track the progress of actions and assist in diagnosing issues. When using Patch Manager, it is possible to specify the level of logging detail.

The following logging detail levels are available:

- ERROR:1 (SEVERE) For an error that results in a failure.
- WARNING:1 (WARNING) For an error that does not result in failure, but should be reviewed.
- NOTIFICATION:1 (INFO) For high-level information about the progress of the process, no action necessary.
- NOTIFICATION:16 (CONFIG) For more detailed information about the progress of the process, no action necessary.
- TRACE:1 (FINE) For generating the first level of trace messages, used for diagnosing issues.
- TRACE:16 (FINER) For generating the second level of trace messages, used for diagnosing issues.
- TRACE:32 (FINEST) For generating the highest level of trace messages, used for diagnosing issues.

When selecting a more detailed level of logging, the log files also include the lower level of details. For example, in case of choose to see INFO messages in log file, WARNING and SEVERE messages also appear in the log files.

8.1.1 Log Files for Single Patch Manager Sessions

To examine the activities performed during patching sessions, review the associated log files. Patch Manager consolidates log files for each patching session under the directory, *APPLICATIONS_CONFIGAPPLICATIONS_CONFIG/lcm/logs/<Fusion Applications Release Version>/FAPMGR.*. This directory contains the top-level log file, `logsummary`*_fapmgr_command_timestamp*`.html`, along with related log files for each task performed during a `fapmgr` session. During a session, it is possible to view the Log Summary HTML file from a browser, which provides links to individual log files. To view the progress of the current patching session, refresh the Log Summary HTML file periodically. If a task fails, access the links to the associated log files to assist in diagnosing the failure. For more information, see the Log Summary (page 8-6) section.

When a patching session completes, its log files are archived in *APPLICATIONS_CONFIG*`/lcm/logs/<Fusion Applications Release Version>/FAPMGR/logarchive/`*Patch Number/fapmgr_command/session ID/timestamp*. The session ID is unique and the time stamp is the start time for the session. Note that whenever Patch Manager runs a command where there is no patch number, such as `bootstrap, abort,` and `report`, the archive logs are named *APPLICATION_CONFIG*`/lcm/logs/<Fusion Applications Release Version>/FAPMGR/logarchive/fapmgr_command/`*timestamp*.

Log files for OPatch actions are initially written to the *FA_ORACLE_HOME*`/cfgtools/opatch/`*patch number_timestamp* directory.

The following table contains a list of log files created by Patch Manager during patching activities:

Table 8-1 Log Files for Single Oracle Fusion Applications Patch Manager Patching Activities

Log file or directory name under *APPLICATIONS_CONFIG*/lcm/logs/<Fusion Applications Release Version>/FAPMGR	Log file generated with:
FAPatchManager_apply_*timestamp*.log	Oracle Fusion Applications Patch Manager apply session
FAPatchManager_abort_*timestamp*.log	Patch Manager abort session
FAPatchManager_bootstrap_*timestamp*.log	Patch Manager bootstrap session
FAPatchManager_report_*reportname_timestamp*.log	Patch Manager report session
FAPatchManager_validate_*timestamp*.log	Patch Manager validate session
adpatch_apply_*timestamp*.log	Oracle Fusion Applications AutoPatch (AutoPatch) apply session
adpatch_abort_*timestamp*.log	AutoPatch abort session
adpatch_apply_*timestamp*_worker*number*.log	AutoPatch worker log file
adpatch_validate_*timestamp*.log	AutoPatch
adpatch_apply_*timestamp*_timingreport.lst	AutoPatch timing report

Table 8-1 (Cont.) Log Files for Single Oracle Fusion Applications Patch Manager Patching Activities

Log file or directory name under *APPLICATIONS_CONFIG*/lcm/logs/<Fusion Applications Release Version>/FAPMGR	Log file generated with:
logsummary_*fapmgr_command_timestamp*.html For reports: logsummary_report_*reportname_timestamp*.html	The consolidation of the log files generated by Patch Manager in HTML format for viewing and accessing links to other log files from a browser
patch number_fapmgr_command_session id_timestamp.marker	Marker file used while moving log files to a backup directory
ConfigContext_*timestamp*.log	Patch Manager in online mode
ExecutionContext_*timestamp*.log	Patch Manager in online mode
FAPMgrDiagnosticsSummary*fapmgr_command_timestamp*.html	The consolidation of the Patch Manager session in HTML format, known as the Diagnostics report
FAPMgrDiagnosticsSummary.html	This file is updated every five minutes during patch application.
diaghtmllogs_*mode_timestamp*	The Diagnostics report for Patch Manager in `apply` and `validate` mode. This is the directory that contains the log files in HTML format.
diaghtmllogs_*mode_timestamp*/FAPatchManager_apply_20120627022503.html	Patch Manager apply session
diaghtmllogs_*mode_timestamp*/adpatch_apply_*timestamp*.html	AutoPatch apply session
diaghtmllogs_*mode_timestamp*/adpatch_validate_*timestamp*.html	AutoPatch validate session
diaghtmllogs_*mode_timestamp*/adpatch_apply_timestamp_workernumber.html	AutoPatch worker log file
Utility Code_Mode_Artifact Type_PatchAction_Action_Task Id_time_stamp.log	Task level log files which are created for each artifact type and its actions, for example: • `fapcore_apply_jeepatch action_startmanagedservers_5827_20140212043616.log` • `fapcore_apply_odipatch action_importodimodel_5832_20140212043616.log`

Table 8-1 (Cont.) Log Files for Single Oracle Fusion Applications Patch Manager Patching Activities

Log file or directory name under *APPLICATIONS_CONFIG*/lcm/logs/<Fusion Applications Release Version>/FAPMGR	Log file generated with:
opatch*timestamp*.log e.g., opatch2017-08-30_14-31-40PM_1.log	OPatch log while running and after completion. • *APPLICATIONS_CONFIG/lcm/logs/<Fusion Applications Release Version>/FAPMGR/opatch/opatch*
fastartstop_*timestamp*.log <Domain name>_*timestamp*.log fastartstop_results.xml	FAStartStop log: while running the logs are located in the following: APPLICATIONS_CONFIG/lcm/logs/<Fusion Applications Release Version>/FAPMGR/fastartstop After completion the logs are located in the following: APPLICATIONS_CONFIG/lcm/logs/<Fusion Applications Release Version>/FAPMGR/logarchive/MAPPLY/<TimeStamp>/<Group number>/<APPLY | VALIDATE>/<Patching Session Number>/<Group time stamp>/fastartstop/fastarstop_<start | stop>_<time stamp>

8.1.2 Log Files for Multi-apply Patch Manager Sessions

The *multi-apply* feature of Patch Manager applies multiple patches after splitting them into groups within a My Oracle Support patch plan. Each group contains at least one patch and it is run as an individual session internally. Log files and diagnostic reports are generated for each individual session, in addition to a primary log file and diagnostics report for the overall multi-apply session.

Examine the activities performed during multi-apply patching sessions by reviewing the associated log files. Patch Manager consolidates log files for each patching session under the directory, *APPLICATIONS_CONFIG*/lcm/logs/<Fusion Applications Release Version>/FAPMGR. This directory contains the top-level log file, logsummary_*fapmgr_command_timestamp*.html, along with related log files for each task performed during a multi-apply session. During a session view this log summary HTML file from a browser, which provides links to individual log files. Periodically refresh the log summary HTML file to view the progress of the current patching session. If a task fails, access the links to the associated log files to assist in diagnosing the failure. See the Log Summary (page 8-6) section to get more information how the log summary is created and how the report is created.

Multi-apply sessions create a primary diagnostics report that provides information about the multi-apply patching session. It also includes links to corresponding diagnostics reports and archived log files for individual patches that were applied as a group.

The following table contains a list of log files created by Patch Manager during multi-apply patching activities:

Table 8-2 Log Files For Multi-apply Patch Sessions

Log file name or directory under FA_ORACLE_HOME/admin/FUSION/log	Log file description
FAPMgr_Multiapply_apply_*timestamp*.log	Primary log file from a multi-apply patching session
FAPMgr_Multiapply_DiagnosticsSummary_*timestamp*.html	Primary Diagnostics report from a multi-apply patching session in HTML format
FAPMgr_Multiapply_DiagnosticsSummary_*timestamp*.xml	Primary Diagnostics report from a multi-apply patching session in XML format
diaghtmllogs_*mode_timestamp*	The Diagnostics report from a multi-apply patching session in `apply` and `validate` mode. This is the directory that contains the log files in HTML format.
FAPMgrDiagnosticsSummary_apply_*timestamp*.html	Diagnostics report from a multi-apply patching session in HTML format
ADPatchDiagnosticsSummary_apply_*patchnumber_timestamp*.html	Diagnostics report for one patch in a multi-apply patching session in HTML format
logsummary_apply_*timestamp*.html	The consolidation of the log files generated by multi-apply patching session in HTML format for viewing and accessing links to other log files from a browser
Utility Code_Mode_Artifact Type_PatchAction_Action_Task Id_time_stamp.log	Task level log files which are created for each artifact type and its actions, for example: • `fapcore_apply_jeepatchaction_startmanagedservers_5827_20140212043616.log` • `fapcore_apply_odipatchaction_importodimodel_5832_20140212043616.log`

At the beginning of a multi-apply patching session, the log files from the previous session move to *`FA_ORACLE_HOME`*`/admin/FUSION/logarchive/MAPPLY/`*`timestamp`*. For each patch that is applied as part of a group, a `LogFilesLocationPointer.log` is created at the location where the log files will be archived, as if the patch was applied as a standalone patch. The content of the pointer files provides the absolute location of the corresponding group apply logs and provides the ability to navigate directly to the log files for a specific patch, even though it is applied using multi-apply. For example, if patches 19191630 and 20193112 are grouped into `GROUP1` and then applied, log files are archived in the following directory structure:

```
 logarchive
|
|-- MAPPLY
|        |-- 20120610235002
|        |       |-- FAPMgr_Multiapply_apply_20120610235002.log
|        |       |-- FAPMgr_Multiapply_DiagnosticsSummary_20120610235002.html
|        |       |-- FAPMgr_Multiapply_DiagnosticsSummary_20120610235002.xml
|        |       |-- GROUP1
|        |       |       |-- APPLY
```

```
|        |       |       |       |-- 123
|        |       |       |       |       |-- 20120610235302
|        |       |       |       |       |       |--
FAPatchManager_apply_20120610235302.log
|        |       |       |-- VALIDATE
|        |       |       |       |-- 122
|        |       |       |       |       |-- 20120610235102
|        |       |       |       |       |       |--
FAPatchManager_validate_20120610235102.log
|-- 19191630
|        |-- APPLY
|        |       |-- 123
|        |       |       |-- 20120610235302
|        |       |       |       |-- LogFilesLocationPointer.log
|        |-- VALIDATE
|        |       |-- 122
|        |       |       |-- 20120610235102
|        |       |       |       |-- LogFilesLocationPointer.log
|-- 20193112
|        |-- APPLY
|        |       |-- 123
|        |       |       |-- 20120610235302
|        |       |       |       |-- LogFilesLocationPointer.log
|        |-- VALIDATE
|        |       |-- 122
|        |       |       |-- 20120610235102
|        |       |       |       |-- LogFilesLocationPointer.log
```

8.2 Diagnostic and Troubleshoot Functional Patching Sessions

Log files are useful files that contain informational and error messages generated during patching. Log files are generated by Patch Manager, which coordinates patching activities by assigning tasks; OPatch, which runs the tasks for updating middleware artifacts; and AutoPatch, which runs the database tasks. If a task fails at any point or in any level, exact information about the failure will be included in the log file. Furthermore, progress of the patching session, including the details of a failed task, can be viewed from a browser by reviewing the Log Summary or the Diagnostic report.

For more information about how to examine the activities performed during the patching sessions, see the Log Files for Single Patch Manager Sessions (page 8-2), the Log Summary (page 8-6) section to understand the log summary creation,, and the Diagnostics Report (page 8-7) section to see the report during the patching session.

8.2.1 Log Summary

The Log Summary is created automatically when the Patch Manager is started. The Log Summary is continuously updated as the session progresses. This report exists in the *FA_ORACLE_HOME*`/admin/FUSION/log` directory and is named `logsummary_`*`fapmgr_command_timestamp`*`.html`. It contains links to all of the log files generated during the session. To view the report, open the report from a browser and periodically refresh the page to see updated links to log files as they are created. Open those links and monitor the progress of the session by refreshing the browser.

8.2.2 Diagnostics Report

After each patching session ends, the Diagnostics report is automatically generated and results may be viewed in a browser. Use this report during a patching session that is currently running, by generating the report from the command line. The Diagnostics report is located in the *FA_ORACLE_HOME*/admin/FUSION/log directory and is named FAPMgrDiagnosticsSummary*fapmgr_command_timestamp*.html.

For the fapmgr apply and validate commands, the Diagnostics report contains links to the line number in the logs file for each task. The links contained in the Diagnostics report go to the specific line in the corresponding HTML log files. The HTML log files exist in the directory diaghtmllogs_*mode_timestamp* directory.

During the upgrade flow, while the Loading Database Components Config Assistant or the Installing Downloaded Fusion Applications Upgrade Patches Config Assistant runs, the diagnostic report is not generated by default. It can be generated manually if required.

8.3 General Troubleshoot for Oracle Fusion Applications Patching

This section contains the following general troubleshooting scenarios for patching:

For troubleshooting information that is specific to patching security artifacts such as the jazn-data.xml file, data security files, and data role templates, see the Patch Security Artifacts (page 10-20) section.

For troubleshooting information that is specific to patching SOA composites, see the Troubleshoot Patching Sessions for SOA Composites (page 8-14) section.

For troubleshooting information that is specific to patching database content, see the Troubleshoot Patching Sessions for Database Content (page 8-20) section.

8.3.1 Start a New Patching Session After the Previous Session Failed

After the failure of a patching session, the following scenarios are possible:

- To abandon the previous session and start a new session, see the Abandon a Failed Patching Session (page 8-8)section.
- Despite failure, some tasks may display as running. For more information, see the Recovery from an Interrupted Patching Session (page 8-9) section.
- There can be only one patching session active at one time for Oracle Fusion Applications, Oracle Fusion Functional Setup Manager, and Oracle Fusion Middleware Extensions for Applications (Applications Core). If there is a failed Applications Core or Functional Setup Manager patching session that must be cleaned up, see the Abandon a Failed Patching Session (page 8-8) section.

8.3.2 Abandon a Failed Patching Session

If the patch session failed and there is no need to restart it, abandon the session by using the `abort` command. Remember that only one patching session can be running at a time and always make sure that processes associated with the previous patching session does not exist.

Mandatory: If the patching session is aborted, this session cannot be restarted and any pending patching actions, such as deployment actions, must be performed manually.

To abandon a previously failed session, run the `fapmgr abort` command . The `abort` command cleans up any intermediate states tracked by `fapmgr` and moves the log files for the abandoned session to an archive log directory so that a new patching session could start. Note that a session that is actively running cannot be abandoned.

Use the following syntax for the `fapmgr abort` command:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh abort [-force] [-logfile log file
name] [-loglevel level
```

The following table describes the options available for the `abort` command:

Table 8-3 abort Command Options

Option	Description	Mandatory
logfile	Name of the log file	No, default value is FAPatchManager_abort _*timestamp*.log
loglevel	Reporting level for messages see the Oracle Fusion Applications Patch Manager Logging (page 8-1)section	No, default value is INFO
force	Always forces the patching session to abort, even when the patching session has not failed see the Use Abort -force (page 8-9) section	No
help	Displays help	No

If the `fapmgr abort` command errors with a message "Another APPLY session is already running", use the `fapmgr forcefail` command first. Also confirm that the `FND_INSTALL_PROCESSES` table does not exist. For more information, see the Recovery from an Interrupted Patching Session (page 8-9) section.

8.3.2.1 Use `abort -force`

The `-force` option allows marking a current patching session as `ABORTED` to proceed to a new patching session after cleaning up the current patching session.

MANDATORY: This feature is intended for hot patching sessions where `APPLY` tasks are not automatically cleaned up and must be used only with the assistance of Oracle Support.

The abort command impacts normal patching and hot patching in the following ways:

Normal Patching

- `fapmgr abort`: If the abort fails, the patching session is set to `Abort_failed`, then attempt to abort the session again.
- `fapmgr abort -force`: Whether the abort fails or succeeds, the patching session is set to `ABORTED` and displays any failed tasks that must be performed manually.

Hot Patching

- `fapmgr abort`: The patching session is set to `Abort_failed` if one or more `APPLY` tasks attempt to run because `APPLY` tasks are not reversible in hot patching.
- `fapmgr abort -force`: The patching session is set to `ABORTED` and displays all tasks as "Failed to Abort". Contact Oracle Support to request assistance in manually performing these tasks.

8.3.3 Recovery from an Interrupted Patching Session

If a patching session was interrupted by a system failure when Patch Manager and AutoPatch were running and the patching-related database tables still show the patching session as running (only one patching session can be active at a time), but no patching-related processes are actually running, use the `fapmgr forcefail` command to update the patching tables by performing the following steps:

1. Confirm that no patching processes are running. From your operating system, check if any of these processes are running `fapmgr, javaworker, adpatch, adadmin`, and `adworker`. To stop these processes, use the appropriate command for your operating system.
2. Use the `fapmgr forcefail` command to update the patching tables. For example:

   `(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh forcefail [-logfile log file name] [-loglevel level]`
3. Abandon or restart the failed session, as follows:
   - To abandon the failed patching session, use the `fapmgr abort` command to start a new patching session.
   - To restart the failed patching session, use the `fapmgr apply` command to apply the same patch. The session starts from the failed task.

8.3.4 Avoid a Lost Connection During the Patching Session

If initiating a patching session from a terminal server, such as a laptop, the connection may be lost during extended periods of time when no messages are sent to the terminal. The terminal server may interpret this as inactivity and then end the session. For example, no messages are sent to the client when Patch Manager is stopping and starting servers, waiting for a failed task to be fixed, or is hung on a database task. To avoid this situation, ensure that the terminal server is configured appropriately to handle long durations of inactivity.

8.3.5 Jobs Are Running After Maintenance Wait Period Using Hot Patching

While applying a patch using hot patching, the following type of error is reported:

```
ESS Jobs are running on the domains (HCMDomain) even after maintenance wait
period.
```

If this error is reported, consider the following alternatives:

- Wait for ESS Jobs execution finish.
- Force all active tasks to terminate after the maintenance wait period, applying the patch using the `forceterminateactivetasks` option, as shown in the following example:

  ```
  fapmgr.sh apply -patchtop patch_top -hotpatch -online -
  stoponerror -maintenancewaitperiod 1 -maintenanceendtime date_time -
  forceterminateactivetasks
  ```

8.3.6 Unable to Apply a Hot Patch

Run the following command to validate if the patch is structured to be applied as a hot patch :

```
sh fapmgr.sh  validate -patchtop /patch_top/18060344 -hotpatch -online
```

If the following error occurs:

```
Validation FAILED.
You cannot apply the patch.
Check the log file for unexpected infrastructure failures.
```

It means that the patch is not structured to be applied as a hot patch. If this error occurs, apply this patch without using the `-hotpatch` option.

8.3.7 Resolve a Webcat Patch File Creation Failure

If a patch is being applied and it contains BI Publisher artifacts, the BI Presentation servers should not be running. If BI Presentation servers are running during the deployment of BI Publisher artifacts, the following error occurs:

```
java.lang.RuntimeException: Webcat patch file creation failed!
```

To resolve this issue, shut down the BI Presentation servers to release locks on the Oracle BI Presentation Catalog.

8.3.8 Resolve an EditTimedOutException Error

If during patch validation, the following error is received:

```
weblogic.management.mbeanservers.edit.EditTimedOutException
```

Manually release the edit session. This situation occurs when a domain is already in "edit" mode during patching, for example, when the server crashes when Patch Manager tries to stop and restart it.

To manually release the edit session, use the following procedure:

1. Log in to the admin console for the domain that is locked in edit mode.
2. In the admin console, confirm that **Release Configuration** is enabled in the **Change Center** menu.
3. Click **Release Configuration** to release the edit session.

8.3.9 Revert to a Previous Flexfield Definition After it is Updated by a Patch

If flexfield changes are not ready to be implemented after applying a patch containing flexfield changes, then revert to a previous version of that flexfield definition. The Flex Modeler creates an MDS label, `FlexPatchingWatermark`*`date+time`*, before it initiates the flexfield deployment, which establishes the watermark for what was in MDS before the patch was applied. The name of the label is included along with the Flex Deployment report in the patching log file as a reference.

To use a previous version of a flexfield definition, use the WLST command `promoteMetadataLabel`.

To delete all previous MDS labels for a flexfield, first confirm the changes delivered by a patch can be used (keep old MDS labels impacts performance), then use the WLST command `deleteFlexPatchingLabels`.

8.3.10 Resolve an Online Validation Error for BI Artifacts

If a patch contains BI artifacts the BI OPMN control process, which is similar to a node manager, then it has to be up for online mode validation to succeed. Online validation reports the following error if the BI OPMN control process is not up:

```
The deployment of BI Publisher artifacts will not be attempted because the
BI Presentation server is neither fully started nor down.
One likely cause is that the BI OPMN control process is not running. Make
sure that the BI OPMN control process is up and the BI
Presentation server is started successfully before applying this patch. If
this server is not fully started, you must stop the BI Presentation
server, manually deploy the BI Publisher artifacts, and then re-start the BI
Presentation server
```

To resolve the issue, ensure that the BI OPMN control process is up and running.

8.3.11 Error Update Status While Apply a Patch

If while applying a patch, Patch Manager fails with the error "Failed while trying to update task status" or "Error while updating session status", solve the issue by performing the following steps:

1. Rerun the `fapmgr` command.
2. The patching session fails again with an error such as the following:

```
Failed while deploying SOA
composite [/u01/APPLTOP/fusionapps/applications/hcm/deploy/
sca_HcmCompWorkbenchRsgnWorkerComposite.jar] using OPatch SDK
deploy.
Reason: [java.lang.RuntimeException: The composite
"HcmCompWorkbenchRsgnWorkerComposite" with revision
"22153501_17358145" is already deployed. Current Operation cannot
be performed.]
```

3. If after undeploy the SOA composite using a WLST command to resolve the failure, Patch Manager fails again with an error such as "The last session APPLY for patch */path_to_patch/*fapatch/17358145 is incomplete", resolve this issue by running the `fapmgr` command in `forcefail` mode. For more information, see the Recovery from an Interrupted Patching Session (page 8-9) section.

8.3.12 Find Artifact Versions

The `opatch -lsinventory -detail` command provides a report that lists all patches, artifacts and artifact versions that were modified within each patch applied to a given Oracle home. This report lists only artifacts that were modified. If an artifact does not appear on this report, then the artifact remains at its base version. Run this report when Oracle Support Services are used and an artifact version needs to be provided.

To generate the report, use the following command syntax:

```
opatch/opatch lsinventory -detail -oh FA_ORACLE_HOME -invPtrLoc \
FA_ORACLE_HOME/oraInst.loc -jre JAVA_HOME
```

The following example depicts the section of the report that displays patches applied. To use the report, find the latest entry for the specific artifact and note the version reported.

```
Interim patches (11) :

Patch  11801        : applied on Wed Feb 02 17:57:53 PST 2011
   Created on 18 Jan 2011, 16:09:54 hrs PST8PDT
   Bugs fixed:
     11801
   Patch is translatable.
   Files Touched:
     AdfPjfIntMspUi.jar, version "23.0" --
> ORACLE_HOME/prj/deploy/EARProjectsFinancials.ear/EARProjectsFinancials.war/WEB-
INF/lib/AdfPjfIntMspUi.jar
   Patch Location in Inventory:
     /u01/FUSIONAPPS_APPLTOP/fusionapps/applications/inventory/oneoffs/11801
   Patch Location in Storage area:
```

```
     /u01/FUSIONAPPS_APPLTOP/fusionapps/applications/.patch_storage/
11801_Jan_18_2011_16_09_54

Patch  12801        : applied on Tue Feb 01 21:30:17 PST 2011
   Created on 28 Jan 2011, 14:26:56 hrs PST8PDT
   Bugs fixed:
     12801
   Patch is translatable.
   Files Touched:
     EFFmetadata.mar, version "35.0" --> ORACLE_HOME/hcm/
deploy/EarHcmCoreSetup.ear/EFFmetadata.mar
     system-jazn-data.xml, version "35.0" --> ORACLE_HOME/hcm/
security/policies/system-jazn-data.xml
   Patch Location in Inventory:
     /u01/FUSIONAPPS_APPLTOP/fusionapps/applications/inventory/oneoffs/12801
   Patch Location in Storage area:
     /u01/FUSIONAPPS_APPLTOP/fusionapps/applications/.patch_storage/
12801_Jan_28_2011_14_26_56
```

8.3.13 Back Out of Patches After They Have Been Successfully Applied

Always test the application of a patch and the functionality related to the patch on a test system before applying it to production system to ensure it will be successful. There is no automated method of backing out patches.

Oracle strongly recommends to work with Oracle Support Services to back out a patch.

8.3.14 FAPMgr Failure

Problem

POD runs Out of Memory and OS kernal ends FApMgr.

Solution

1. Take process memory map to know which process has taken more memory.
2. Take a snapshot of the used and free memory.
3. Contact the team owner of the process to get the workaround to release the memory.
4. When the memory has at least of 5 GB of free memory then, contact the FAPMgr support team to perform below steps:
   a. Run `Fapmgr focefail`.

      > **Note:**
      >
      > DO NOT run ABORT.

   b. Clean all the child processes if any in all the hosts.
   c. Restart the patching session.

8.4 Troubleshoot Patching Sessions for SOA Composites

This section describes the troubleshooting process for errors that can occur when patching Service-Oriented Architecture (SOA) composites. These processes assume that the validation and application of the patch are done in online mode. SOA patching errors can be divided into the following categories:

- **Error occurs during validation**

  Patch Manager can detect and report validation errors before changes have occurred to the file system. If the validation errors are not resolve before applying the patch, the patch fails and the SOA composites has to be manually deployed after resolve the validation errors.

- **Error occurs during the patch apply phase**

  These errors may require contacting Oracle Support Services to restore the system back to a known working state and can be further divided into these categories:

  – The SOA composite failed to deploy and Patch Manager recovered from the failure.

  – The SOA composite was not deployed successfully and the recovery failed. Therefore, the composite may be partially deployed.

  – The system is in an unknown state, as the result of a timeout occurring during deployment. Patch Manager cannot determine if the SOA composite is deployed, not deployed, or partially deployed.

When SOA composite failures occur, review the error messages in the Diagnostics report and related log files and follow the applicable steps in one or more of the following categories:

- Basic Troubleshoot for SOA Composite Failures (page 8-14)
- Troubleshoot SOA Composite Validation Failures (page 8-16)
- Troubleshoot SOA Composite Deployment Failures (page 8-18)
- Troubleshoot Complex Failures during SOA Patching (page 8-19)

8.4.1 Basic Troubleshooting for SOA Composite Failures

SOA composite validation and deployment can fail for multiple reasons. Review the following steps for basic troubleshooting:

1. After validate or apply a patch that contains SOA composites, review the Diagnostics report for errors. The report output is in HTML format for viewing from a browser and is located here:

   FA_ORACLE_HOME/admin/FUSION/log/
   FAPMgrDiagnosticsSummary*date:session*.html

   The **Module Task Details** section of the report displays the following information to assist in your troubleshooting:

   - Mode: Middleware, in this case.
   - Phase: Validation or Patch Application, in this case.

- Product Family: The short name of the product family.
- Task.

  The following information displays for SOA composites:

  - Name of composite
  - Domain name
  - Path to composite JAR in *FA_ORACLE_HOME*
  - Revision of composite
- Status: Possible values of Success, Failed, or Skipped.
- Duration: Total time the task ran.
- Start Time: Time and date the task started.
- End Time: Time and date the task ended.
- Warning/Error Message: The error message displays as a `java.lang.RuntimeException`. The message often includes a suggestion for resolving the failure.
- Log file: The path and file name of the high level log file, `FAPatchManager_`*`fapmgr_command_timestamp`*`.log`, associated with the task. From the Module Execution Summary section of the Diagnostics report, review log files by accessing the link to the Log Summary. For more information, see the Log Summary (page 8-6) section.
- Line Numbers: The line numbers in the log file associated with the task.

2. SOA log files are located in this directory: *FA_ORACLE_HOME*`/admin/FUSION/log/fapatch/fapatch_`*`release_number`*`/soalogs`. If merge operations are performed on a SOA composite, due to runtime customizations, such as design time and run-time (DT@RT) changes or property changes, a merge log file is generated. There is one merge log file per domain and the name of merge log files follows this naming convention: `fapatch_`*`domain_nametimestamp`*`.merge.log`
3. Restart the SOA servers and for each failure, follow Steps 4 through 9.
4. Determine if there is a cause for the error that must be resolved, in addition to restarting the server, by referring to the Diagnostics report, Oracle Fusion Applications log files and SOA log files. Examples of other causes include database errors, coherence configuration errors, and out of memory issues.
5. Determine if a manually restore the system back is need it to its state before the application of the patch was attempted. The following scenarios do not require manual restoration of the system:

   - Errors occurred during the validation phase.
   - Errors occurred during the patch application phase but the recovery was successful, so the system was recovered to its original state. The Diagnostics report displays this message in this case:

     ```
     Deployment of SOA composite[artifact_name and path]failed, but the
     system has recovered successfully.
     Suggestion: You must manually deploy the composite using the WLST
     command.
     ```

   If the system needs to be restored, follow the steps in Step 6.

6. If a failed deployment leaves a composite in an inconsistent state, restore the system to its original state. In case of prefer to use Oracle Fusion Applications Control to restore the system, see Step 7. Perform the following WLST commands to restore the system:

   a. Before the application of the patch from the Diagnostics report, please find out the last active revision of the composite, it is indicated by this message: `The last active version of the composite before patch application began was` [*version*].

      In the following examples, 1.0 is the previous revision and 2.0 is the patched revision.

   b. Undeploy the patched revision of the composite if it exists in the system.

      ```
      sca_undeployComposite('http://server01:8001', 'POProcessing', '2.0')
      ```

   c. Mark the previously active revision of the composite as active and as a default revision.

      To activate the old revision, use the following command:

      ```
      sca_activateCompositeMb('POProcessing', '1.0')
      ```

      To assign the default composite, use the following command:

      ```
      sca_assignDefaultCompositeMb('POProcessing', '1.0')
      ```

7. Restore the system to its original state using Enterprise Manager Cloud Control with Oracle, follow up the steps below:

   a. Find the active revision of the composite before the application of the patch from the Diagnostics report, as indicated by this message: `The last active version of the composite before patch application began was` [*version*].

      In the following steps, 1.0 is the previous revision and 2.0 is the patched revision.

   b. Undeploy the patched revision of the composite if it exists in the system.

   c. Mark the previously active revision of the composite as active and as a default revision.

8. Manually deploy the SOA composites included in the patch by following the steps :

   ```
   sca_patchComposite('SOA-Infra URL', user, password,
   '/FA_ORACLE_HOME/crm/deploy//sca_FinAprev2.0.jar', mergeLogFile='/tmp/
   mergelog
   ```

9. If you are unable to resolve the failure, file a service request with Oracle Support Services and provide the logs and information as described in the Get Started with Troubleshooting and Logging Basics for Oracle SOA Suite section in the *Oracle Fusion Applications Administrator's Guide*.

8.4.2 Troubleshoot SOA Composite Validation Failures

This section describes common problems and solutions for SOA composite validation failures.

MANDATORY :Errors that occur during the validation phase must be resolved before applying the patch. In case of these errors occurred during patch application, manually deploy the SOA composites after resolve the validation errors.

Patch Manager captures the OPatch validation log files, which can be found by referencing the Diagnostics report or the Log Summary. The errors in the log files provide information about the cause of the failure and are often followed by recommended actions.

This section contains troubleshooting information about the following failures:

8.4.2.1 Oracle JDeveloper Customization Error

An error that is related to a JDeveloper customization occurs when a SOA composite was customized but the customization was not saved. Before applying a patch that includes the next revision of the composite, save the customization. Follow the steps mentioned in the Preserve SOA Composite JDeveloper Customizations Before Apply a Patch (page 10-32) section to resolve this error.

8.4.2.2 SOA Server Not Available

If the SOA server is down or not available for patching, patch validation succeeds, but in case of a warning message stating that the deployment of the composite will not be performed because the SOA infrastructure is down.

Use Oracle Enterprise Manager Fusion Applications Control (Fusion Applications Control) to check the state of the SOA server. For example, if an "Initializing SOA" warning message displays on the home page, Oracle recommends wait until the SOA server is completely up and running, with all composites initialized.

For more information, see the SOA Server Does Not Start section in the *Oracle Fusion Applications Administrator's Guide*.

8.4.2.3 Administration Server Not Available

If the Administration Server is down or not available for patching, patch validation fails. Use Fusion Applications Control to check the state of the Administration Server. For more information, see the Access Fusion Applications Home Pages in Cloud Control section in the *Administrator's Guide*.

8.4.2.4 SOA-Infra Server Is Ready

If the SOA-infra server is down or not available for patching, patch validation fails, use Fusion Applications Control to check the state of the SOA-infra server. Confirm that all dependent services are running and that all composites deployed into the SOA-infra server are present. It may take some time after SOA-infra is up for all services to initialize. In case of use a cluster, perform this check for all SOA-infra servers in the cluster.

8.4.2.5 Composite with Identical Revision Is Already Deployed

In case of receive an error stating that a composite with a specific revision is already deployed, the SOA composite in the patch was previously deployed by a patch or manually by a user.

Resolve this error either by not applying the current patch or by undeploying the composite before applying the patch. Note if the composite is undeployed, all customizations may have made in the composite will be lost. Use the following command to undeploy a composite:

```
sca_undeployComposite(serverURL, compositeName, revision)
```

For more information about using the `sca_undeployComposite` command, see "Undeploy SOA Composites Using WSLT Command" in the *Oracle Fusion Applications Upgrade Guide*.

8.4.3 Troubleshoot SOA Composite Deployment Failures

This section describes common problems and solutions for SOA composite deployment failures during patching.

MANDATORY: Errors that occur during the deployment phase must be resolved as soon as possible because the system has been modified. Do not try to roll back or reapply patches that have errors during deployment. After resolve the cause of the error, you must deploy the composite manually.

This section contains the following topics:

8.4.3.1 Failed to Make New Composite Revision the Default

In case of receive an error in making the new composite the default, the option available is manually assign the new composite as the default.

Use the following WLST command to assign the new composite as the default:

```
sca_assignDefaultComposite('host', 'soaport', 'user', 'password',
'composite_name', 'composite_revision')
```

If the WLST command is successful, verify if the new composite is active. .

MANDATORY: If the new composite is not active, manually deploy the composite that failed, by following the steps in the Manually Deploying SOA Composites (page 10-33) section.

For more information, see the "sca_assignDefaultComposite" in the *WLST Command Reference for WebLogic Server* section.

8.4.3.2 Failed to Retire Previous Composite Revision

In case of receive an error in retiring the previous version of the composite, the old composite was not retired and both the new and old composites may be running. The old SOA composite was supposed to be retired so that only the new SOA composite would be active.

To resolve this error, use the following Oracle WebLogic Scripting Tool (WLST) command to retire the old composite:

```
sca_retireComposite('host', 'soaport', 'user', 'password', 'composite_name',
'composite_revision')
```

If the WLST command is successful, verify if the new composite is active. If it is not, manually deploy the composite that failed, by following the steps in the Manually Deploying SOA Composites (page 10-33) section.

For more information, see ”sca_retireComposite” in the *WLST Command Reference for WebLogic Server* section.

8.4.3.3 Custom Metadata and Key Flexfield Changes Are Not Propagated Across Clusters

If custom metadata and key flexfield changes are not propagated across clusters after applying a patch, custom and flexfield medatada has not been manually exported from a source system and migrated to the other systems.

Each SOA cluster maintains its own SOA MDS schema, which results in a duplicate set of metadata for each SOA cluster that must be synchronized. Patch Manager manages this synchronization, but any custom metadata or flexfield metadata must be manually exported from a source system and then migrated to the other systems.

8.4.4 Troubleshoot Complex Failures during SOA Patching

In case of the failures mentioned below are unable to resolve after following the steps in the Basic Troubleshoot for SOA Composite Failures (page 8-14) section, please file a service request with Oracle Support Services and provide the logs and information.

- **No Base Composite Has Been Deployed**

  An earlier revision of the SOA composite, which is being patched, is not currently deployed in the system. This could mean that the composite was not previously provisioned on the system. Therefore, the patch validation reports the following error:

  ```
  The base composite is not set as default composite. Suggestion: You must
  manually set the base composite as the default composite using the WLST
  command.
  ```

- **Failure at Preparation Step**

  The SOA composite fails during export actions, extract or attach plans, or merge updates, causing patch validation to the report the following error:

  ```
  Deployment of SOA composite [{0}] failed at preparation step. Reason: [{1}]
  Suggestion: You must manually deploy the composite using the WLST command.
  ```

- **Unknown Deployment Status**

The deployment of the composite reported an unknown deployment status, as shown by the following example message:

```
No information is available about the recovery status. The RecoverStatus
object obtained is null.
```

8.5 Troubleshoot Patching Sessions for Database Content

The AD Controller utility, `adctrl`, can monitor and control the progress of the workers called by AutoPatch to update database content and by AD Administration. For more information about workers, see the Worker Processes (page 2-7) section.

The following sections contain steps for troubleshooting issues that may occur during patching sessions for database content:

8.5.1 Start AD Controller

Follow these steps to start AD Controller:

1. Start AD Controller with the `adctrl` command as follows:

   `(UNIX) FA_ORACLE_HOME/lcm/ad/bin/adctrl.sh`

   Prompts are as follows:

   - Confirm the value of the Oracle Fusion Applications Oracle home
   - Specify an AD Controller log file. This log file is written to the current working directory. The default is `adctrl.log`.

2. After the main menu displays, enter a number to select an option. Press enter at any time to return to the AD Controller main menu.

8.5.2 Review Worker Status

When AutoPatch or AD Administration runs tasks in parallel, it assigns tasks to workers for completion. There may be situations that cause a worker to stop processing. To determine the status of workers, use AD Controller, as tracked in the

database, and manage worker actions. Also the status of workers can be found by reviewing the Log Summary. For more information, see the Log Summary (page 8-6) section.

Follow these steps to review the status of the workers from AD Controller:

1. Start AD Controller. For more information, see the Start AD Controller (page 8-20) section.
2. Review worker status.

   Select **Show worker status** from the AD Controller main menu. AD Controller displays a summary of current worker activity, as tracked in the database. The summary columns are:

   - Control Code: The last instruction from the manager to the worker
   - Worker: The worker number
   - Context: The general action the manager is executing
   - Filename: The file the worker is running, if any
   - Status: The status of the worker

     The following table describes the status that may be assigned to a worker., as follows:

Table 8-4 Worker Status

Status	Meaning
Assigned	The manager assigned a task to the worker and the worker has not yet started.
Completed	The worker completed the task and the manager has not yet assigned it a new task.
Failed	The worker encountered a problem.
Fixed, Restart	Fix the problem, and the worker will retry the task that failed.
Restarted	The worker is retrying a task or has successfully restarted a task. Note that the status does not change to Running.
Running	The worker is running a task.
Wait	The worker is idle.

If the worker shows a status of Failed, refer to the Determine Why a Worker Failed (page 8-21) section for assistance in fixing the problem so Patch Manager can complete its processing. In certain situations, a worker may have a status of "Running" or "Waiting", but the worker has terminated unexpectedly. To verify the worker status, check whether the worker process exists by using the command that is appropriate for the operating system.

8.5.3 Determine Why a Worker Failed

A worker usually runs continuously in the background and when it fails to complete the job it was assigned, it reports a status as Failed. When a worker fails its task, the next step is review the worker log files to determine what caused the failure, without wait until the other workers and the manager stop. Workers do not proceed to run tasks in the subsequent phase until all tasks in the current phase complete successfully. An

immediate action has to be taken to resolve the failure so the workers can continue to run tasks in the next phase. If the task was deferred after the worker failed, there may be no action required.

The first time a task fails, the manager defers the task and assigns the worker a new task. If the deferred task fails a second time, the manager defers it a second time only if the run time of the task is less than 10 minutes. If the deferred task failed a third time, or if its run time is greater than 10 minutes, the task stays at a failed status and the worker waits for manual intervention. Action is then required because the worker stops any further processing. An example of when this scenario can occur is during seed data upload. The seed data upload may fail due to records being locked by another process. If the lock is released before the second or third attempt of the upload, the upload succeeds.

Follow these steps to find out why a worker failed:

1. In the Log Summary, located in FA_ORACLE_HOME/`admin/FUSION/log/logsummary_`*`fapmgr_command_timestamp`*`.html`, review the AutoPatch Apply log file to find the worker that failed. For more information, see the Log Summary (page 8-6) section.
2. Open and review the log file for the failed worker to determine the cause of the worker failure.

   Each worker logs the status of tasks assigned to it in `adpatch_apply_`*`timestamp`*`_worker`*`number`*.log. The worker log file contains information that describes exactly what task it was running and what error occurred that resulted in a failure. Review the worker log file for the failed worker to determine the source of the error. For more information, see the Log Files for Single Patch Manager Sessions (page 8-2) section.
3. Determine how to fix the problem that caused the failure.

   Resolve the error using the information provided in the log files. If needed, search for the resolution at the My Oracle Support site or file a service request with Oracle Support Services if you do not understand how to resolve the issue.
4. After the problem that caused the failure be resolved, restart the failed worker.

   Select **Tell worker to restart a failed job** from the AD Controller main menu. For more information, see the Restart a Failed Worker (page 8-22) section.
5. Verify the worker status.

   Select **Show worker status** from the AD Controller main menu. The Status column for the worker that failed should now display Restarted or Fixed, Restart.

8.5.4 Restart a Failed Worker

If a worker job failed or if some worker process are hanging, the issues that caused the worker status must be resolved, and the worker has to be manually restarted. Some worker processes spawn other processes called child processes. In case of terminate a child process that is hung, the worker that spawned the process shows *Failed* as the status. After fix the problem, restart the failed worker. After the worker restarts, the associated child processes start as well.

Follow these steps to restart a failed worker, after to resolve the cause of the failure:

1. Start AD Controller. For more information, see the Start AD Controller (page 8-20) section.

2. Select **Show worker status** from the AD Controller main menu.
3. Take the appropriate action for each worker status:
   - If the worker status is Failed and its job has failed, select **Tell worker to restart a failed job**. When prompted, enter the number of the worker that failed.
   - If the worker status is Running or Restarted, but the job is hung, follow the steps in the Terminate a Hung Worker Process (page 8-23) section.

8.5.5 Terminate a Hung Worker Process

When running AD utilities, there may be situations when a worker process appears to hang, or crash. If this occurs, check if the worker process exists by using the command that is appropriate for the current operating system. After verifying there is no worker process, use the adctrl option to change the status of the worker process to “Failed”. After changing the status of the worker process to “Failed”, restart that process manually.

If Oracle Fusion Applications Patch Manager hangs, monitor the progress of the patch on the console and wait until only the hung task is running and the remaining tasks either completed successfully or failed.

MANDATORY: A process that appears to be hung could be a long-running task. Be careful when terminating processes.

To terminate a process, start AD Controller, obtain the worker's process ID from the current operating system, and then stop any hanging processes. After making the necessary changes, restart the worker.

Take the following steps to terminate a worker process that is hung:

1. Start AD Controller. For more information, see the Start AD Controller (page 8-20) section.
2. Select **Show worker status** from the AD Controller main menu.
3. Open and review the log file for the failed worker to determine the cause of the worker failure.

   Each worker logs the status of tasks assigned to it in the `adpatch_worker`*`number`*`.log`. The worker log file contains information that describes exactly what tasks it runs and what errors occurred that resulted in a failure. Review the worker log file for the failed worker to determine the file being processed and review the worker log file to see if it is progressing. It is also possible verify whether the process is consuming CPU resources from your operating system.

   For more information, see the Log Files for Single Patch Manager Sessions (page 8-2) section.
4. Confirm that the operating system process associated with the worker is not running. If the task is identified as "hanging", determine the worker's process ID by looking for processes being run by the worker, as follows:

   ```
   (UNIX) ps -a| grep worker number
   ```
5. Determine what processes the worker has started, if any. If there are child processes, get their process IDs.

6. Stop the hung process, using the command that is appropriate for your operating system.
7. If a SQL*Plus session spawned by a worker, the SQL*Plus session. need to be killed . The worker immediately detects the FAILED state. For other processes, follow Steps 7 through 11.

   Verify that the worker process does not exist by using the command that is appropriate for the current operating system. Toverify there is no worker process, then use the adctrl option to change the status of the worker process to "Failed".

   In AD Controller, select **Tell manager that a worker failed its job** and enter the worker number of the hung workers. This should cause the worker to fail.
8. Select **Tell worker to quit**. When prompted, enter the worker number of the hung worker.
9. Select **Tell manager that a worker acknowledges quit**. When prompted, enter the number of the hung worker. This`adctrl` option allows exits out of the patching session cleanly.
10. Check for any active database connections by running the following query:

```
SELECT substr(S.MODULE,instr(S.MODULE,':')+1) as WORKER_ID,
       case when  S.PROGRAM like '%JDBC%' then
         'Java Worker' else 'C Worker' end as WORKER_TYPE,
       s.sid as DB_SESSION_ID,
       s.serial# as serial#
FROM GV$SESSION s
WHERE s.MODULE LIKE 'PATCHING_SESSION%'
AND ( s.PROGRAM LIKE '%JDBC%' OR s.PROGRAM LIKE '%adworker%');
```

    If the query returns any rows, terminate the active connection, using the command that is appropriate for your operating system.
11. Fix the issue that caused the worker to hang. Search the My Oracle Support site or file a service request with Oracle Support Services.
12. Select **Restart a worker on the current machine**. When prompted, enter the number of the failed worker, the worker will restart its assigned tasks and spawn the associated child processes. If the worker process is running, do not select **Restart a worker on the current machine**, doing so creates duplicate worker processes with the same worker ID and will cause incorrect results. When running Upgrade Orchestrator, patching is called in distributed mode to load database components, and workers are spawned on remote hosts. To restart a worker in this case, run `adctrl` on the machine where the worker is spawned. Information about the hosts and workers spawned in each of the host is available in the `fapmgr` log file.

8.5.6 Shut Down the Manager

There may be situations when AD utility must be shut down while it is running. For example, to shut down the database while Oracle Fusion Applications is running AutoPatch or during an AD Administration session. The shutdown should be performed in an orderly fashion so that it does not affect the data. To shut down the workers manually, use the following procedure:

1. Start AD Controller.
2. Select **Tell worker to quit**, and enter `all` for the worker number. The worker completes its current task and then quits.

3. Verify that no worker processes are running, using a command that is appropriate for your operating system.
4. When all the workers have shut down, the manager and the AD utility quits.

8.5.7 Reactivate the Manager

Oracle recommends to follow up the steps below in order to Reactivate the Manager, in cases where no workers are running tasks. They are either failed or waiting.
A restarted worker resumes the failed task immediately if the worker process is running. Workers change to a Waiting status if they cannot run any tasks because of dependencies on a failed task, or because there are no tasks left in the phase. When no workers are able to run, the manager becomes idle.

Complete the following steps for each worker:

1. Start AD Controller. For more information, see the Start AD Controller (page 8-20) section.
2. Determine the cause of the error.

   Select **Show worker status**. Review the worker log file for the failed worker to determine the cause of the error.
3. Resolve the error using the information provided in the log files.

   Search for the resolution in the My Oracle Support site or file a service request with Oracle Support Services in case of need more information about how resolve the issue.
4. Restart the failed worker.

   Select **Tell worker to restart a failed job** on the AD Controller main menu. The worker process restarts, causing the AD utility and the manager to become active again.

8.5.8 Resolve the Error "Unable to start universal connection pool"

This error occurs during patching, "Unable to start universal connection pool". Connections to the database cannot be established due to timeout limits.

Consider tuning the listener parameters `INBOUND_CONNECT_TIMEOUT_listenername` and `SQLNET.INBOUND_CONNECT_TIMEOUT` for the server.

For more information, see ”SQLNET.EXPIRE_TIME Parameter” and ”INBOUND_CONNECT_TIMEOUT Parameter" sections in the *Oracle Fusion Applications Performance and Tuning Guide*.

8.5.9 Resolve a Worker Blocked by a Session

When you patch database artifacts, your system should be in an idle state. If this is not the case, the patching session may hang due to locks. Examples of locks that can cause the patching session to hang are PL/SQL packages that are accessed by Oracle Enterprise Scheduler Service Server, a locked table, or a locked table row. After a specific wait time, such as 30 minutes, Patch Manager performs a check to determine whether the patching session is blocked by another session. If a blocking session is found, a message describing the block is sent to the log file, as shown in the following example:

```
"[2011-03-14T02:12:18.112-07:00] [apps] [NOTIFICATION] [] [AutoPatch] Worker 4 is
blocked by session 3868 ... Please fix the session to avoid indefinite waiting
```

The worker that is blocked remains in a Running status. To resolve the issue and release the lock, stop the blocking session using the command that is appropriate for your operating system. After the blocking session is no longer running, the hung worker proceeds to complete its task. To identify the sessions that are blocking patching sessions, following SQL*Plus query :

```
SELECT blocking_session,
       sid "Blocked Session",
       module "Blocked Module",
       seconds_in_wait
 FROM gv$session
 WHERE status = 'ACTIVE'
 AND module like 'PATCHING_SESSION:%'
 AND blocking_session_status = 'VALID'
  AND user = '<FUSION schema>';
```

8.5.10 Resolve an Error During Upload of Flexfield Data

When multiple seed data files are uploaded for the same flexfield but for different flexfield contexts, the upload tasks can fail due to locking issues. The failed tasks appear in the log file as the following error:

```
Loading failed with a JboException: JBO-25014: Another user has changed the
row with primary keyoracle.jbo.Key ...
```

AutoPatch defers any failed tasks to the end of the phase and reattempts the failed tasks only after attempting all tasks in the phase at least once. Usually, the flexfield seed data tasks that failed due to the locking issue succeed on subsequent attempts. If the flexfield seed data task succeeds on the retry, ignore these errors. If the task remains in a failed state, use the AD Controller utility to retry the failed task.

For more information, see the Restart a Failed Worker (page 8-22) section.

8.5.11 Understand the Impact of Automatic Conflict Resolution for Seed Data

If in the data log file notice values, such as `display_value_1`, this most likely means that Oracle Fusion Applications Patch Manager found conflicting seed data for a non-primary unique index, due to customer defined data. To avoid a failure during seed upload, a numeric value is added to the row with the duplicate value. If subsequently update the customer defined data so that it no longer conflicts with the Oracle defined data, then the next patch that delivers this data will remove the numeric value. Note that Oracle recommends not update Oracle defined data, as this results in marking the data as customized and prevents any future updates from being delivered to the customized row.

8.5.12 Set the Environment for Troubleshoot Database Issues

In case of the Oracle Fusion Applications database need to be connected to troubleshoot database related issues, by running SQL*Plus, for example, the appropriate environment variables needs to be set up . For setting any environment

variable, run the `adsetenv` script to generate the `APPSORA.env` file, which when sourced, sets all environment variables, as follows:

UNIX

```
sh adsetenv.sh
source APPSORA.env
echo $TWO_TASK
```

8.5.13 Resolve the Error "Initialization of OPatchFMWTarget failed"

This error occurs during patching, “Error while running deployment pre-reqs: Initialization of OPatchFMWTarget failed”.

In case of the Patch validation (Run level 63) fails with the following log snippet:

```
Error while running deployment pre-reqs: Initialization of
OPatchFMWTargetfailed.
```

Reason: There are no end point instances defined in <OPatchFAConfigInstance> object. Please define end point instances in <OPatchFAConfigInstance> object.Oracle Fusion Applications Patch Manager cannot proceed further with validation. Review the following steps for basic troubleshooting:

- Running the following SQL query should turn up at least one record:

  ```
  select count(*) from ad_patch_util_sessions where
  status='COMPLETED_WITH_WARNINGS';
  ```

- Apply apply data fix patch from net/den01fhc/scratch/sallamse/ p24811973_1111110_Fusion_GENERIC.zip (instructions included in the *readme.txt* bundled into the patch; the patch itself will be uploaded to ARU once certified).

8.6 Troubleshoot Patching Sessions for FASPOT

Usually FASPOT scripts ran successfully and fast, and the number of issues are minimal. When applying manually it is important to read every single patch `README` file carefully as it might contain special pre- and post-install instructions. Also, there might be requirements to rollback previously installed patches to resolve conflicts. In general these technology patches are applied by using opatch, adpatch and bsu (Weblogic). As mentioned before FASPOT automates this process.

The following sections contain steps for troubleshooting issues that may occur during patching sessions for FASPOT:

- Typos on faspot.properties file (page 8-27)
- Obsolete Parameters in faspot.properties File (page 8-28)
- FASPOT Script Directory (page 8-28)
- FASPOT on Alternate Platforms (page 8-28)

8.6.1 Typos on `faspot.properties` file

When *`faspot.properties`* presents any typo, run to target `prepare-local-env` as mentioned in the Prepare the Patch Staging Area (page 4-7) section.

8.6.2 Obsolete Parameters in `faspot.properties` File

In file `faspot.properties` there might be some parameters which are effectively obsolete like `OC_DOMAIN_HOME_LOC_LIST` .

File a service request with Oracle Support Services and provide the logs and information.

8.6.3 FASPOT Script Directory

Currently FASPOT script directory is copied to PATCH_WORK_DIR folder. It is NOT required to run FASPOT scripts from there.

8.6.4 FASPOT on Alternate Platforms

FASPOT has been developed for Linux 64-bit based installations of Fusion Apps. It is possible to run it on alternate platforms but some restrictions apply:

- The usage of *`faspot.sh`* does not work (instead of *`ant`* directly).
- The usage of a native OS specific JDK is required and according modifications in *`setEnv.sh`* have to be done manually.

9
Patch Manager Command Reference

This chapter describes the commands used to validate and apply patches, to run the Product Family Report, the Patches Applied Report and the Patching Report.
For more information, see the topics:

9.1 Validate Patches

The `fapmgr validate` command reads the actions in the patch driver file to determine whether a patch is compatible with your environment and can be applied successfully. It looks for the status of impacted servers, patch conflicts, and prerequisites, but it does not perform any updates. Validation can be performed in offline mode.

Patch Manager automatically performs patch validation when run the `fapmgr apply` command. The steps for validating a patch are provided here because Oracle recommends validate every patch before applying it, especially those patches that contain updates to SOA composites. The validation will reduce downtime and potential failures during patching. For more information about resolving these issues, see the Troubleshoot SOA Composite Validation Failures (page 8-16) section.

Validation performs the following actions:

- Checks if prerequisite patches have been applied.
- Checks whether required taxonomy details can be successfully retrieved.
- Checks whether the servers that are required for automated deployment of the Oracle Fusion Middleware artifacts in the patch are running.
- Checks whether an Oracle Fusion Middleware artifact will be copied based on version checking.
- Checks for patch conflicts.
- Determines if the patch can be applied in `hotpatch` mode, if `-hotpatch` is used.

Use the following syntax for the `validate` command:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh validate -patchtop (or grouptop)
patchtop_directory \
[-patchingplan path_to_patch_plan_xml_file][-hotpatch] [-taxonomyurl
hostname:portnumber]
[-logfile log_file_name][-loglevel log level]
```

The following table lists the options available for the `validate` command:

Table 9-1 validate Command Options

Option	Description	Mandatory
`patchtop`	Identifies the directory where the patch is unzipped	Yes, unless applying patches in a patch plan
`grouptop`	Identifies the directory where the patches in a patch plan are unzipped	Yes, when applying patches in a patch plan
`patchingplan`	Identifies the directory path to the patch plan XML file	Yes, when applying patches in a patch plan
`taxonomyurl`	Identifies the host name and port number that overrides the default taxonomy information stored in the environment properties file. The Administration Server passes this value to Patch Manager	Conditionally required only when the value present in the environment properties file has to be overridden and when using the online option
`hotpatch`	Determines if patch can be applied in `hotpatch` mode	No
`logfile`	Overrides the default log file name and sends the processing information to the file you specify, under the *APPLICATIONS_CONFIG/lcm/logs/<Fusion Applications Release Version>/FAPMGR* directory. If you enter an existing file name, the output is appended to the file	No, the utility generates a log file under *APPLICATIONS_CONFIG/lcm/logs/<Fusion Applications Release Version>/FAPMGR* using this naming convention: `FAPatchManager_validate_`*timestamp*`.log`
`loglevel`	Records messages in the log file at the level you specify. See the Oracle Fusion Applications Patch Manager Logging (page 8-1) section	No, default value is `INFO`
`help`	Displays help	No

9.2 Apply the Patches

To run the Oracle Identity Management Patcher, use the command line utility, `idmpatch`, located in the `$IDM_LCM_TOP/patch/bin` directory. Its shell script sets the environment and calls the utility. The following command shows the basic syntax for the `idmpatch` utility:

```
(UNIX) $IDM_LCM_TOP/patch/bin/idmpatch.sh run
```

OPTIONAL: To run only the prerequisites, use the `prereq` option . This will not stop and start the services or apply and rollback patches. The syntax to run the `idmpatch` is as follows:

```
(UNIX)$IDM_LCM_TOP/patch/bin/idmpatch.sh run -prereq
```

9.3 Product Families Report

The Product Families report provides a list of installed product families along with each associated Oracle Universal Installer (OUI) component name and version. The report

can run for all product families or specific product families can be selected. This report reads the local patch inventory and the current view snapshot.

The report includes the following information:

- OUI Component: Component name assigned to the product family
- Version: The version of the product family
- Product Family: The product family name
- Description: The product family description

9.4 Run the Patches Applied Report

Before running the Patches Applied report, ensure that the snapshot is current for the environment.

Use the following syntax to run the Patches Applied report:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -listpatches [optional
parameters]
```

The following table describes the parameters used by the Patches Applied report:

Table 9-2 Patches Applied Report Parameters

Parameter	Mandatory	Description
`comps`	No	Supply a comma-separated list of product families (components) desired on the report. The report includes all product families if you do not use this parameter
`outputfile`	No	Sends the report output to the file specified after this parameter. The existing file name cannot be reused. If this parameter is not use, no output file is created
`logfile`	No	Overrides the default log file name and sends the processing information to the file specified, under the *APPLICATIONS_CONFIG/lcm/logs/<Fusion Applications Release Version>/FAPMGR* directory. If an existing file name is entered, the output is appended to the file. If the paremeter is not use, the utility generates a log file under *APPLICATIONS_CONFIG*/lcm/logs/<Fusion Applications Release Version>/FAPMGR, using this naming convention: `FAPatchManager_report-listpatches_`*timestamp*`.log`
`loglevel`	No	Records messages in the log file at the level you specified. See the Oracle Fusion Applications Patch Manager Logging (page 8-1) section
`reportwidth`	No	Sets the column width to either 80 columns or 132 columns by specifying `NORMAL` or `WIDE`. The default value is 80 columns, or `NORMAL`

9.5 Run Patching Reports

Run `fapmgr report` command for generate Patch Manager reports. The patch-related information can be viewed from different perspectives to plan the patching strategy. These reports provide information that can be useful before and after of apply patches.

Note that the Online Patch Progress report is generated by Patch Manager automatically every five minutes during patch application.

The following table describes the patching reports that can be generated by Patch Manager:

Table 9-3 Patching Reports

Report Name	Report Option	Description	Variations
Patch Impact Report (page 3-6)	`-patchimpact`	Displays the impact of a patch in terms of bug fixes, prerequisites, and product families, by displaying what exists on your system. Also provides a list of artifact types, along with related servers and required manual actions	None
Product Families Report (page 9-2)	`-listcomps`	Displays a list of installed components (product families) and their versions	Specifies a list of product families or see all product families
Patches Applied Report (page 9-6)	`-listpatches`	Displays information about patches and bug fixes that have been applied to your system	Specifies a list of product families or see all product families
Patch Status Report (page 9-5)	`-isapplied`	Tells you whether specific patches or bug fixes were applied to your system	None
Diagnostics Report (page 8-7)	Not applicable, as this report is run automatically by Patch Manager	Online Patch Progress report displays the progress of a patching session that is currently running and the Diagnostics reports displays the same information at the end of the patching session	The progress report runs automatically every five minutes during a patching session and the diagnostics report runs at the end of the patching session

The `fapmgr report` command requires an option to specify which report has to run, followed by mandatory and optional parameters.

Use the following syntax to run a report:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -report_option
-mandatory parameters [optional parameters]
```

9.5.1 Run the Patch Status Report

Use the following syntax to run the Patch Status report:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -isapplied -bug or -patch
[comma-separated_list_of_patches/bug_fixes [optional parameters]
```

The following table describes parameters used by the Patch Status report:

Table 9-4 Patch Status Report Parameters

Parameter	Mandatory	Description
`bug`	Yes, unless the `patch` parameter is used	Supply a comma-separated list of bug fixes. In case of request language bug fixes, append the language code to the bug number, for example, `123456:KO`
`patch`	Yes, unless the `bug` parameter is used	Supply a comma-separated list of patches. In case of request language patches, append the language code to the patch number, for example, `123456:KO`
`outputfile`	No	Sends the report output to the file specified after this parameter. Do not use an existing file name. If this parameter is not used, no out file is created
`logfile`	No	Overrides the default log file name and sends the processing information to the file specified, under the *APPLICATIONS_CONFIG*`/lcm/logs/<Fusion Applications Release Version>/FAPMGR` directory. In case of enter an existing file name, the output is appended to the file. If this parameter is not used, the utility generates a log file under *APPLICATIONS_CONFIG*`/lcm/logs/<Fusion Applications Release Version>/FAPMGR` using this naming convention: `FAPatchManager_report-listpatches_`*timestamp*`.log`
`loglevel`	No	Records messages in the log file at the level specified. See the Oracle Fusion Applications Patch Manager Logging (page 8-1) section
`reportwidth`	No	Sets the column width to either 80 columns or 132 columns by specifying `NORMAL` or `WIDE`. The default value is 80 columns, or `NORMAL`

9.5.2 Patch Status Report

The Patch Status shows if specific patches or bug fixes have been applied to an environment. When running the report, a list of patch numbers or bug fixes is provided. The output indicates whether each patch or bug fix has been applied. This report queries the local patch inventory and current view snapshot.

The report output contains a table with the following columns:

- Bug Number: The bug number.
- OUI Component: Component name associated with the product family. This column displays `Not Applied` if the patch was not applied.
- Status: Possible values are `Applied` and `Not Applied`.
- Patch: The patch number. This column displays `Not Applied` if the patch was not applied.

- Date Applied: The date the patch was applied. This column is null if the patch was not applied.

9.5.3 Example Syntax for the Patches Applied Report

Examples of the command syntax for running the Patches Applied report follow:

How to show all patches applied and set the report width to 132

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -listpatches -reportwidth WIDE
```

How to show all patches applied for a list of product families

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -listpatches -comps
oracle.fusionapps.fin,oracle.fusionapps.crm
```

9.5.4 Patches Applied Report

The Patches Applied report provides information about patches that have been applied to an environment. It is possible to run the report for specific product families or all product families. This report depends on a current snapshot having been run.

The report is organized by product family (OUI component) and each product family section contains the following information:

- Patch Number: The patch number
- Patch Type: Possible values are `Standard` or `ONE-OFF`
- Date Applied: The date the patch was applied
- Bugs Fixed: The bug fixes included in each patch that was applied

9.5.5 Example Syntax for the Product Families Report

Examples of the command syntax for running the Product Families report follow:

How to show all installed product families and their versions

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -listcomps
```

How to show specific product families and specify the report output file name and log file name

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -listcomps -comps
oracle.fusionapps.crm, oracle.fusionapps.fin
-outputfile listproducts.txt -logfile /log/listproducts.log
```

9.5.6 Run the Patch Impact Report

The Patch Impact report can be run when applying only one patch or multiple patches downloaded in a patch plan. Before running the Patch Impact Report, ensure that the snapshot is current for the environment.

Use the following syntax to run the Patch Impact report for a single patch:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -patchimpact -patchtop
 path_to_unzipped_patch [optional parameters]
```

Use the following syntax to run the Patch Impact report for a single patch:

```
(UNIX)FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -patchimpact -
grouptoptop_directory_for_patches -patchingplanfull_path_to_patching_plan
[optional parameters]
```

9.5.7 Run the Product Families Report

Before to run the Product Families report, ensure that the snapshot is current for the environment.

Use the following syntax to run the Product Families report:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -listcomps [optional
parameters]
```

The following table describes the parameters used by the Product Families report:

Table 9-5 Product Families Report Parameters

Parameter	Mandatory	Description
comps	No	Supply a comma-separated list of product families (components) that want to see on the report. If this parameter is not used, the report will include all product families
outputfile	No	Sends the report output to the file specified after this parameter. Do not use an existing file name. If this parameter is not used, no output file is created
logfile	No	Overrides the default log file name and sends the processing information to the file specified, under the *APPLICATIONS_CONFIG*/lcm/logs/<Fusion Applications Release Version>/FAPMGR directory. If an existing file name is entered, the output is appended to the file. If this parameter is not used, the utility generates a log file under *APPLICATIONS_CONFIG*/lcm/logs/<Fusion Applications Release Version>/FAPMGR using this naming convention: FAPatchManager_report-listcomps_*timestamp*.log
loglevel	No	Records messages in the log file at the level you specify. See the Oracle Fusion Applications Patch Manager Logging (page 8-1) section
reportwidth	No	Sets the column width to either 80 columns or 132 columns by specifying NORMAL or WIDE. The default value is 80 columns, or NORMAL

9.5.8 Online Patch Progress Report and Diagnostics Report

During every patching session, the Online Patch Progress report is automatically generated so that you can view the results of the session. This report is updated every five minutes during the patching session. The report output is in HTML format so that it can be viewed from a browser and is located in the *APPLICATIONS_CONFIG*/lcm/logs/<Fusion Applications Release Version>/FAPMGR/directory. The file name

is `FAPMgrDiagnosticsSummary_mode_timestamp.html`, where *mode* can be `apply` or `validate` and `timestamp` is represented in *YYYYMMDDHHmmSS* format. After the completion of every patching session, the Online Patch Progress report is replaced by the Diagnostics report so that you can view the results of the session.

Both the Online Patch Progress report and the Diagnostics report contain the following sections, unless otherwise noted. The Online Patch Progress report is not available during application of multiple patches.

Module Execution Summary

The Module Execution Summary displays high-level information about the tools used during a patching session, such as Patch Manager, OPatch, and AutoPatch. For each tool, the report displays the following information:

- Module: Tool called during the patching session, such as Patch Manager
- Status: Completion status of the task, such as Success, Failed, or Skipped
- Duration: Total time that the module ran
- Start Time: Time and date the module started
- End Time: Time and date the module ended
- Log Files: Link to the Log Summary generated by Patch Manager. For more information, see the Log Summary (page 8-6) section.

Module Phase Summary

The Module Phase Summary displays summary information about tasks executed by Patch Manager. The tasks are summarized by each AutoPatch and OPatch phase and the following information is displayed:

- Mode: The patching mode is either Generic, Database or Middleware
- Phase: The name of the patching phase
- Duration: Total time the task ran
- Start Time: Time and date the task started
- End Time: Time and date the task ended
- Task Count: Total number of tasks within the phase
- Skipped: The number of tasks that were skipped during the phase
- Failed: The number of tasks that failed during the phase
- Completed: The number of tasks that completed successfully during the phase
- Percent Complete: The percentage of tasks that completed successfully during the phase

Tasks With Warnings or Failures

This section displays the following detailed information about each task that produced a warning or failed:

- Mode: The patching mode is either Generic, Database or Middleware
- Phase: The name of the patching phase and sub-phase
- Product Family: The short name of the product family, which displays only for database tasks.

- Task: The name of the artifact related to the task including the full path, and the domain, if applicable.
- Status: Completion status of the task, such as Failed or Skipped.
- Warning/Error Message: The error message is displayed if the task failed. Nonfatal messages appear as warning messages. The message also includes additional steps that are required to resolve the failure, if applicable.
- Log File: The name and location of the log file.
- Line Number: The line numbers within the log file that pertain to the task.

Module Task Details

The Module Task Details section displays the following detailed information about each task executed by Patch Manager:

- Mode: The patching mode is Database, Middleware, or Generic. In Generic mode, database validation and taxonomy URL validation are performed.
- Phase: The name of the patching phase, such as Patch Validation, Environment Validation, and Patch Application.
- Product Family: The short name of the product family, which displays only for database tasks.
- Task: The name of the artifact related to the task including the full path, and the domain, if applicable.
- Status: Completion status of the task, such as Success, Failed, or Skipped.
- Duration: Total time the task ran.
- Start Time: Time and date the task started.
- End Time: Time and date the task ended.
- Warning/Error Message: The error message is displayed if the task failed. Nonfatal messages appear as warning messages. The message also includes additional steps that are required to resolve the failure, if applicable.
- Log File: The name and location of the log file.
- Line Number: The line numbers within the log file that pertain to the task.

Tasks to be Completed (Online Progress Report Only)

The Tasks to be Completed section displays a summary of the tasks that have not yet been attempted by Patch Manager, and are waiting in the queue to run. The following information is displayed:

- Mode: The patching mode is Database, Middleware, or Generic. In Generic mode, database validation and taxonomy URL validation are performed.
- Phase: The name of the patching phase, such as Patch Validation, Environment Validation, and Patch Application.
- Product Family: The short name of the product family, which displays only for database tasks.
- Task: The description of the task that must be performed.

10

Manual Patching of Oracle Fusion Applications During Offline Patching

This chapter describes how Oracle Fusion Applications Patch Manager (Patch Manager) supports the patching of middleware and database artifacts. It also provides detailed steps for the manual deployment of artifacts, if needed.
The following topics are discussed:

10.1 Oracle Fusion Applications Patch Manager Middleware Artifact Support

The online mode of Oracle Fusion Applications Patch Manager (Patch Manager) supports most of the deployment actions required for patching middleware and database artifacts used by Oracle Fusion Applications. Depending on the type of artifact included in a patch, the artifact deployment may require manual actions. Some manual actions are required only if the patch is applied in offline mode, while others are always required. Before applying any patch, Oracle recommends to run the Patch Impact report to determine which artifact types are included in the patch and if manual actions are required by the patch. For more information, see the Patch Impact Report (page 3-6) section.

Table 10-1 (page 10-3) provides a quick reference that depicts how Patch Manager supports the Oracle Fusion Middleware artifacts that could be included in a patch. This table assumes that online patching is used unless otherwise specified.

An explanation of the information presented in this table follows:

- Automated Actions Performed by Patch Manager

  Patch Manager always copies the artifacts from the patch to the appropriate location on the system. This column describes additional actions that are performed automatically in online mode for each artifact.

- Actions to Be Performed Manually in Online Mode

  This column describes the actions that must be performed when the patch includes the specified artifact. These actions are described in more detail in this section.

- Actions to Be Performed Manually in Offline Mode and in the Case of Failures

  This column describes the actions that must be performed when the patch includes the specified artifact and the patch is applied in offline mode. If the patch is applied in online mode and there is a failure, these actions may also be required.

- What Must Be Running During Online Patching Mode and Manual Actions

  This column describes what must be running while applying the patch in online mode and when manual actions are performed.

After applying a patch, review the Diagnostics report to find out which manual steps are required for the artifacts included in the patch and where the artifacts were copied in *FA_ORACLE_HOME*. For more information, see the Diagnostics Report (page 8-7) section. For more detailed information about manual actions for each artifact, refer to the relevant sections in this chapter.

The following table describes the Oracle Fusion Middleware artifacts:

Table 10-1 Oracle Fusion Middleware Artifacts Supported by Oracle Fusion Applications Patch Manager

Artifact Type	Automated Actions Performed by Oracle Fusion Applications Patch Manager	Actions to Be Performed Manually in Online Mode	Actions to Be Performed Manually in Offline Mode and in the Case of Failures	What Must Be Running During Online Patching Mode and Manual Actions
Applications Help Content (AHC)	Stop and start the HelpPortalServer for the Common Domain	None	Stop and start the HelpPortalServer for the Common Domain	Domain Administration Server, Node Manager, database, HelpPortalServer
Applications Policies (`system-jazn-data.xml`)	Deploy using the `patchPolicyStore` silent install command for JAZN	None	Deploy using Oracle Authorization Policy Manager	Oracle Authorization Policy Manager, OPSS Security Store
B2B Metadata	Deploy trading partner agreements	None	Deploy agreements if the change needs to be implemented	Database
Oracle Business Intelligence Publisher (Reports and Captions)	Shut down the BI Presentation server, deploy to the Business Intelligence repository using Catalog Manager, and start the BI Presentation server after patching	None	Shut down the BI Presentation server, deploy to the Business Intelligence repository using Catalog Manager, and start the BI Presentation server after patching	None
Oracle Business Process Management (Oracle BPM) Template	Publish template to the Oracle Metadata Services (MDS) repository	None	Publish template to the MDS repository	Database
C Artifact	None	None	None	Database must be running. Oracle Enterprise Scheduler Service server must be down.
Common Resource (Activity Strings)	None	Stop and start all Managed Servers in all domains after patching	Stop and start all Managed Servers in all domains after patching	Administration Server, Node Manager, database
Data Security	Run the `DSDataMigrator` utility to reconcile GUID in LDAP	None	Run the `DSDataMigrator` utility to reconcile GUID in LDAP	OPSS Security Store, database

Table 10-1 (Cont.) Oracle Fusion Middleware Artifacts Supported by Oracle Fusion Applications Patch Manager

Artifact Type	Automated Actions Performed by Oracle Fusion Applications Patch Manager	Actions to Be Performed Manually in Online Mode	Actions to Be Performed Manually in Offline Mode and in the Case of Failures	What Must Be Running During Online Patching Mode and Manual Actions
Diagnostic Testing Framework JAR	None	None	None	None
E-Mail and Web Marketing (EWM)	Start and stop the relevant servers that host the Java EE application	None	Start and stop the relevant servers that host the Java EE application	Administration Server, Node Manager, database
Fatp JAR	Stop and start the relevant servers that host the Java EE application	None	Stop and start the relevant servers that host the Java EE application	Administration Server, Node Manager, database
Fatp C	None	None	None	Database must be running. Oracle Enterprise Scheduler Service server must be down.
Fatp directory	None	None	None	None
Flexfield	Stop and start the FNDSETUP Managed Servers and then deploy the flexfield	None	Stop and start the FNDSETUP Managed Servers and then deploy the flexfield	Administration Server, Managed Servers hosting `FndSetup` application, database
Group Space Template	Deploy template	None	Deploy template	WebCenter Managed Servers (`WC_Spaces`, `WC_Collaboration`), `ucm_server1`, OPSS Security Store, database
Image Routing (IPM)	Deploy to IPM servers	None	Deploy to IPM servers	See prerequisites in the Patch Image and Process Management (IPM) Artifacts in Offline Mode (page 10-14) section
Java EE	Stop and start the relevant servers that host the Java EE application	None	Stop and start the relevant servers that host the Java EE application	Administration Server, Node Manager, database

Table 10-1 (Cont.) Oracle Fusion Middleware Artifacts Supported by Oracle Fusion Applications Patch Manager

Artifact Type	Automated Actions Performed by Oracle Fusion Applications Patch Manager	Actions to Be Performed Manually in Online Mode	Actions to Be Performed Manually in Offline Mode and in the Case of Failures	What Must Be Running During Online Patching Mode and Manual Actions
Mobile Artifacts (MOBILE and MOBILESCRIPT)	Import MAF Artifact Archive (.maa) file into MDS	None	Start Common Domain Administration Server, CRM Domain Administration Server, SalesServer, and CRMCommonServer. Call the WLST command to import the mobile files.	Common Domain Administration Server, CRM Domain Administration Servier, SalesServer, CRMCommonServer
Oracle Data Integrator (ODI)	Import to ODI repository	None	Import to ODI repository	ODI repository import tool, database
Oracle Fusion Applications Patch Manager	None	Apply the patch with OPatch	Apply the patch with OPatch	None
Data Role Template (RGX)	Deploy the template	None	Deploy the template	Administration Server, Oracle Authorization Policy Manager, database
SOA Composite	Deploy and merge	Preserve any JDeveloper customizations	Deploy and merge	Administration Server, SOA-INFRA Managed Servers, database
SOAEXTENSION	None	Stop and restart all SOA-INFRA Managed Servers in all domains	Stop and restart all SOA-INFRA Managed Servers in all domains	None
SOA Resource Bundle	Deploy resource bundle and restart dependent composites	Reset SOA-INFRA `MBean` property if resource bundle contains human task-mapped attribute labels and standard view names	Deploy resource bundle and restart dependent composites	Administration Server, SOA-INFRA Managed Servers, Node Manager, database
SPE Inline Service	Deploy SPE files	None	Deploy SPE files	Oracle BI Server, database

10.2 Oracle Fusion Applications Patch Manager Database Artifact Support

The following table provides a quick reference that displays the Oracle Fusion Applications database artifacts that could be included in a patch. Database artifacts typically do not require manual actions be performed during online mode, as they are copied and deployed automatically in online mode. In offline mode, database artifacts are copied but they are not deployed. Before patching database artifacts, the database must be in an idle state with no locks being held on any of the database objects. All background jobs, including jobs in the database, must be terminated before patching to avoid locks on patched objects. There should not be any active processes, such as Oracle Enterprise Scheduler Service jobs running against the database. This is to prevent locking and other data issues during patching.

The following table describes the Oracle Fusion Database artifacts:

Table 10-2 Oracle Fusion Applications Database Artifacts Supported by Oracle Fusion Applications Patch Manager

Artifact Type	Description	Actions to be Performed Manually
Applications Seed Data (XML,XLIFF files)	Examples include static lists of values, functional or error messages, and lookup values. Any non-transactional data values loaded into the database can be considered seed data	Oracle recommends that patches containing seed data be applied from a machine that is co-located in the same subnetwork as the database server to maximize performance
Applications Database schema changes (SXML)	Examples include tables, triggers, views, sequences, synonyms, queues, queue tables, policies, and contexts	None
PL/SQL objects (pkh, pkb files)	Package headers and bodies	Manually shut down the Oracle Enterprise Scheduler Service servers before applying patches that contain PL/SQL changes
SQL scripts	Scripts that update the database	None

10.3 Patch Applications Help Content (AHC) Artifacts

Oracle recommends to patch AHC artifacts in online mode. When AHC artifacts are patched in online mode, no manual steps are required, except to ensure that the following are running:

- Administration Server for the Common Domain
- Node manager
- Database
- HelpPortalServer (In the case of Scaled out server any one of the servers must be up)

In offline mode, the impacted Managed Servers that host the `FndSetup.ear` application, such as the `HelpPortalServer`, must be manually stopped, patched, and restarted. To determine which product family is affected by the patch being applied, run the Patch Impact report. For more information, see the Patch Impact Report (page 3-6) section. The Patch Impact report indicates the domain and the corresponding managed server that will be impacted by this patch. Examples of artifacts in this category include the accompanying graphics and PDF files referenced by the content in the Seed Data Framework (SDF) files.

10.4 Patch Oracle B2B Metadata in Offline Mode

Oracle recommends to patch Oracle B2B metadata in online mode. When updates to Oracle B2B metadata are introduced in a patch, no manual steps are required in online mode to redeploy all Trading Partner Agreements that are affected by the metadata change.

When a patch containing Oracle B2B metada updates is applied in offline mode, and the change is implemented, Trading Partner Agreements that are affected by the metadata change must be manually redeployed. If the redeployment is not performed, the runtime continues to use the older metadata. The agreements can be deployed using the B2B User Interface (UI) or by running the `b2bdeploy` utility from the command line.

10.4.1 Deploy Agreements from the User Interface

To deploy all agreements from the UI, follow the steps in "Deploying an Agreement" in the *User's Guide for Oracle B2B*.

10.4.2 Deploy Agreements from the Command Line

To import the patched metadata and deploy the agreements, follow these steps:

1. Follow the steps in the *Prerequisites for Running the Command-line Tools* section in *User's Guide for Oracle B2B* with one exception. In Step 2 under "Create `jndi.properties`", this command must be used:

   ```
   cd FMW_HOME/soa/bin
   ```

   instead of this command:

   ```
   cd $ORACLE_HOME\bin
   ```

2. Export the entire repository for backup purposes. Before running the following command, set *`ANT_HOME`* to `/APPTOP/fusionapps/modules/org.apache.ant_1.7.1` and set the *`PATH`* to include *`ANT_HOME`*`/bin`.

   ```
   ant -f ant-b2b-util.xml b2bexport -Dexportfile=/local_directory/
   backup_export.zip
   ```

3. Import the patched export file.

   ```
   ant -f ant-b2b-util.xml b2bimport -Dexportfile=/local_directory/
   patch_export.zip -Dlocalfile=true -Doverwrite=true
   ```

4. Run the `b2bdeploy` command. If there is no Trading Partner Agreement found, this step is not needed.

```
ant -f ant-b2b-util.xml b2bdeploy -
Dtpanames="Agreement_name,Agreement_name"
```

5. If the patch introduces new documents for Trading Partner agreements, the document definition must be added.

10.5 Patch Oracle Business Intelligence Publisher Artifacts in Offline Mode

When the Oracle Business Intelligence Publisher (BI Publisher) artifacts (Reports and Captions) are patched in online mode, Patch Manager shuts down the BI Presentation Server before the patch applies and restarts it after successful patch application. No manual steps are required in online patching mode. If the shutdown of this server fails for any reason, the BI Publisher artifacts must be manually deployed.

Oracle recommends not to use offline mode when a patch contains BI Publisher artifacts. If applying a patch in offline mode is decided, the changes to the Oracle Business Intelligence repository must be manually deployed, in addition to stopping and restarting the BI Presentation server. These manual steps are required to keep the Oracle home and the Oracle Instance versions of the Oracle Business Intelligence Presentation Catalog synchronized. If these manual steps are not followed as described, subsequent patches containing BI Publisher artifacts may fail.

Perform the following steps to manually deploy BI Publisher artifact after applying a patch in offline mode:

1. The `opmn` process must be running. Follow these steps:
   a. To verify if the process is running, go to the *FA_ORACLE_HOME*`/instance/BIInstance/bin` directory and run this command:

      ```
      opmnctl status
      ```

   b. If the opmn process is not **Alive**, start it with this command:

      ```
      opmnctl start
      ```

2. Patch Manager must have copied one or more Oracle BI Presentation Catalog files into the Oracle home-based catalog.
3. Within the patch, there are some catalog *diff* files. These files are used with the Oracle Business Intelligence Catalog Manager tool to apply changes to a catalog. These changes must be applied to:
   - The run-time, or Oracle Instance, Oracle BI Presentation Catalog
   - The Oracle home Oracle BI Presentation Catalog
4. Special care must be taken when the patch being applied is a standard patch. With a standard patch, OPatch may choose to copy only a subset of the total files in the patch archive.

   Before performing the following steps, first determine exactly which files were actually copied to the Oracle home during the OPatch apply stage. Review the Patch Impact report to get this list of files. Capture the list of files from the messages sent to the console and to the `FAPatchManager_apply_`*timestamp*.log file.

   After the list is done, apply only the *diff* files that correspond to the files that were actually copied to the Oracle home. The *diff* files are named the same as

the original files, except they have a *.diff* extension added. If additional *diff* files are applied beyond the files that were actually copied to the Oracle home, then previous patch updates may be undone and the Oracle BI Presentation Catalog may be in an unsupportable state. Basically, a previous patch have been partially rolled back.

5. Shut down the BI Presentation server.
6. Unzip the middleware portion of the patch, the OPatch archive file, into a temporary location, such as `C:\patch`. To see which files are included in the patch, run the Patch Impact report. For more information, see the Patch Impact Report (page 3-6) section.
7. Using the example in the previous step, go to the `C:\patch\custom\scripts` directory.
8. Locate the Catalog Manager *diff* files listed in the subdirectories under the directory in the previous step. These files have *.diff* extensions. This is referred to as *`diff_file_location`* in subsequent steps.
9. Use Catalog Manager to apply each of these *diff* files to the Oracle home Oracle BI Presentation Catalog, using the following commands:
   a. Create the Catalog Manager patch file, as follows:

      ```
      oracle-instance/runcat.sh -cmd createPatch -inputFile diff_file_location
       -production webcat_location -outputFile webcat_patch.out -winsConflict
      latest
      ```

      - *`diff_file_location`* refers to the file from the previous step
      - *`webcat_location`* is the Oracle home Oracle BI Presentation Catalog location
      - *`webcat_patch.out`* is a temporary file created by this step and used in Step 5b

        Example of *`oracle-instance`* on Unix:

        ```
        APPLICATIONS_BASE/instance/BIInstance/bifoundation
        /OracleBIPresentationServicesComponent/coreapplication_obips1/
        catalogmanager
        ```

   b. Apply the Catalog Manager patch file, as follows:

      ```
      oracle-instance/runcat.cmd -cmd applyPatch -inputFile webcat_patch.out
      -outputFile  -persistNewApplicationsRoles webcat_applypatch.out
      ```

      - *webcat_patch.out* is the file created in Step 5a
      - *webcat_applypatch.out* is the output file from this deployment process
10. Repeat the previous step for the run-time catalog.
11. Restart the Oracle Business Intelligence system components using `opmnctl`:

    ```
    cd APPLICATIONS_CONFIG/BIInstance/bin/opmnctl
    ./opmnctl stopall
    ./opmnctl startall
    ```

For more information, see the "Starting and Stopping Oracle Business Intelligence" section in the *System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

10.6 Patching Oracle Business Process Management (Oracle BPM) Templates in Offline Mode

Oracle recommends to patch Oracle BPM templates in online mode. When updates to Oracle BPM templates are introduced in a patch, no manual steps are required in online mode to publish the new Oracle BPM Template to the Oracle Metadata Services (MDS) repository.

If a patch containing updates to Oracle BPM templates is applied in offline mode, the new Oracle BPM Template must be manually published to the Oracle MDS repository supporting the Oracle BPM Composer instance after the patch is applied. Use the `publish_template` WebLogic Scripting Tool (WLST) command from the WLST shell. The WLST `publish_template` command connects to the SOA MDS data using the `mds-config.xml` configuration file created. Provide the location of the `mds-config.xml` configuration file as one of the input parameters of the `publish_template` command.

Perform the following steps to create the `mds-config.xml` configuration file:

1. Copy the `mds-config-template.xml` file from the SOA server installation to a local directory, as follows:

   ```
   cp SOA_ORACLE_HOME/bpm/config/mds-config-template.xml /local_directory/mds-config.xml
   ```

2. Modify the following properties in the recently copied file to the temporary directory:
   a. Set `jdbc.userid` to the database user name of the SOA MDS database
   b. Set `jdbc.passwd` to the database password of the SOA MDS database
   c. Set `jdbc.url` to the connection URL of the SOA MDS database, for example, `jdbc:oracle:thin:@host2:1525:mds`
   d. Set `partition.name` to `obpm`

Perform the following steps to publish the new Oracle BPM template to the MDS repository:

1. Review the Diagnostics report to find the location of the archive file that contains the BPM template. For more information, see the Diagnostics Report (page 8-7) section.
2. Expand the archive that contains the new BPM template, so the `publish_template` command can find the template, as follows:
   a. Create or use an existing local directory.
   b. Untar the patched archive file, as shown in this example:

   ```
   cd /local_directory
   mkdir preboardWorker
   cd preboardWorker
   jar xf FA_ORACLE_HOME/hcm/deploy/
   bta_HcmCommonProcessesPreboardWorkerComposite.jar
   ```

3. Access the WLST shell, as follows:

   ```
   (UNIX) SOA_ORACLE_HOME/common/bin/wlst.sh
   ```

4. Deploy the Oracle BPM template, passing the temporary directory, `/local_directory/preboardWorker`, as the directory containing the template in the example in Step 2.

   Generic command syntax follows:

   ```
   publish_template(templateName, fsLocation, mdsconfigLocation, [Override],
   [oracleHome] )
   ```

   Note that the `publish_template` command simply updates the existing Oracle BPM template with a newer version. It has no impact on the projects deployed or instantiated from the existing template.

For more information about the `publish_template` command syntax, see the "publish_template" section in the *WLST Command Reference for WebLogic Server*.

10.7 Patch C Artifacts

When updates to C artifacts are introduced in a patch, no manual steps are required in either online or offline mode. Note that before applying C artifacts, all C executable files and the Oracle Enterprise Scheduler Service servers that host the C files must be shut down and the database must be running.

10.8 Patching Common Resource (Activity Strings) Artifacts

When updates to Common Resource artifacts are introduced in a patch, the Administration Server, Node Manager, and database must be running. Stop and restart all Managed Servers in all domains after the patch is applied.

10.9 Patch Customized Seed Data

A conflict between Oracle supplied seed data and customized seed data may occur when applying a patch that uploads seed data. The following types of conflicts may occur:

- When uploaded seed data conflicts with a primary unique index, the customer record is preserved and the Oracle seed data is not uploaded.
- When uploaded seed data conflicts with a non-primary unique index, a uniqueness violation occurs and an automatic conflict resolution feature prevents a seed data upload failure by appending numeric values, such as _1, _2, _3, to the display value.

For more information, see the Understand the Impact of Automatic Conflict Resolution for Seed Data (page 8-26) section.

10.10 Patch Diagnostic Test Framework (DTF) JAR Files

No manual steps are required when patching DTF artifacts in either online or offline mode.

10.11 Patch E-Mail and Web Marketing (EWM) Artifacts

Oracle recommends to patch EWM artifacts in online mode. When updates to EWM artifacts are introduced in a patch, no manual steps are required in online mode. In offline mode, follow the steps in mentioned in the Patch Java EE Artifacts (page 10-15) section.

10.12 Patch Flexfield Artifacts

Oracle recommends to patch flexfield artifacts in online mode. When flexfield changes are introduced in a patch, no manual steps are required to automatically deploy the flexfields in online mode, except to ensure that the following are running:

- Administration Server
- Managed Servers that host the `FndSetup` application
- Database

Users must log out and log in after a successful patch application to see the latest flexfield changes because flexfields reload upon user logout and login. If the changes to a flexfield are not implemented, it is possible to revert to a previous version of a flexfield. For more information, consult following links:

- Revert to a Previous Flexfield Definition After it is Updated by a Patch (page 8-11)
- Manually Deploying Patched Flexfields (page 10-12)
- Perform Flexfield NameSpaces Merge (page 10-13)

10.12.1 Manually Deploy of Patched Flexfields

Follow these steps when patching flexfields in offline mode to manually deploy the patched flexfield:

1. Ensure that the Administration Server and database are running.
2. Stop and start the `FNDSETUP` Managed Servers. For more information, see the “Starting and Stopping Components” section in the *Oracle Fusion Applications Administrator's Guide*.
3. Connect to the Oracle WebLogic Server Administration Server for the domain that hosts the `FndSetup` application. This is typically the Common Domain.
4. Run the `deployPatchedFlex()` WLST command. As this is run on a domain that hosts the `FndSetup` Application, this application within the parentheses do not have to be specified. However, the `FndSetup` application must be running for the command to succeed.

   Example:

   ```
   connect('weblogic' , 'weblogic1' , 't3://localhost:7101')
   deployPatchedFlex()
   ```
5. Review the report for errors.

Example of confirmation that flexfield changes were successfully deployed

As an example, assume that the patch delivered a new flexfield segment to the Calculation Defaults in Payroll Definitions. To confirm that the new flexfield segment was successfully deployed, follow these steps in the Payroll application:

1. From the Oracle Fusion Payroll application, select **Manage Payroll Definition**.
2. Click the **Create a New Payroll** icon.
3. Select a **Legislative Data Group**.
4. Confirm that the new flexfield segment appears under **Calculation Defaults**.

10.12.2 Perform Flexfield NameSpaces Merge

If a patch is applied in offline mode that includes flexfield changes that require a `NameSpaces` merge, the manual steps in this section must be manually performed, as follows:

1. Stop the managed servers that host the FNDSETUP application.
2. Run the following script: *`APPLICATIONS_BASE`*`/fusionapps/atgpf/atgpf/bin/flex_namespaces_merge.py`
3. Start the managed servers that host the FNDSETUP application.

10.13 Patch Group Space Templates

When a Group Space template is included in a patch, the patch introduces a new template with a version number attached, which is unlike other artifacts where the patched version replaces the existing one. If any customizations are on the template that the patched version will replace, the customizations must be manually incorporated in the new version of the template. If there are any settings or configurations that refer to the Group Space template name, ensure to update these to reflect the new template name. For all WebCenter services configured in a Group Space template, ensure that connections are configured properly.

Oracle recommends to patch Group Space templates in online mode. When updates to Group Space templates are introduced in a patch and the template included in the patch has not been customized, no manual steps are required in online mode, except to ensure that the following servers must be running:

- WebCenter Managed and Servers: `WC_Spaces, WC_Utilities`
- Oracle UCM Managed Server: `ucm_server1`
- LDAP Policy Store Server

10.13.1 Manually Deploy Group Space Templates

In offline mode, or in the case of failure, the new Group Space template must be manually deployed using the `importGroupSpace` WLST command.

1. Ensure that the following WebCenter Managed Servers are running:
   - WebCenter Managed Servers: `WC_Spaces, WC_Utilities`
   - Oracle UCM Managed Server: `ucm_server1`

- LDAP Policy Store Server

2. Access the WLST shell from the Oracle home where WebCenter is installed, using the following command:

   `(UNIX) WC_ORACLE_HOME/common/bin/wlst.sh`

3. Deploy the Group Space template, using the following command:

   ```
   importGroupSpaces('appName', 'fileName')
   ```

   The `appName` is always `webcenter` and the `fileName` is the name of the WebCenter archive file, from the patch to be imported. Refer to the Diagnostics report to get the full path and file name. For more information, see the Diagnostics Report (page 8-7) section.

10.14 Patch Image and Process Management (IPM) Artifacts in Offline Mode

Oracle recommends to patch IPM artifacts in online mode. When updates to IPM artifacts are introduced in a patch, no manual steps are required in online mode, other than ensuring all prerequisites are met. In offline mode, the IPM artifacts must be manually deployed.

Perform the following steps to manually deploy IPM artifacts:

1. The `opmn` processes must be running. Follow these steps:
   a. To verify if the process is running, go to the *FA_ORACLE_HOME*/`CommonDomain_webtier` directory and run this command:

      ```
      opmnctl status
      ```

   b. If the opmn process is not **Alive**, start it with this command:

      ```
      opmnctl start
      ```

2. The imaging application must be running. The format for the IPM URL is `http://` *host name:Port*`/imaging/`.
3. The Financials SOA server (`soa_server1`) must be running.
4. The Financials Payables Invoice and Expense Report SOA composites must have been successfully deployed and be in an active state.
5. The `FIN wsm-pm` application must be in an active state, which means the `FinancialCommon` server must be running.
6. The IPM to UCM connection, "Fusion Applications UCM Connection", must exist.
7. The IPM to SOA connection, "Financial SOA Connection", must exist.
8. The IPM Input should be set to **Offline** from the **Manage Inputs** section of the IPM UI. For example, select **Invoices** under **Manage Inputs** and then deselect **Online** under **Basic Information**.
9. Follow these steps to back up the existing IPM application definition:
   a. Log in to the IPM server as the IPM super user.
   b. From **Tools**, select **Export Definitions**.

c. Export the Oracle Fusion Payables Application and Expenses Application, all related searches, and inputs to a local file.

10. Review the Diagnostics report to find the location of the IPM artifacts that were copied to *FA_ORACLE_HOME*. For more information, see the Online Patch Progress Report and Diagnostics Report (page 9-7) section.

11. Access the WLST shell, as follows:

```
(UNIX) ECM_ORACLE_HOME/common/bin/wlst.sh
```

12. Deploy the IPM artifact, as follows:

```
connect(IPM Server user name,IPM Server password,IPM Server
hostname:port)
importIPMApplication(ipmAppFile,'Update',appDefName,'None');
importIPMInput(ipmAppFile,'Update',inputDefName);
```

Example:

```
connect('FAadmin','fusion','t3://IPMserver01.mycompany.com:17014');

importIPMApplication(exportFile='/net/server01/fusionapps/
applications/fin/ap/ipm/ApInvoiceIpmApp.xml',
action='Update', name='Payables Invoice Application','None')

importIPMInput(exportFile='/net/server01/fusionapps/applications/fin/ap/ipm/
ApInvoiceIpmApp.xml',
action='Update', name='Payables Invoice Input')
```

13. If applicable, perform the customizations on the new file, based on the exported file.

10.15 PatchJava EE Artifacts

Oracle recommends to you patch Java EE artifacts in online mode. When Java EE artifacts are patched in online mode, no manual steps are required, except to ensure that the following are running:

- Administration Server
- Node manager
- Database

In offline mode, manually stop, patch, and restart the impacted Managed Servers that host the Java EE application. To determine which product family is affected by the patch being applied, run the Patch Impact report. For more information, see the Run the Patch Impact Report (page 9-6) section. For example, if the Patch Impact report indicates that the patch updates a Java EE artifact in the Financials Domain, then the Financials Domain must be stopped, apply the patch, and then start the Financials Domain after the patch applies successfully. Examples of artifacts in this category include Oracle ADF Resource JAR files and Oracle Enterprise Scheduler Service MAR files.

10.16 Patch JEECONFIG Artifacts

Oracle recommends to patch JEECONFIG artifacts in online mode. When JEECONFIG artifacts are patched in online mode, no manual steps are required, except to ensure that the following are running:

- Administration Server
- Node manager
- Database
- Impacted Managed Servers

Also, the JEECONFIG application should be in an active state.

In offline mode, manually stop, patch, and restart the impacted Managed Servers that host the JEECONFIG application. To determine which product family is affected by the patch being applied, run the Patch Impact report. For more information, see the Run the Patch Impact Report (page 9-6) section. After bouncing the servers manually invoke the `CrmExtnUpgradeMBean` mbean from the Administration Server of the impacted domains.

For more information about stopping and starting servers, see the "Starting and Stopping an Oracle WebLogic Server Domain for a Product Family" section in the *Oracle Fusion Applications Administrator's Guide*.

10.17 Patching Mobile and Mobile Script Artifacts

Oracle recommends to patch Mobile and Mobile Script artifacts in online mode. When Mobile and Mobile Script artifacts are introduced in a patch, no manual steps are required to automatically import the mobile files into MDS in online mode, except to ensure that the following are running:

- Administration Server
- Database

To perform the validation for these artifacts, Fusion Applications Patch Manager must be run in online validate mode. In this mode, Patch Manager calls the corresponding python script from the patch to check for the availability of servers which are required to patch these artifacts. After the online validation completes successfully, the patch can be applied in online mode.

Manually Importing Mobile Artifacts into MDS

If the patch is applied in offline mode, perform the following steps to manually import Mobile artifacts into MDS:

1. Ensure the following servers are up and running:
   - Common Domain Administration Server
   - CRM Domain Administration Server
   - `SalesServer` (In the case of high availability servers, any one instance must be up)
   - `CRMCommonServer`
2. Confirm that Patch Panager copied one or more Mobile artifacts and the corresponding Mobiles Script artifact into the respective product family directory, as shown in the following examples:
   - Mobile Artifact - *`FA_ORACLE_HOME`*`/crm/mobile/deploy`
   - MobileScript Artifact - *`FA_ORACLE_HOME`*`/crm/mobile/script`
3. Run the following command to import the files:

```
$MW_HOME/oracle_common/common/bin ./wlst.sh FA_ORACLE_HOME/crm/mobile/script/
OracleSalesCloudMobileArchiveImport.py $FA_ORACLE_HOME wls_host_name
wls_admin_port_no wls_admin_user [apply|validate]
```

Provide the `wls_admin_password` on the command line. Note that the last parameter, ["apply" or "validate"] is optional. If no value is specified, the script runs in apply mode.

10.18 Patch Oracle Data Integrator (ODI) Artifacts

Oracle recommends to patch ODI artifacts in online mode. When updates to ODI artifacts are introduced in a patch, Patch Manager imports the ODI changes in online mode. If ODI artifacts are patched in offline mode, the changed ODI content must be manually imported to the ODI repository. In both online and offline modes, the ODI repository import tool and the database must be running.

Oracle Fusion Applications Provisioning does not install ODI Studio. ODI Studio must be installed before manually importing ODI changes. Therefore, ODI Studio must be installed before applying the patches that deliver ODI changes in offline mode or when a failed ODI import step is required to be manually retried in online mode.

Manually Importing ODI Changes

Oracle recommends that the ODI import be performed from a machine that is co-located in the same subnetwork as the database server to maximize performance. The following steps shows the instruction for the Manual Import ODI Changes:

1. Review the instructions in the patch README file to determine which ODI Project or Model must be deleted and imported again. The patch README file contains a list of the ODI files that are included in the patch, in the order that they must be imported.
2. Review the Diagnostics report to determine the location and file name for each ODI artifact that is to be imported. For more information, see the Online Patch Progress Report and Diagnostics Report (page 9-7) section.
3. Start the ODI Studio, as follows:

   ```
   (UNIX) odi.sh
   ```
4. Access the ODI Studio. Follow these steps:
   a. Select **View**, then **ODI Designer Navigator**.
   b. Click **Connect to Repository**.
   c. Log in using the super user name and password for the ODI repository.
5. Delete the Model or Project if specified in the patch README file. The README file specifies whether any Model or Project must be deleted and in what order.

   Right-click the Model or Project name and click **Delete**.
6. Import the ODI files in the order they are listed in the patch README file. Follow these steps:
   a. To import a project, right-click the Project name and click **Import**, then **Import Project**.
   b. To import a model, right-click the Model name and click **Import**, then **Import Model**.

c. Select **Synonym Mode INSERT_UPDATE** from the list in the Import Dialog window.

d. For the File Import directory, select the directory that contains the ODI file to be imported.

e. Select the ODI file to import.

f. Click **OK** to import.

Repeat Steps 5 and 6 for each Model or Project in the patch.

7. Close the ODI Studio after importing all the files in the order specified in the patch README file.

10.19 Patch Oracle Forms Recognition and Oracle Document Capture Artifacts

Oracle Forms Recognition (OFR) and Oracle Document Capture (ODC) artifacts are used only by Windows. When a patch that contains OFR and ODC artifacts is applied, Patch Manager backs up the customized files, if any, before copying the new files to *FA_ORACLE_HOME*., the OFR and ODC artifacts that were delivered in the patch must be manually installed. If the OFR and ODC have not been installed, refer to the Set Up Oracle Document Capture and Oracle Forms Recognition section in the *Oracle Fusion Applications Installation Guide* for the installation steps. Then return to this section for patching.

Install OFR Artifacts

If the `AP_Packaged_Project_1004a.ini` file was previously customized, then this file should not be installed from the patch as the patch delivers it. Instead keep the existing file and copy the file from the patch to a directory with a different name. Then compare the files and manually update the existing file with the changes to preserve the customizations, as follows:

1. Review the Patch Impact report to find the location of the files related to OFR.
2. Rename the file `AP_Packaged_Project_1004a.ini` to another name as a backup and copy `APPLICATIONS_BASE/fusionapps/fin/ap/ofr/AP_Packaged_Project_1004a.ini` to a new file on the Windows environment.
3. Move this `.ini` file to the OFR AP Project directory, which is located in the OFR directory structure, `Projects\AP\Global` (for example, `C:\Program Files\OFR\Projects\AP\Global`). If the existing `AP_Packaged_Projects_1004a.ini` file is in this location, rename it to a backup file.
4. Create a data link:

   a. Click the Windows **Start** menu button.

   b. Select **Run**.

   c. Enter **Notepad** into the **Open** field and click **OK**.

      Do not right click on the desktop to create a new file. Windows will assign a hidden file type to the file that will interfere with the following steps.

   d. Click **File - Save**.

   e. Navigate to the Desktop folder.

f. In the **File name** field, enter the following, including quote marks: "*odbc_dns*.udl". Substitute the actual ODBC data source name for *odbc_dns*. The data source name can be found by opening the **Control Panel**, then **Administrative Tools**, and **Data Source (ODBC)**.

g. Click **Save**.

h. Find the file on the desktop and double-click it to open the Data Link Properties dialog. If a text file is opened instead, go back and carefully follow the instructions for creating this file again.

i. Set **Data Source**: *ODBC data source name*

j. Select the **Use a specific user name and password** option.

k. Enter the READ-ONLY user name and password

l. Select the **Allow saving password** option.

m. Select **Test Connection**.

n. Click **OK**. An example of the file contents follows:

```
Provider=MSDASQL.1;Password=fusion;Persist Security Info=True;User
ID=fusion;Data Source=ffinde
```

5. Open the data link (`.udl` file) previously created, as follows:

a. Click the Windows **Start** menu button.

b. Select **Run**.

c. Enter **Notepad** into the **Open** field and click **OK**.

d. Open the `.udl` file.

e. From the `.udl` file, copy the entire string, starting with **Provider=**.

6. Open the `AP Packaged Project_1004a.ini` file (under C:\OFR_Projects\AP\Global) in another instance of Notepad and make the following changes:

a. Replace the connection string for the line starting with **SQL_VL_01_ConnectionString=**, with the text copied in Step 5.

b. Update attribute `ASA_VL_01_ImportODBCDSN` with the System DSN name of the Oracle Fusion Applications database.

c. Update attribute `ASA_VL_01_ImportODBCUser` with the READ-ONLY user name.

d. Update the attribute, `ASA_VL_01_ImportODBCPWD`, with the READ-ONLY account password.

e. Proceed with any additional customizations on the `.ini` file, as required by the implementation.

f. Verify the changes.

g. Save and close the file.

Update ODC Expenses Metadata

Perform the following steps to update ODC Expenses metadata:

1. Install the ODC Import-Export Utility from the Companion DVD, if this has not already been done.

2. Copy the ODC metadata ZIP files from `APPLICATIONS_BASE/fusionapps/fin/ap/odc` and `APPLICATIONS_BASE/fusionapps/fin/exm/odc` to a temporary directory in the Windows desktop environment.
3. Go to **Start**, then **Oracle Document Capture**, then **Import-Export Utility** and log in. The user name is `ADMIN` and the password is `admin`.
4. In the utility, go to **File** - **Import** or click **Import** and then import the metadata files one at a time. Ensure that all files are imported.

10.20 Patch Oracle Fusion Applications Patch Manager Artifacts

When updates to Oracle Fusion Applications Patch Manager are introduced in a patch, the patch must be applied with the OPatch utility. Oracle Fusion Applications is compatible with a specific version of OPatch instead of the generic version of OPatch. If an incompatible version of OPatch exists in *FA_ORACLE_HOME*, errors can occur while applying patches and running Upgrade Orchestrator. The compatible version of Opatch is available on My Oracle Support under patch 14044793.

During provisioning, the data model for Oracle Fusion Applications Patch Manager is updated by running the `fapmgr bootstrap` command. If the data model is updated again by a patch, the patch README file instructs to run the `fapmgr bootstrap` command.

Use this syntax to run `bootstrap`:

`(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh bootstrap [-logfile log_file_name] [-loglevel level]`

10.21 Patch Script Files

When updates to script files or control files are introduced in a patch, no manual steps are required in either online or offline mode. Note that before applying Scripts Control artifacts (SCRIPTSCTL), no processes should be running that use these scripts or control files.

10.22 Patch Security Artifacts

In Oracle Fusion Applications, the following artifact types related to security can be patched:

- Function Security Policies (Applications Policies, using the `system-jazn-data.xml` file)
- Data Security Grants (using Seed Data)
- Data Role (RGX) Templates
- JAR files secured by the Data Security Grants and Function Security Grants

A patch can contain one or more of these artifacts. This section describes the steps for applying security patches and recovering from patch failures. Examples of scenarios that may be needed when following the recovery steps include:

- Apply a security patch that introduces a set of new policies and LDAP GUIDs, after backing up the policy store and the applications database. The GUID reconciliation has not been performed with the applications database due to an unrelated database issue. To resolve the database issue, restore from the backup, resulting in the policy store containing extra GUIDs from the database and a synchronization issue between the policy store and the database.
- Apply a security patch that includes updates to the applications policies and the patch fails, resulting in a set of LDAP GUIDs not applying correctly.

The patching and recovery scenarios for the following combinations of security artifact patches are included:

- Patch Applications Policies (system-jazn-data.xml) (page 10-21)
- Patch Data Security Grants (page 10-23)
- Patch Data Role (RGX) Templates in Offline Mode (page 10-24)
- Patch Data Security Grants and Data Role (RGX) Templates (page 10-27)
- Back up the Data Security Store (page 10-30)
- Recover Data Security Seed Data from the Backup (page 10-30)

Oracle recommends to apply security patches in online mode and run the Patch Impact report to understand which artifacts are included in the patch. The Patch Impact report displays security artifacts as `JAZN`, `Data Security`, and `RGXTEMPLATE`. For more information, see the Patch Impact Report (page 3-6) section.

10.22.1 Patch Applications Policies (system-jazn-data.xml)

Oracle Fusion Applications uses the XML file, `system-jazn-data.xml`, to package function security policies through application roles, role hierarchies, grants, and policies. Function security policies are shipped as a `system-jazn-data.xml` file that resides in the Oracle home. After provisioning, these policies are migrated to an LDAP Policy store. Patching function security policies requires steps to absorb changes delivered by Oracle (`system-jazn-data.xml` in a patch), changes currently deployed, which include changes by the policies in the LDAP server, and `system-jazn-data.xml` contents previously shipped from Oracle (`system-jazn-data.xml` in the Oracle home).

Oracle Fusion Applications Manager runs a comprehensive analysis tool during patch validation to check for conflicts in applications policy changes before the patch is applied. If a change is considered *safe*, apply the patch in online mode. If a change is considered to be a *conflict*, follow the steps to apply the patch in offline mode, which includes manually resolving conflicts. The following table describes a summary of changes that are safe and those that cause a conflict:

Table 10-3 Changes to Applications Policies

Type of Change	Safe - Apply Patch in Online Mode	Conflicts - Apply Patch in Offline Mode
Additions	New artifacts shipped from Oracle	Artifacts retained by Oracle in a patch with or without modifications, but deleted by the customer

Table 10-3 (Cont.) Changes to Applications Policies

Type of Change	Safe - Apply Patch in Online Mode	Conflicts - Apply Patch in Offline Mode
Modifications	Artifacts modified by Oracle in a patch but not by the customer	• Artifacts modified by Oracle in a patch and by the customer • Artifact created by both Oracle in a patch and by the customer, using the same identifier, but with some other differences
Deletions	All artifact deletions must be applied in offline mode	• Artifacts deleted by Oracle in a patch and not touched by the customer • Artifacts deleted by Oracle in a patch and modified by the customer • Artifacts deleted by Oracle in a patch, and where the customer created new references to the Oracle deleted artifact in their system. Examples include but are not limited to permission and resource grants, entitlements grants, role inheritance relationships, and entitlements to resource associations

This section contains information about the following methods for patching applications policies:

- Prerequisites for Patching Applications Policies in Online Mode (page 10-22)
- Patch Applications Policies in Offline Mode using APM (page 10-22)

10.22.1.1 Prerequisites for Patching Applications Policies in Online Mode

Oracle recommends to patch applications policies in online mode because Patch Manager automates the deployment of the `system-jazn-data.xml` file by running the `patchPolicyStore` silent install command.

Ensure that the following steps are completed before patching applications policies in online mode:

1. Validate the patch in online mode and ensure that the validation output does not contain any conflicts. For more information, consult the Validate Patches (page 9-1) section and the Table 10-3 (page 10-21) section.

   If the validation reports any conflicts, then choose to apply all safe changes in online mode. Later, apply conflicting changes in offline mode, as described in the Patch Applications Policies in Offline Mode using APM (page 10-22) section.
2. All domains that use the OPSS Policy store in Oracle Internet Directory for authorization policies must be shut down before the patch applies.

10.22.1.2 Patch Applications Policies in Offline Mode using APM

All domains, except the OPSS Security Store and the domain that hosts APM, must be shut down before JAZN patching and restarted after JAZN patching. The following steps must be performed if applications policies are patched in offline mode using APM:

1. Run the Patch Impact report to see which artifacts are included in the patch. For more information, see the Patch Impact Report (page 3-6) section. The location where the `system-jazn-data.xml` is located in the patch, because this location is requested in Step 4.
2. Run Patch Manager to apply the patch, which copies system-jazn-data.xml from the patch to the Oracle home in offline mode.
3. Log in to Authorization Policy Manager.
4. Open the Policy Upgrade Management tab as follows:
   a. Perform the analysis
   b. Before applying a patch, take off line any WebLogic domain that uses the Security Store where the application policies to be patched reside
   c. Backup the Security Store by using Oracle Platform Security Services `migrateSecurityStore` to export the Security Store into a replica of it. Now the patch can be applied.

   When the application to patch from the pull-down Application list is selected, choices such as the following should appear:

   - `fscm_system-jazn-data.xml: FSCM stripe`
   - `crm_system-jazn-data.xml: CRM stripe`
   - `hcm_system-jazn-data.xml: HCM stripe`
   - `bip_jazn-data.xml: OBI stripe`
5. Analyze Patch Differences and Resolve Patch Differences and, if there are errors during this step, restore the backup, as mentioned in the step 4.
6. Restart all Oracle Fusion Applications domains.
7. Oracle delivers changes to `system-jazn-data.xml`in its own patch. Related code change patches, if any, should be applied only after all of the steps in this section complete successfully.

The following document provides additional information related to subjects discussed in this section:

- For more information about general troubleshooting for Patch Manager, see the Monitor and Troubleshoot Patches (page 8-1) section.
- For more information about security in Oracle Fusion Applications, see the Secure Oracle Fusion Applications in the *Oracle Fusion Applications Administrator's Guide*.

10.22.2 Patch Data Security Grants

Oracle recommends to patch data security grants in online mode. In both online and offline patching mode, ensure that the prerequisites are met. The topics related to Patch Data Security Grants area as follows:

- Prerequisites for Patching Data Security Grants (page 10-24)
- Patch Data Security Grants in Offline Mode (page 10-24)

10.22.2.1 Prerequisites for Patching Data Security Grants

Follow these steps to Patching Data Security Grants steps:

1. Run the Patch Impact report to see which artifacts are included in the patch. For more information, see the Patch Impact Report (page 3-6) section .
2. Back up the data security store by using the Oracle Database data pump export tool. For more information, see the Back up the Data Security Store (page 10-30) section.

10.22.2.2 Patch Data Security Grants in Offline Mode

Follow these steps to patch data security in Offline Mode:

1. Run Patch Manager to apply the seed data changes to the data security system. When data security changes are introduced in a patch, no manual steps are required in online mode to update the data security subsystem with the GUIDs of the new application roles seeded in the policy story. In offline mode or in the case of patch failure, manually run the `DSDataMigrator` utility as described in the Data is Missing After Migrating or Patching the Policy Store: Use DSDataMigrator to Reconcile GUIDs section in the *Oracle Fusion Applications Administrator's Guide*.
2. If there are any database errors during Step 1, such as running out of tablespace, fix the database errors that occurred and restart the patch.
3. If the errors that occurred while applying the seed data changes are not resolved, recover the seed data from the backup export file created in Step 2 of the prerequisites.

10.22.3 Patch Data Role (RGX) Templates in Offline Mode

Oracle recommends to patch data role templates in online mode. When data role template changes are introduced in a patch, no manual steps are required in online mode to deploy the changed templates.

Every data role template consists of two XML files. One is for data role generation and the other XML file is for data security policies generation. Both of these files must be deployed after a patch that contains changes to data role templates is applied, so that they remain synchronized with each other. Follow the steps in this section when applying a patch in offline mode that contains data role templates:

1. Run the Patch Impact report to see which artifacts are included in the patch. For more information, see the Patch Impact Report (page 3-6) section. Note that the Patch Impact report refers to data role templates as `RGXTEMPLATE`.
2. The following must be running while patching data role templates:
   - Administration Server
   - Oracle Authorization Policy Manager
   - Database
3. Run Patch Manager to copy the data role templates to *`FA_ORACLE_HOME`*.
4. To create a save point before deploying the data role templates, use the `createMetadataLabel` WLST command to label the MDS partition for `oracle.security.apm`, using the following syntax:

```
createMetadataLabel(application, server, name)
```

The following example creates the label `data_role_save_point` for the application `oracle.security.apm` deployed in the Administration Server:

```
createMetadataLabel('oracle.security.apm', 'AdminServer',
data_role_save_point')
```

5. Follow these steps to manually deploy the data role templates using the `importMetadata` WLST command against the Administration Server in the Common Domain for the `oracle.security.apm` application:

   a. Access the WLST shell, as follows:

   `(UNIX) SOA_ORACLE_HOME/common/bin/wlst.sh`

   b. Connect to WebLogic Server, as follows:

   ```
   > connect ('admin user name','admin user password','URL of the
   AdminServer')
   ```

   c. Deploy the data role template using the `importMetadata` WLST command. Refer to the Diagnostics report to find the directory where the template was copied by the patch. For more information, see the Diagnostics Report (page 8-7) section.

   Syntax for the `importMetadata` command follows:

   ```
   importMetadata(application='oracle.security.apm', server='Name of
   AdminServer',
   fromLocation='Directory in FA_ORACLE_HOME where data role templates were
   copied',
   docs='Path to the changed data role templates starting with APM
   partition')
   ```

   The following table displays the parameters required by the `importMetadata` command:

Table 10-4 Parameters for the importMetadata WLST Command

Parameter Name	Description
application	Enter the value of `oracle.security.apm`
server	Enter the name of theAdministration Server
fromLocation	Enter the absolute path to the directory in *FA_ORACLE_HOME* where the patch copied the data role templates. The path must not include the APM partition, because the APM partition is included in the next parameter, `docs`. The Diagnostics report provides the full path and file name in *FA_ORACLE_HOME* for each data role template that was copied from the patch
docs	Enter the directory for the APM partition, starting with `/oracle/apps/apm`, followed by the remainder of the path, which includes the data role template itself

Example for Importing the FinancialAssetBook.xml data role template

In this example, the `FinancialAssetBook.xml` data role template is located in this directory:

```
(UNIX)FA_ORACLE_HOME/fin/fa/apm/oracle/apps/apm/fin/fa/rgx/template
```

Example of the `importMetadata` command:

```
(UNIX)
importMetadata(application='oracle.security.apm', server='AdminServer',
fromLocation='FA_ORACLE_HOME/fin/fa/apm',
docs='/oracle/apps/apm/fin/fa/rgx/template/FinancialAssetBook.xml'
```

Example for Importing multiple XML files in one command by using a wild card in the docs parameter

The XML file for data role generation is located in this directory:

```
(UNIX) /net/machine1/oracle/apps/oracle/fin/gl/rgx/template/
GeneralLedger.xml
```

The XML file for data security policies generation is located in this directory:

```
(UNIX) /net/machine1/oracle/apps/oracle/fin/gl/rgx/dataSecPolicy/
fndDataSecProvider/GeneralLedger.xml
```

The following command imports both XML files at the same time:

```
(UNIX) importMetadata(application='oracle.security.apm',
server='AdminServer',
fromLocation='/net/machine1', docs='/oracle/apps/oracle/fin/gl/**'
```

6. If there are any errors in Step 5, follow these steps to recover by restoring the data role templates. Proceed to Step 7 if there are no errors in Step 5.
   a. Promote the MDS label created in Step 4 using the following command:

      ```
      promoteMetadataLabel(application, server, name)
      ```

      The following example promotes the metadata label `data_role_save_point` to the `oracle.security.apm` application deployed in the Administration Server:

      ```
      promoteMetadataLabel('oracle.security.apm',
      'AdminServer','data_role_save_point')
      ```

   b. Delete any new data role templates that were delivered in the patch, using the following command:

      ```
      deleteMetadataLabel(application, server, name)
      ```

      The following example deletes the data role templates in the metadata label `data_role_save_point` from the `oracle.security.apm` application deployed in the Administration Server:

      ```
      deleteMetadataLabel('oracle.security.apm','AdminServer','data_role_save_p
      oint')
      ```

7. Assuming Steps 3 through 5 are successful, Oracle recommends to preview the execution of the changed data role templates. Run the preview from the **Summary** tab after the data role template is opened from the APM console.

   If the preview results are not correct, follow the recovery described in Step 6 to restore the data role templates. Otherwise, proceed to Step 8.

8. Run the changed data role template and confirm that the data roles and grants are generated correctly. Use the APM role templates summary for reconciliation of the generated artifacts.

9. If all steps are successful, delete the MDS label created in Step 4, using the `deleteMetaDataLabels` command:

   ```
   deleteMetadataLabel(application, server, name)
   ```

   The following example deletes the metadata label `data_role_save_point` from the `oracle.security.apm` application deployed in the Administration Server:

   ```
   deleteMetadataLabel('oracle.security.apm','AdminServer','data_role_save_point
   ')
   ```

10.22.4 Patch Data Security Grants and Data Role (RGX) Templates

Oracle recommends to patch data security grants in online mode. Follow the steps in this section when a patch contains both data security grants and data role templates. Every data role template consists of two XML files. One is for data role generation and the other XML file is for data security policies generation. Both of these files must be manually deployed after a patch is applied, so they remain synchronized with each other.

Perform the following steps to patch data security grants and data role templates:

1. Run the Patch Impact report to see which artifacts are included in the patch. Note that the Patch Impact report refers to data role templates as `RGXTEMPLATE`.
2. Back up the security store by using the Oracle Database data pump export tool, as described in the Back up the Data Security Store (page 10-30) section.
3. The following must be running when patching data security grants and data role templates:
   - OPSS Security Store
   - Administration Server
   - Oracle Authorization Policy Manager
   - Database
4. Run Patch Manager to apply the seed data changes to the data security system and to copy the data role templates to *`FA_ORACLE_HOME`*.

   When data security changes are introduced in a patch, no manual steps are required in online mode to update the data security subsystem with the GUIDs of the new application roles seeded in the policy story. In offline patching mode or in the case of patch failure, manually run the `DSDataMigrator` utility as described in Data is Missing After Migrating or Patching the Policy Store: Use DSDataMigrator to Reconcile GUIDs in the *Oracle Fusion Applications Administrator's Guide*.
5. If there are any database errors during Step 4, such as running out of tablespace, fix the database errors that occurred and restart the patch.

   If the errors that occurred while applying the seed data changes are not solved, recover the seed data from the backup export file created in Step 2.
6. To create a save point before deploying the data role templates, use the `createMetadataLabel` WLST command to label the MDS partition for `oracle.security.apm`, using the following syntax:

   ```
   createMetadataLabel(application, server, name)
   ```

The following example creates the label `data_role_save_point` for the application `oracle.security.apm` deployed in the Administration Server:

```
createMetadataLabel('oracle.security.apm', 'AdminServer',
data_role_save_point')
```

7. Follow these steps to manually deploy the data role templates using the `importMetadata` WLST command against the Administration Server in the Common Domain for the `oracle.security.apm` application:

   a. Access the WLST shell, as follows:

   ```
   (UNIX) SOA_ORACLE_HOME/common/bin/wlst.sh
   ```

   b. Connect to WebLogic Server, as follows:

   ```
   > connect ('admin user name','admin user password','URL of the
   AdminServer')
   ```

   c. Deploy the data role template using the following `importMetadata` WLST command. Refer to the Diagnostics report to find the directory where the template was copied, as follows:

   ```
   importMetadata(application='oracle.security.apm', server='Name of
   AdminServer',
   fromLocation='Directory in FA_ORACLE_HOME where data role templates were
   copied',
   docs='Path to the changed data role templates starting with APM
   partition')
   ```

   The following table displays the parameters required by the `importMetadata` command:

Table 10-5 Parameters for the importMetadata WLST Command

Parameter Name	Description
application	Enter the value of `oracle.security.apm`.
server	Enter the name of the Administration Server
fromLocation	Enter the absolute path to the directory in *FA_ORACLE_HOME* where the patch copied the data role templates. The path must not include the APM partition, because the APM partition is included in the next parameter, `docs`. The Diagnostics report provides the full path and file name in *FA_ORACLE_HOME* for each data role template that was copied from the patch
docs	Enter the directory for the APM partition, starting with `/oracle/apps/apm`, followed by the remainder of the path, which includes the data role template itself

Example of Importing the Financial AssetBook.xml data role template:

```
(UNIX)FA_ORACLE_HOME/fin/fa/apm/oracle/apps/apm/fin/fa/rgx/template
```

Example of the **`importMetadata`** command:

```
(UNIX)
importMetadata(application='oracle.security.apm', server='AdminServer',
fromLocation='FA_ORACLE_HOME/fin/fa/apm',
docs='/oracle/apps/apm/fin/fa/rgx/template/FinancialAssetBook.xml'
```

Example of importing multiple XML files in one command by using a wild card in the doc parameter. The XML file for data role generation is located in this directory:

```
(UNIX) /net/machine1/oracle/apps/oracle/fin/gl/rgx/template/
GeneralLedger.xml
```

The XML file for data security policies generation is located in this directory:

```
(UNIX) /net/machine1/oracle/apps/oracle/fin/gl/rgx/dataSecPolicy/
fndDataSecProvider/GeneralLedger.xml
```

The following command imports both XML files at the same time:

```
(UNIX) importMetadata(application='oracle.security.apm',
server='AdminServer',
fromLocation='/net/machine1', docs='/oracle/apps/oracle/fin/gl/**'
```

For more information, see the Diagnostics Report (page 8-7) section.

8. If there are any errors in Step 7, follow these steps to recover by restoring the data role templates. Otherwise, proceed to Step 9.
   a. Restore the security seed data from the backup created in Step 2. For more information, see the Recovery Data Security Seed Data from the Backup (page 10-30) section.
   b. Promote the MDS label created in Step 6 using the following command:

      ```
      promoteMetadataLabel(application, server, name)
      ```

      The following example promotes the metadata label `data_role_save_point` to the `oracle.security.apm` application deployed in the Administration Server:

      ```
      promoteMetadataLabel('oracle.security.apm',
      'AdminServer','data_role_save_point')
      ```

   c. Delete any new data role templates that were delivered in the patch, using the following command:

      ```
      deleteMetadataLabel(application, server, name)
      ```

      The following example deletes the data role templates in the metadata label `data_role_save_point` from the `oracle.security.apm` application deployed in the Administration Server:

      ```
      deleteMetadataLabel('oracle.security.apm','AdminServer','data_role_save_p
      oint')
      ```

9. Assuming Steps 2 through 7 are successful, Oracle recommends to preview the execution of the changed data role templates. Run the preview from the **Summary** tab after the data role template is opened from the APM console.

   If the preview results are not correct, follow the recovery described in Step 8 to restore the seed grants and data role templates. Otherwise, proceed to Step 10.

10. Run the changed data role template and confirm that the data roles and grants are generated correctly. Use the APM role templates summary for reconciliation of the generated artifacts.

    If the results are not correct, restore the database from the backup created in Step 2. For more information, see the Recovery Data Security Seed Data from the Backup (page 10-30) section.

11. If all steps are successful, delete the MDS label created in Step 6, using the `deleteMetaDataLabels` command:

    ```
    deleteMetadataLabel(application, server, name)
    ```

    The following example deletes the metadata label `data_role_save_point` from the `oracle.security.apm` application deployed in the Administration Server:

    ```
    deleteMetadataLabel('oracle.security.apm','AdminServer','data_role_save_point
    ')
    ```

10.22.5 Back up the Data Security Store

Back up the data security store by using the Oracle Database data pump export tool. Before running the export tool, ensure that the `TWO_TASK` environment variable is set to point to the Oracle Fusion Applications instance. Oracle Fusion Applications database user name and password are requested. Perform the following steps for Back up the Data Security Store:

1. For setting any environment variable, run the `adsetenv` script to generate the `APPSORA.env` file, which when sourced, sets all environment variables:

    ```
    (UNIX)
    sh adsetenv.sh
    source APPSORA.env
    echo $TWO_TASK
    ```

2. Run the data pump export tool as follows:

    ```
    ORACLE_HOME/bin/expdp directory=local_directory dumpfile=fndds1.dmp
    tables='(FND_GRANTS,
    FND_MENUS_TL,FND_MENUS,FND_MENU_ENTRIES,FND_COMPILED_MENU_FUNCTIONS,
    FND_FORM_FUNCTIONS_TL,FND_FORM_FUNCTIONS,FND_OBJECT_INSTANCE_SETS_TL,
    FND_OBJECT_INSTANCE_SETS,FND_OBJECTS_TL,FND_OBJECTS)' NOLOGFILE=y
    ```

The following document provides additional information related to subjects discussed in this section:

- For more information about Oracle Data Pump, see the Oracle Data Pump Overview page.

10.22.6 Recovery Data Security Seed Data from the Backup

Follow these steps only if a data security seed data patch failed and there is no way to resolve the failure and reapply the patch:

1. Remove the existing data security grant data from the data security tables. Connect to the `fusion` account using SQL*Plus and run the following commands:

    ```
    truncate table fusion.fnd_objects;
    truncate table fusion.fnd_objects_tl;
    truncate table fusion.fnd_object_instance_sets;
    truncate table fusion.fnd_object_instance_sets_tl;
    truncate table fusion.fnd_form_functions;
    truncate table fusion.fnd_form_functions_tl;
    truncate table fusion.fnd_menus;
    truncate table fusion.fnd_menus_tl;
    truncate table fusion.fnd_menu_entries;
    truncate table fusion.fnd_grants;
    ```

2. Import the data security seed data from the backup export file previously created, as follows:

```
ORACLE_HOME/bin/impdp dumpfile=fndds1.dmp tables='(FND_GRANTS,
FND_MENUS_TL,FND_MENUS,FND_MENU_ENTRIES,FND_COMPILED_MENU_FUNCTIONS,
FND_FORM_FUNCTIONS_TL,FND_FORM_FUNCTIONS,FND_OBJECT_INSTANCE_SETS_TL,
FND_OBJECT_INSTANCE_SETS,FND_OBJECTS_TL,FND_OBJECTS)'
NOLOGFILE=y
```

10.23 Patch Service-Oriented Architecture (SOA) Composites

When updates to SOA composites are introduced in a patch, no manual steps are required if *both* of the following conditions are met:

- SOA composite customizations are not available in Oracle JDeveloper. If customizations exist, follow the steps in the Preserve SOA Composite JDeveloper Customizations Before Applying a Patch (page 10-32) section.
- Apply the patch in online mode and no validation or deployment errors occurred during the application of the patch that contains SOA composites. Oracle recommends not to use offline mode when a patch contains SOA composites. If the patch fails while attempting to deploy a SOA composite, manually deploy the composite. For more information, see the Manually Deploy SOA Composites (page 10-33) section.

For information about resolving validation errors, see the Troubleshoot SOA Composite Validation Failures (page 8-16) section. For information about recovering from deployment errors, see the Troubleshoot SOA Composite Deployment Failures (page 8-18) section.

If SOA composites used by Oracle Fusion Applications are customized in JDeveloper, preserve these customizations before applying a patch that includes the next revision of the composite. Other customizations to the SOA composite being patched are automatically merged by the SOA deployment command called during patching. These runtime customizations, such as design time and run-time (DT@RT) changes or property changes, do not require a manual merge process.

What must be running when patching SOA composite:

- Administration Server
- SOA-INFRA Managed Servers
- Database
- At least one server must be running the Policy Manager component from the Web Services Manager (WSM-PM) application. Typically in an Oracle Fusion Applications environment, this is the Common Cluster, for example in the `CRMDomain`, it is the `CRMCommonCluster`. Find out which server is running by logging in to Fusion Applications Control to verify that the application named `wsm-pm` is running with an `OK` or **green** status.

10.23.1 Preserve SOA Composite JDeveloper Customizations Before Apply a Patch

If JDeveloper customizations were performed, not supported by OPatch, to a SOA composite and then the composite to the SOA runtime is deployed, subsequent patches are not directly deployable. The Patch Manager validation process returns the appropriate error, which instructs to take the newer version of the composite that is in the patch, redo the same customizations that were performed on the previous version of the composite, and then apply the patch in online mode to deploy the composite.

Perform the following steps to preserve SOA composite JDeveloper customizations before applying a patch:

1. Run Patch Manager validation in online mode to determine which composites have JDeveloper customizations. If any customizations are detected, the validation results display the SOA composite name, its location in the `patch_top` directory, and the requirement to merge JDeveloper customizations into the `sca_*.jar` file in the `patch_top` directory before applying the patch in online mode. For more information, see the Validate Patches (page 9-1) section. Run Patch Manager validation before applying every patch, especially patches that contain SOA composites. If the JDeveloper customizations are not merged into the `sca_*.jar` file in the `patch_top` directory, the deployment of the SOA composite that was changed inside the patch will fail when applying the patch.
2. Open the custom SOA workspace and the customized version of the Fusion Applications SOA composite in JDeveloper using "Oracle Fusion Applications Developer".
3. Import the composite `sca_*.jar` file from the `patch_top` directory into the project, for example revision `yy_patchnum`, in JDeveloper. Make note of this revision number in the deployment window because it will be needed it in Step 8. Find the revision number on the Patch Impact report.
4. Restart JDeveloper in the Oracle Fusion Applications Administrator Customization role.
5. Verify that there are no errors in JDeveloper.
6. Verify that the changes introduced in both the customized version and the patched version are present.
7. Right-click the composite project in the Application Navigator, select **Deploy**, select the composite, click **Deploy to SAR**, and click **Next**.
8. Manually change the value in **New Revision ID** to the revision from Step 3, for example, `yy_patchnum`, and click **Finish**.
9. If the deployment folder is set to a location different from that of the `patch_top` directory, copy and replace the JAR in the patch under *`patch_top`*`/patch_mw/files/`*`productfamily`*`/deploy`. If the file name is different, rename it to the original name.
10. Now validate and apply this patch successfully using Patch Manager in online mode.

For more information about customizing SOA composites, see the *Oracle Fusion Applications Extensibility Guide for Developers Guide* .

10.23.2 Manually Deploying SOA Composites

If a customized SOA composite deployment fails during patching, manually deploy this composite using WLST commands. Also manually deploy SOA composites if a patch is applied in offline mode that contains SOA composites.

Perform the following steps to apply a SOA composite manually after a deployment failure or when patching in offline mode:

> In the following steps, the example composite, `FinAp`, is patched from revision 1.0 to revision 2.0 and the SAR file of revision 2.0 is in *`FA_ORACLE_HOME`*`/crm/deploy/sca_FinAp_rev2.0.jar`. Note that the parameters are for illustration purposes only.

1. Refer to the Diagnostics Report (page 8-7) section, to find the name and location of the `sca_*.jar` file that was copied to *`FA_ORACLE_HOME`* by Patch Manager.
2. If the previous revision contained JDeveloper customizations, ensure that the patched revision is deployed with the merged JDeveloper customizations. Using the `sca_*.jar` file from Step 1, follow the JDeveloper customization merge instructions that are described in the Preserve SOA Composite JDeveloper Customizations Before Apply a Patch (page 10-32) section. Then use the merged sca_*.jar for Step 3.
3. Deploy the patched composite using the single patch composite command, as follows:

```
sca_patchComposite('SOA-Infra URL', user, password,
'/FA_ORACLE_HOME/crm/deploy//sca_FinAprev2.0.jar', mergeLogFile='/tmp/merge-log.txt')
```

10.24 Patch SOAEXTENSION Artifacts

When updates to SOAEXTENSION artifacts are introduced in a patch, stop and restart all SOA-INFRA Managed Servers in all domains. Both online and offline patching require this step.

10.25 Patch SOA Resource Bundles

Oracle recommends to patch SOA resource bundles in online mode. No manual steps are required when patching SOA resource bundles in online mode unless the SOA resource bundle JAR file contains translatable strings for human task-mapped attribute labels and standard view names, as indicated by a JAR name that ends with `FlexFieldSoaResource.jar`. In offline mode, in case of patch failure, or if the patch contains human task-mapped attribute labels and standard view names, manually deploy the SOA resource bundle and restart the SOA composites that depend on the SOA resource bundle.

The following must be running when SOA resource bundles are patched:

- Administration Server
- SOA-INFRA Managed Servers
- Node manager

- Database

After applying the patch, refer to the Diagnostics Report (page 8-7)section to get a complete list of composites that depend on each SOA resource bundle and also the domains.

Manually Deployment SOA Resource Bundle JAR Files

Perform the following steps to manually deploy SOA resource Bubdle JAR files:

1. From the Diagnostics report for patch validation, review the list of SOA resource bundle JAR files included in the patch and the domain where they should be deployed. Use the `ant-sca-deploy.xml` script to deploy the appropriate SOA cluster for each JAR included in the patch.

   Set the `ANT_HOME` variable:

   ```
   ANT_HOME=FA_ORACLE_HOME/modules/org.apache.ant_1.7.1; export ANT_HOME
   ```

   Deploy the appropriate cluster:

   ```
   ant -f Middleware_Home/SOA_Suite_Home/bin/ant-sca-deploy.xml
        -DserverURL=URL_to_SOA_server
        -DsarLocation=Location_of_resource_bundle_jar under FA_ORACLE_HOME
        -Duser=weblogic
        -Dpassword=weblogic_password
   ```

2. The Diagnostics report lists the composite affected by the patch and the domain where the composite is deployed. Follow these steps for each affected composite:

   a. Log in to in the domain where the composite is deployed.

   b. Go to ***domain name,*** then **SOA**, then ***soa-infra*** (SOA cluster name), then **default**, and then ***composite name.***

   c. Click **Shut Down**.

   d. Click **Yes** in the confirmation window.

   e. Click **Start Up**.

   f. Click **Yes** in the confirmation window.

3. Review the list of SOA resource bundle JAR files being patched.
   If a patch contains a JAR file with a name which starts with "jar_" and ends with "FlexFieldSoaResource.jar", for example, `jar_AppCmmnCompNotesFlexFieldSoaResource.jar`, perform the following steps to ensure that the patched resource bundle is reflected in the Oracle BPM Worklist. These steps describe how to set the `WorkflowCustomClasspathURL` MBean property to null, and then set it to `oramds:///apps/resource/` and apply the changes in Fusion Applications Control.

   a. Log in to Fusion Applications Control in the domain where the JAR was deployed.

   b. Go to **SOA**, then **soa-infra** in the left-hand panel. Go to **SOA Infrastructure**, then **Administration**, and then **System MBean Browser** in the right-hand panel.

   c. Go to **Application Defined MBeans**, then **oracle.as.soainfra.config**, then **Server: *SOA cluster name***, then **WorkflowConfig** and then **human-workflow**.

d. Remove the contents in the **Value** column of the `WorkflowCustomClasspathURL` attribute.

e. Click **Apply**.

f. Enter `oramds:///apps/resource/` in the **Value** column of the `WorkflowCustomClasspathURL` attribute.

g. Click **Apply**.

The following documents provide additional information related to subjects discussed in this section:

- For more information about the `ant-sca-deploy.xml` script that is used to deploy the SOA resource bundle, see "How to Manage SOA Composite Applications with ant Scripts" in the *Developing SOA Applications with Oracle SOA Suite*.
- For information about shutting down and starting up SOA composites in Oracle Enterprise Manager, see "Managing the State of Deployed SOA Composite Applications" in the *Administering Oracle SOA Suite and Oracle Business Process Management Suite*.

10.26 Patch Sales Prediction Engine (SPE) Inline Service Artifacts

Oracle recommends to patch data role templates in online mode. When updates to SPE Inline Service artifacts are introduced in a patch, no manual steps are required in online mode to deploy changed SPE artifacts.

Updates to SPE Inline Service are delivered in the `SPE_ILS.zip` file and the `AdfZspPredictionModuleSupportUtilities.jar` file. This section contains information about manual deployment of SPE artifacts, in the case of offline patching or failure during online patching.

SPE artifacts are provisioned only when Oracle Fusion CRM Performance Management is provisioned. If CRM Performance Management is not provisioned in the environment, do not deploy SPE artifacts.

Manually Deploying SPE Artifacts After Applying the Patch

Perform the following steps:

1. JDK 1.6 or later must be running.
2. Access to the ZIP file, `rtd-deploytool-11.1.1.zip`. This ZIP file resides inside another ZIP file, *FA_ORACLE_HOME*`/bi/clients/rtd/rtd_client_11.1.1.zip`.
3. Make a backup copy, in a temporary directory, of the existing `SPE_ILS.zip` file, which is located under *`FA_ORACLE_HOME`* in this directory:

   `(Unix) crm/components/crmPerformance/zsp/predictionmodule/inlineservice`
4. Oracle suggests to refer to the text file that was created when provisioning completed, which is a textual overview of the topology, as you follow these steps.
5. Stop and start `bi_server1` to include the changes in `AdfZspPredictionModuleSupportUtilities.jar`.
6. Follow these steps to deploy the new `SPE_ILS`.zip artifact:

a. Unzip `rtd_client_11.1.1.zip` in a temporary directory. To find this file, refer to Step 2 in this section.

b. In the unzipped files, go to the folder `.../client/CommandLineDeploy` and find `rtd-deploytool-11.1.1.zip`.

c. Unzip `rtd-deploytool-11.1.1.zip` and go to the folder, `.../OracleBI/RTD/deploytool`.

d. In this folder, open a command terminal. Ensure having the JDK class path set for this terminal.

e. Run this command:

```
java -jar deploytool.jar -deploy
-server Host name of the server where BI domain is created
-port Managed server port where the OracleRTD app is deployed
-terminateSessions true Full directory path to SPE_ILS.zip
```

Example:

```
(UNIX) java -jar deploytool.jar -deploy -server server01.oracle.com -port
7001 -terminateSessions true FA_ORACLE_HOME/crm/components/ \
crmPerformance/zsp/predictionmodule/inlineservice/SPE_ILS.zip
```

f. When prompted, enter the user name and password to connect to the RTD server. This user must have a role that includes ILS deploy permission. Both the `BIAdministrator` and `BIAuthor` have this permission.

g. This message indicates the deployment is complete:

```
deploymentStateId is 5Inline service "SPE_ILS"
in "FA_ORACLE_HOME/crm/components/crmPerformance/zsp/predictionmodule/
inlineservice/SPE_ILS.zip/SPE_ILS.zip"
deployed to server: "server01.oracle.com" port: "7001" deployment state:
"Development"
```

10.27 Patch Tree Artifacts

Tree artifacts are delivered as seed data in patches and therefore, do not typically require manual steps after they are patched. A process called *tree flattening* automatically runs during the patching process. If this process fails, perform the following additional steps:

1. To determine if the patch contains any files related to tree flattening, refer to the Patch Impact report and look for a file named `FndTreeVersionSD.xml`. This is the only file that requires tree flattening. For more information, see the Patch Impact Report (page 3-6) section.
2. Confirm that the tree version changes were successfully flattened by reviewing the worker logs for errors related to tree flattening. To determine the worker that executed the specific seed data task, based on the file name `FndTreeVersionSD.xml`, refer to the Diagnostics report generated at the end of the patching session. Note any tree versions that failed because the version numbers are needed to manually flatten the tree version changes.
3. Follow these steps to manually flatten tree versions:
   a. Access the administrative area of Oracle Fusion Functional Setup Manager by logging in to Oracle Fusion Applications with a user account that is provisioned with the necessary role. Contact the security administrator for details.

b. From the **Administration** menu in the work area of Oracle Fusion Applications, choose **Setup and Maintenance.**

c. Choose the **Manage Trees and Tree Versions** task.

d. Search for the tree versions that require flattening.

e. Choose the appropriate tree version and optionally choose **Audit** from the **Actions** menu to diagnose the issues.

f. If changes to the tree version need to be made, click the tree version and edit it.

g. Choose **Flattening**, **Row Flattening**, then **Flattening, Column Flattening** from the **Actions** menu to flatten the selected tree version.