Oracle® Fusion Applications

Patching Guide

11*g* Release 7 (11.1.7)

E16602-23

October 2013

Documentation for installers and system administrators that describes how to use the patching framework tools to update and maintain Oracle Fusion Applications software between major releases.

ORACLE

Oracle Fusion Applications Patching Guide, 11*g* Release 7 (11.1.7)

E16602-23

Primary Author: Vickie Laughlin

Contributors: Subash Chadalavada, Lori Coleman, Rick Lotero, Jay Lu, Prashant Salgaocar, Venkatesh Sangam, Praveena Vajja, Anupama Pundpal

Contents

3 Using Oracle Fusion Applications Patch Manager

4 Patching Oracle Fusion Applications Artifacts

5 Introduction to Oracle Fusion Applications Languages

6 Patching Oracle Identity Management Artifacts

11 Monitoring and Troubleshooting Patches

Preface

This guide provides information about using the patching framework tools to update and maintain your Oracle Fusion Applications software between major releases.

Audience

This guide is intended for system administrators and patch administrators who are responsible for performing Oracle Fusion Applications patching tasks.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents:

- *Oracle Fusion Applications Administrator and Implementor Roadmap*
- *Oracle Fusion Applications Concepts Guide*
- *Oracle Fusion Applications Administrator's Guide*
- *Oracle Fusion Applications Installation Guide*
- *Oracle Fusion Applications Patching Guide*
- *Oracle Fusion Applications Customer Relationship Management Enterprise Deployment Guide*
- *Oracle Fusion Applications Post-Installation Guide*
- *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

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

What's New in This Guide

The following topics introduce the 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.

New and Changed Features for 11*g* Release 7 (11.1.7)

Oracle Fusion Applications 11*g* Release 7 (11.1.7), includes the following new and changed patching features for this document:

- Added Chapter 6, "Patching Oracle Identity Management Artifacts" describing the new Oracle Identity Management patching framework for Oracle Fusion Applications used to automate the Oracle Identity Management patch application.
- Added steps in Section 3.6, "End-to-End Process for Applying Individual Patches" and Section 3.7, "End-to-End Process for Applying Multiple Patches" to run the Health Checker utility before and after applying patches.
- Updated path to the `defaults.txt` file. See Section 10.4.1, "Creating a Defaults File".

Other Significant Changes in this Document for 11*g* Release 7 (11.1.7)

For 11*g* Release 7 (11.1.7), there are no other significant changes.

1

Introduction to Oracle Fusion Applications Patching Framework

The Oracle Fusion Applications patching framework provides the tools to support updates to Oracle Fusion Applications software. This chapter introduces the Oracle Fusion Applications patching framework and its components.

This chapter contains the following topics:

1.1 Maintaining Oracle Fusion Applications

Oracle Fusion Applications is a deployment of applications product offerings built on the Oracle Fusion Middleware technology stack and Oracle Database. Each application artifact and the underlying Oracle Fusion middleware engine has its own development life cycle and patching requirements. The primary purpose of the Oracle Fusion Applications patching framework is to simplify and expedite the maintenance of the code and functionality shipped as part of Oracle Fusion Applications.

Even though a variety of middleware technologies is involved, the patching framework transparently uses multiple tools and utilities. The tools used to maintain Oracle Fusion Applications and its underlying components vary based on the type and requirements of the components.

Maintaining Oracle Fusion Applications

Oracle Fusion Applications requires various maintenance actions during its life cycle. Those actions impact the installed applications, their middleware dependencies, and their database components. Maintenance actions include applying patches to fix bugs, adding new functionality and features, installing a maintenance release, or providing interoperability to new technology stacks. Oracle Fusion Applications Patch Manager can be used to apply patches, while RUP Installer can be used to add functionality that is not delivered in individual patches.

Adding and Maintaining Languages

In addition to helping you maintain the Oracle Fusion Applications core functionality, the patching framework helps you add and maintain language content. The provisioning process installs the American (US) English language. When you want to add a different language after completing the installation process, you apply a language pack with Oracle Fusion Applications Language Pack Installer.

For information about Language Pack Installer, see "Installing and Maintaining Oracle Fusion Applications Languages" in the *Oracle Fusion Applications Administrator's Guide*.

Maintaining Oracle Fusion Applications Patch Manager

When updates to Oracle Fusion Applications Patch Manager (Patch Manager) are available, you must use the OPatch utility to install these updates. If you need to uptake a new version of OPatch other than what is installed as part of Oracle Fusion Applications Provisioning and RUP Installer, you must use the OPatch version that is compatible with Oracle Fusion Applications.

For more information about OPatch, see the *Oracle Fusion Middleware Patching Guide*.

Maintaining Oracle Fusion Middleware Extensions for Applications

When updates to the Oracle home for Oracle Fusion Middleware Extensions for Applications are required, you use Oracle Fusion Applications AutoPatch to apply patches. For more information, see the applicable chapter:

- Chapter 7, "Patching Oracle Fusion Middleware Extensions for Applications"
- Chapter 8, "Patching Oracle Fusion Functional Setup Manager"
- Chapter 9, "Patching Oracle Fusion Applications Functional Core"

Maintaining Oracle Fusion Middleware Components

The Oracle Fusion Applications patching framework patches only the Oracle Fusion Applications content inside the Oracle Fusion Applications Oracle home. For more information, see Section 2.1.2, "Oracle Fusion Applications Oracle Home". When you must update middleware components, such as Oracle Enterprise Scheduler Service and Oracle WebLogic Service, use the OPatch utility. For more information, see the *Oracle Fusion Middleware Patching Guide*.

1.2 Oracle Fusion Applications Patching Framework Components

The Oracle Fusion Applications patching framework provides the following components, tools, and utilities to maintain and update Oracle Fusion Applications.

1.2.1 Oracle Fusion Applications Patch Manager

The primary function of Patch Manager is to manage patches that update both the database and middleware artifacts under the Oracle Fusion Applications Oracle home. It calls the Oracle Fusion Applications AutoPatch utility to patch database artifacts and the OPatch utility to coordinate activities for updating middleware artifacts. For conceptual information about Patch Manager, see Chapter 2, "Understanding Oracle Fusion Applications Patching Concepts". For information about how to use this product, see Chapter 3, "Using Oracle Fusion Applications Patch Manager".

1.2.2 Oracle Fusion Applications RUP Installer

RUP Installer allows you to upgrade your version of Oracle Fusion Applications to the next release. For more information, see the *Oracle Fusion Applications Upgrade Guide*.

1.2.3 Oracle Fusion Applications Language Pack Installer

Language Pack Installer allows you to add languages at any time after the initial provisioning and installation of Oracle Fusion Applications. For more information, see

"Installing and Maintaining Oracle Fusion Applications Languages" in the *Oracle Fusion Applications Administrator's Guide*.

1.2.4 AD Administration

The AD Administration utility performs maintenance tasks that keep applications files and database objects current. For more information, see Chapter 10, "Performing System Maintenance Tasks".

1.2.5 AD Controller

The AD Controller utility helps you troubleshoot database tasks performed by Oracle Fusion Applications Patch Manager or AD Administration and allows you to control their actions. For more information, see Section 11.5, "Troubleshooting Patching Sessions for Database Content".

1.2.6 HomeChecker

The HomeChecker utility verifies the correctness of any Oracle Fusion Applications Oracle home directory. For more information, see Section 10.5, "Running the HomeChecker Utility".

1.3 Oracle Fusion Applications Patching Framework Processes

The high-level processes related to patching Oracle Fusion Applications are outlined in Table 1–1, along with links contain the detailed instructions.

Table 1–1 Oracle Fusion Applications Patching Processes

Process	Description	Link to Documentation
Apply patches and run patching reports	Run Oracle Fusion Applications Patch Manager	Chapter 3, "Using Oracle Fusion Applications Patch Manager"
Deploy artifacts	Perform required deployments steps for specific artifacts	Chapter 4, "Patching Oracle Fusion Applications Artifacts"
Upgrade to Oracle Fusion Applications 11*g* Release 6	Run RUP Installer	Oracle Fusion Applications Upgrade Guide
Maintain languages	Install language packs and apply language patches	Chapter 5, "Introduction to Oracle Fusion Applications Languages"
Maintain Oracle Fusion Middleware Extensions for Applications	Run Oracle Fusion Applications AutoPatch	Chapter 7, "Patching Oracle Fusion Middleware Extensions for Applications"
Maintain Oracle Fusion Functional Setup Manager	Run Oracle Fusion Applications AutoPatch	Chapter 8, "Patching Oracle Fusion Functional Setup Manager"
Maintain Oracle Fusion Applications Functional Core	Run Oracle Fusion Applications AutoPatch	Chapter 9, "Patching Oracle Fusion Applications Functional Core"
Run maintenance tasks	Run AD Administration	Chapter 10, "Performing System Maintenance Tasks"
Monitor, verify, and troubleshoot the application of patches	Monitor Oracle Fusion Applications Patch Manager sessions and resolve any reported issues	Chapter 11, " Monitoring and Troubleshooting Patches"

2

Understanding Oracle Fusion Applications Patching Concepts

This chapter describes concepts that you should understand before you use the Oracle Fusion Applications patching framework.

This chapter contains the following topics:

- Patching Topology and Configuration
- Patch Types
- Patch Directory Structure and Contents
- Oracle Fusion Applications Patching and the Security Model

2.1 Patching Topology and Configuration

Provisioning a new Oracle Fusion Applications environment begins with a choice of the applications product offerings you intend to install and continues through configuring and deploying the applications. The patching framework must *know* about the configuration of the offerings and their middleware and database components to identify the artifacts and servers that are affected during patch application. The patching software is installed and configured with other system components during the provisioning process.

See "Applications Topology: Oracle WebLogic Server Domains" in the *Oracle Fusion Applications Installation Guide* for more information about installing, configuring, and deploying applications.

This section contains the following topics related to patching topology and configuration:

- Patching Installation and Configuration
- Oracle Fusion Applications Oracle Home
- Patch Top Directory
- Backup Copies of Patched Database Artifacts
- Oracle Universal Installer (OUI) Inventory
- Taxonomy URL

2.1.1 Patching Installation and Configuration

The provisioning process installs the artifacts required by patching. Then the process calls the patching configuration utility to configure the patching framework for the Oracle Fusion Applications system, as follows:

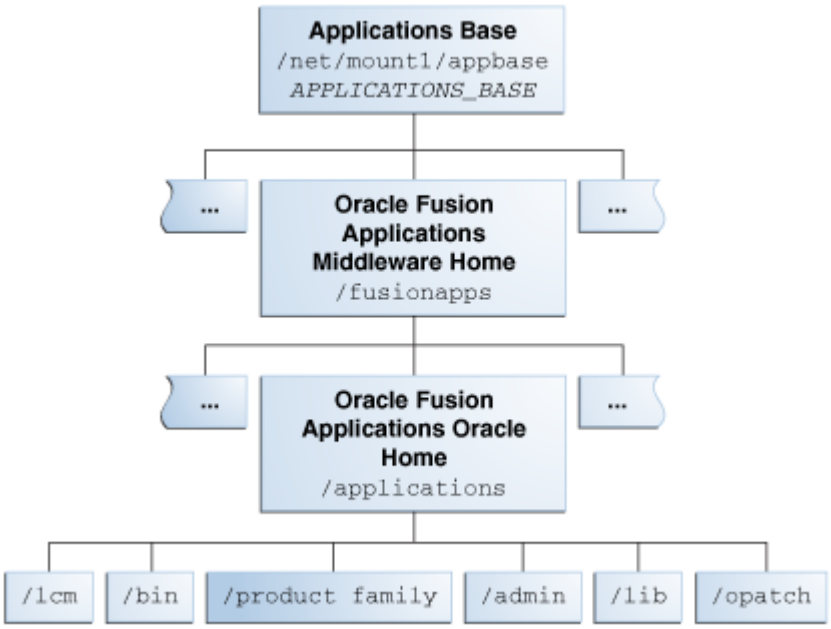
- Populates a properties file in the `admin` directory, `FUSION_env.properties`, that contains complete environment setup information required by the patching framework. This is the source of information that patching framework utilities use when setting up the environment for patching.
- Creates the patching framework configuration scripts that set the environment and call utilities. For example, it creates the script, `fapmgr.sh` in UNIX (`fapmgr.cmd` in Windows), which sets up the environment and then calls Oracle Fusion Applications Patch Manager (Patch Manager).

2.1.2 Oracle Fusion Applications Oracle Home

The patching framework and the Oracle Fusion Applications software are 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. Figure 2–1 shows the related directory structure, beginning with `APPLICATIONS_BASE`.

Note: Unless otherwise specified, the usage of "Oracle home" and *`FA_ORACLE_HOME`* in this guide refers to the Oracle Fusion Applications Oracle home.

Figure 2–1 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. 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. Note that you may have multiple versions of OPatch to support your enterprise software.

For more information about the components that are part of this directory structure, see "Provisioned Oracle Fusion Applications Home Directories" in the *Oracle Fusion Applications Administrator's Guide*.

Note: 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, RUP Installer, and Patch Manager.

2.1.3 Patch Top Directory

The patch top directory is any directory you select for downloading patch ZIP files. You unzip the patches in this directory and refer to this directory path as needed when applying patches. This directory is also called `patch_top` or `PATCH_TOP`. For example, if you download patch 1234567.zip into `/home/mypatches` and unzip it there, the patch top directory is `/home/mypatches/1234567`.

2.1.4 Backup Copies of Patched Database Artifacts

When applying a patch that includes a later version of an existing database artifact in the Oracle home, Patch Manager automatically backs up the existing database artifacts being replaced into a backup directory. The default location for the backup directory is `admin/pbackup` under the Oracle home. If needed, you can override this location by editing the `PATCH_BACKUP_DIR` parameter in the `FUSION_env.properties` file.

2.1.5 Oracle Universal Installer (OUI) Inventory

The Oracle Universal Installer inventory stores information about all Oracle software products and components installed in all Oracle homes. Each product, such as Oracle Fusion Applications, maintains its own local inventory and Oracle home. Local inventory files for Oracle Fusion Applications exist in the Oracle Fusion Applications Oracle home and the patching framework reads and updates these files. 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 and to perform patch validation, patch verification, and reporting. The location of the OUI Inventory directory, can be found at `/etc/oraInst.loc` (UNIX) or `C:\Program Files\Oracle\Inventory` (Windows).

For more information, see "Oracle Universal Installer Inventory" in the *Oracle Universal Installer and OPatch User's Guide*.

2.1.6 Taxonomy URL

Patch Manager queries the taxonomy MBean URL, as defined by the environment property called `taxonomy_url`, to determine which domains a specific patch impacts, such as 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. You can override this value 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, you can enter an overriding URL from the command line.

2.2 Patch Types

Oracle Fusion Applications patches typically contain one or more bug fixes. A *bug fix* is associated with a bug number, which is used by Oracle development to track fixes to the software. A *patch* is a delivery vehicle for releasing bug fixes to customers.

On occasion, patches may contain new features, test and diagnostic scripts, and additional documentation. For example, a standard patch focuses on solving specific issues and is applied using Patch Manager, while a language pack is applied with Oracle Fusion Applications Language Pack Installer and contains the translated content required to add a language other than English. The *patch type* describes the way the patch is packaged and applied. Patches are released in the types shown in Table 2–1.

Table 2–1 Patch Types

Format	Description	Utility Used
Standard Patch	A patch delivered to solve one or more customer issues. It may contain multiple bug fixes within a product family and includes the high-watermark of all related files or software artifacts included in the patch.	Oracle Fusion Applications Patch Manager
One-Off Patch	A patch containing a single bug fix for specific artifacts. It is created on an exception basis at the request of a customer for an issue that affects only that customer.	Oracle Fusion Applications Patch Manager

Table 2–1 (Cont.) Patch Types

Format	Description	Utility Used
Language Pack	Translation content for a language other than English for the entire Oracle Fusion Applications suite, for a specific release.	Language Pack Installer
Release Update Patch	A set of cumulative patches for the entire Oracle Fusion Applications Suite.	RUP Installer

2.2.1 Impact of a One-off Patch

Oracle may provide a one-off patch to fix a customer specific issue. A one-off patch is different from a standard patch because it contains only a single bug fix for each artifact included in the patch. A standard patch includes the high water mark of changes for the artifacts included in the patch.

A one-off patch is applied on an exception basis. After the one-off patch is delivered, Oracle provides a standard patch that includes the same fix as the one-off patch. When the standard patch is available, it replaces the one-off patch and should be applied to your environment as soon as possible.

Note: After you apply a one-off patch for a middleware artifact, your environment contains versions of artifacts that will conflict with any subsequent standard patch for that same component. Patch Manager prevents any new standard patches from being applied by setting a lock for that component. For example, after you apply a one-off patch for a database artifact, a lock is set for that artifact. To remove this lock, you must apply the standard patch to supersede the one-off patch. This standard patch delivers the same fix as the one-off patch, and also includes the high water mark for related artifacts.

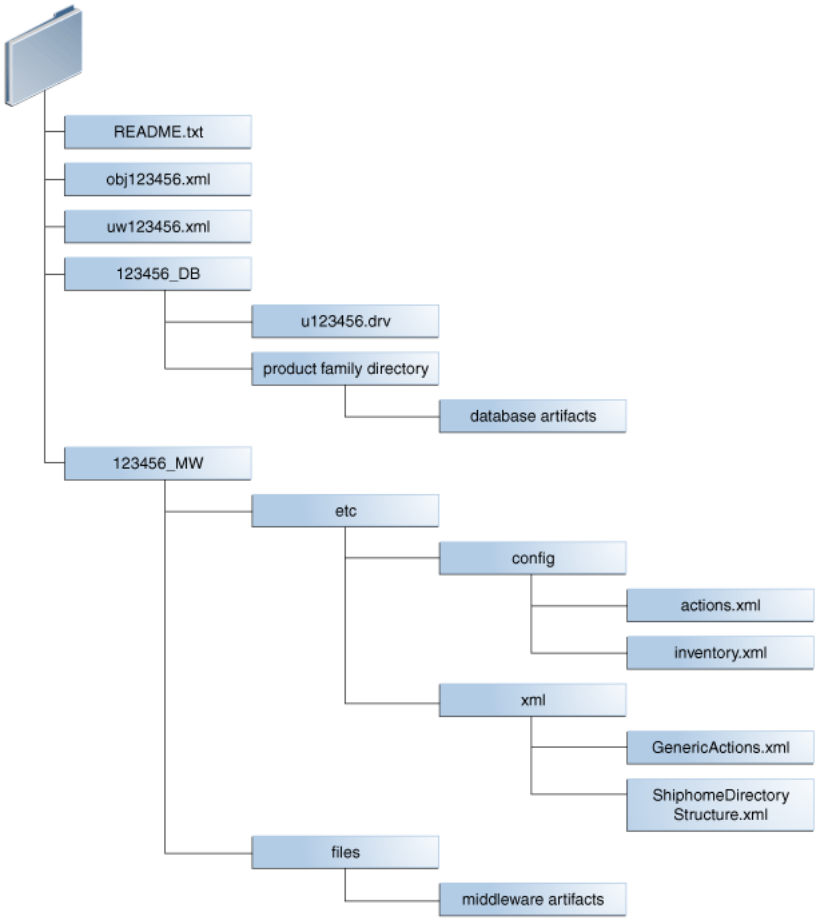
2.3 Patch Directory Structure and Contents

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.

2.3.1 Patch Directory Structure

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 Figure 2–2. If a patch contains only database artifacts or only middleware artifacts, the `123456_MW` directory or the `123456_DB` directory does not exist, respectively.

Figure 2–2 Example of the Directory Structure of a Patch



2.3.2 Patch Contents

Sample patch contents follow, using patch number 123456 as an example of a patch that contains both database and middleware artifacts:.

- `README.txt`: Provides general instructions for applying the patch and for performing manual steps, if required by the patch. If there are patches listed under "Other Patches" in the README file, you must download and apply them before you apply 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` file follows.

```
<?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/Adf
PjgTopPublicUi.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 follows:

```
<?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 follows:

```
<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/li
b" file_name="prj/deploy
/EARProjectsFinancials.ear/EARProjectsFinancials/WEB-INF/lib/AdfPjgTopPubli
c
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 follows:

```
<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 follows:

```
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 follows:

```
<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>
```

2.4 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. This user must be an owner of the Oracle Fusion Applications Oracle home.

For more information, see the *Oracle Fusion Middleware Application Security Guide*.

Obtaining Credentials

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

- 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.

Using 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.

No Password Prompts in Interactive Mode

Security can be breached when you are 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.

Removing Credentials 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. Likewise, Patch Manager does not read or write credentials from restart files or log files.

3

Using Oracle Fusion Applications Patch Manager

This chapter describes how to use the features of Oracle Fusion Applications Patch Manager.

This chapter contains the following topics:

3.1 Introduction to Oracle Fusion Applications Patch Manager

The primary function of Oracle Fusion Applications Patch Manager (Patch Manager) is to apply standard and one-off patches. It can also validate whether patches can be applied and generate patching reports.

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. Both AutoPatch and OPatch have long been used as the standard patching tools in previous releases of Oracle products.

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

For more information about patching Applications Core, see Chapter 7, "Patching Oracle Fusion Middleware Extensions for Applications".

The following topics provide additional information about Patch Manager:

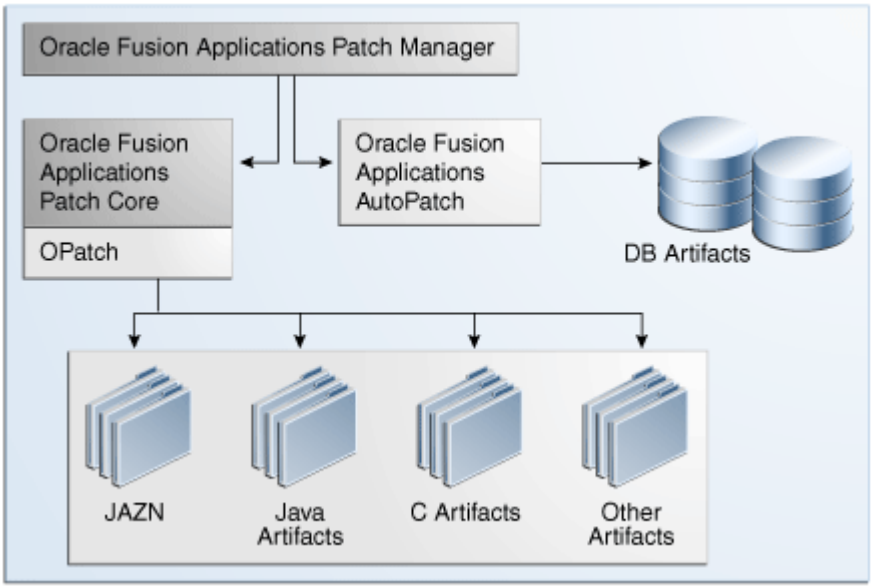
- Coordinated Patching
- Patching Oracle Fusion Middleware Artifacts
- Patching Database Artifacts
- Online and Offline Patching
- Applying Multiple Patches Using a Patch Plan

3.1.1 Coordinated Patching

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. If the patch contains both database and Oracle Fusion Middleware changes, Patch Manager coordinates the application of both changes, applying database changes first, followed by Oracle Fusion Middleware changes. You apply the patch in a single operation, regardless of the type of artifacts that are updated.

Figure 3–1 illustrates the patching process coordinated by Patch Manager.

Figure 3–1 Oracle Fusion Applications Coordinated Patching



Example of Patch Coordination

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

- 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.

3.1.2 Patching Database Artifacts

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 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. You must reduce the number of workers if the machine where you are applying the patch has a lower number of VCPUs when compared to those on the database server. To override the default number of workers when you apply a patch, specify the number of workers by using the `workers` option, as described in Table 3–3.

  The number of workers used for patching database artifacts also imposes a requirement on the open file descriptors configured for your 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. For more information, see "Increase the Open Files Limit" in the *Oracle Fusion Applications Installation Guide*.

  For more information about workers, see Section 3.1.2.1, "Worker Processes".
- **Patch validation:** Validates whether the database portion of the patch is compatible with your environment and can be applied. If the patch is not valid, the patching session fails. 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.
- **Application of the patch:** Copies the database artifacts to the Oracle home and then makes changes in the Oracle Fusion Applications database using the updated files.
- **Compilation of invalid objects:** Compiles all invalid objects in the database. For more information, see Section 3.1.2.3, "Compiling Invalid Objects".
- **Consolidation of log files:** Collects the patching results and location of log files for reporting purposes.

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 must be 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.

For more information, see Section 3.1.2.1, "Worker Processes" and Section 3.1.2.2, "File Version Verification".

3.1.2.1 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 that 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.

3.1.2.2 File Version Verification

This AutoPatch feature enables incremental running of database-related actions in a patch so that only the actions that have not already been performed against your database are actually run. The first time a database update action runs, the version of the file used to update the database is recorded in tables in the applications database. The next time AutoPatch applies a patch containing that action, it compares the version of the last file run against the current version of the file in the patch. It runs the action only if the version in the patch is newer than the last version run.

3.1.2.3 Compiling 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.

3.1.3 Patching Oracle Fusion Middleware Artifacts

Oracle Fusion Middleware artifacts may be deployed to Managed Servers running on different nodes, but the artifacts are copied to the Oracle Fusion Applications Oracle home that is shared by the different nodes. When a patch contains updates to Oracle Fusion Middleware artifacts, such as Java EE applications or Service-Oriented Architecture (SOA) composites, Patch Manager coordinates the following tasks during online patching:

- Patch preparation: Sets up credentials and other necessary environment variables.
- Patch validation: Validates the patch against the patch inventory under the Oracle home and domain to ensure that the patch is compatible and the prerequisite patches can be applied.
- Topology discovery: Queries the taxonomy tables to find information relevant to the artifacts contained in the patch, such as domains, hosts, the Administration Server URL and Oracle homes. If the query returns exceptions, patching may proceed or fail, depending on the artifact type.
- Application of the patch: Calls OPatch to copy the Oracle Fusion Middleware artifacts from the patch to the Oracle Fusion Applications Oracle home. Also deploys those artifacts to the appropriate run-time container so that they are usable, such as deploying a SOA composite to the appropriate SOA server. Note that all artifacts in the patch are copied, but deployment actions occur only for

those product families that have been deployed during the provisioning process. Finally, tracks the success or failure of each patching action and performs validation as needed, based on the directives specified in the patch metadata.

- Post-patch apply actions: Starts and stops the Managed Servers impacted by the patch in the respective domains, only in online mode. Performs the deployment of certain Oracle Fusion Middleware artifacts.
- Consolidation of log files: Collects the patch results and location of log files for reporting purposes.

For more information, see Table 4–1, " Oracle Fusion Middleware Artifacts Supported by Oracle Fusion Applications Patch Manager".

3.1.4 Online and Offline Patching

Patch Manager supports two patching modes: *online* and *offline*. To apply any patch in online mode, the Administration Servers must be running. For both offline and online modes, the database is still running, but it should be idle. Oracle WebLogic Server is not connected to the database for performing any transactions, so no requests from users should be processed. As the patch administrator, it is your responsibility to ensure that there are no active transactions or processes running during patching.

3.1.4.1 Online Mode

The primary difference between online and offline modes is that 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. For more information, see Section 4.1, "Oracle Fusion Applications Patch Manager Middleware Artifact Support". When patching in online mode, Patch Manager provides messages about the steps you must take after the patch is applied, to resolve any failures that occurred during the post-apply tasks.

To enable online patching mode, you specify the `online` option when you run Patch Manager. You must also use the `stoponerror` option, as described in Table 3–3. 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 Section 2.1.6, "Taxonomy URL".

Note that this automation feature attempts to stop and start only those impacted servers that are running. 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. During online patching, all servers and applications are running, but they must be idle. Applications login should be restricted and no Oracle Fusion Applications functions should be available to users during online patching. If you prefer to start and stop your Managed Servers using your own process, you can apply online patches in offline mode.

3.1.4.2 Offline Mode

After you apply a patch in offline patching mode, you must 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. Before you apply the patch, you can run the Patch Impact report to see which servers will be impacted by the patch. For information about server management for offline patching, see "Starting and Stopping a Product Family Oracle WebLogic Server Domain" in the *Oracle Fusion Applications Administrator's Guide*.

To minimize downtime, you could choose to leave servers running and start and stop the servers impacted by the patches after the patching session ends. In offline mode, all applications are unavailable to users, but only the servers impacted by the patch must be 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.

3.1.5 Applying Multiple Patches Using a Patch Plan

If you download more than one patch, you have the option to apply all of the patches during one Patch Manager session by creating a patch plan. You can also validate these patches in one session. Depending on the contents of the patches, Patch Manager applies the patches either one at a time or it divides the patches into groups, to optimize server management. Applying groups of patches in a single execution minimizes down time because any impacted Managed Servers are stopped and restarted only once.

A set of patches can be applied in a single execution, rather than individually, only if servers can be shutdown at the beginning of the apply session and none of the included patches require the servers to be available. Therefore, true multiple patch application in online mode occurs if the patches involved contain only the following types of artifacts:

- Oracle Business Process Management (Oracle BPM) templates
- Common Resource artifacts
- Database artifacts other than flexfields
- Diagnostic Testing Framework (DTF) JAR files
- Java EE artifacts
- Oracle Data Integrator (ODI) artifacts
- SOAEXTENSION artifacts

If any of the patches in the patch plan contain additional artifact types, then Patch Manager applies each patch in the plan sequentially, one patch at a time, because certain servers must be running to deploy certain artifacts. In this case, the server shutdown and restart occurs multiple times, as required by each patch.

For more information about patch downloads, see the online help in the **Patches & Updates** tab in My Oracle Support.

For more information about applying multiple patches, see Section 3.7, "End-to-End Process for Applying Multiple Patches".

3.2 Running Oracle Fusion Applications Patch Manager

You run Patch Manager by using the command line utility, `fapmgr`, located in the *FA_ORACLE_HOME*`/lcm/ad/bin` directory (*FA_ORACLE_HOME*`\lcm\ad\bin` for Windows). Its shell script sets the environment and calls the utility. For UNIX, the shell script is `fapmgr.sh` and for Windows, it is `fapmgr.cmd`. You can run `fapmgr` with various commands and options. Only one patching session can be running at any given time. All patch administrators log in as the same operating system user to apply patches. This user must be an owner of the Oracle Fusion Applications Oracle home.

The following command shows the basic syntax for the `fapmgr` utility:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh command [-options]
(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd command [-options]
```

In the preceding example, the following variables are used:

- command: The `fapmgr` utility manages patching-related activities by using one of the commands as described in Table 3–1.
- options: The `fapmgr` commands accept options by using command-line arguments, as described in the specific sections for each `fapmgr` command.

Table 3–1 Oracle Fusion Applications Patch Manager Commands

Command	Description and Link to Documentation
`validate`	Reads the actions in the patch metadata to determine whether a patch is compatible with your environment and can be applied. See Section 3.3, "Validating Patches".
`apply`	Applies a patch. See Section 3.4, "Applying Patches".
`report`	Provides options for generating reports related to patching. See Section 3.5, "Running Patching Reports".
`retry`	Provides the ability to retry post-apply tasks that failed. See Section 11.3.4, "Retrying Failed Post-Patching Tasks in a Previous Session".
`bootstrap`	Updates the Oracle Fusion Applications Patch Manager data model. See Section 4.16, "Patching Oracle Fusion Applications Patch Manager Artifacts".
`abort`	Abandons the previous patching session that failed. See Section 11.3.2, "Abandoning a Failed Patching Session".
`forcefail`	Forces a previously hung session to fail. See Section 11.3.3, "Recovering from an Interrupted Patching Session".

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

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh command -help
(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd command -help
```

To display basic help for the `fapmgr` command, enter `fapmgr` with no options.

> **Note:** In command syntax examples, the brackets ([]) indicate that the value inside the brackets is optional.

3.3 Validating 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 both offline and online modes.

Patch Manager automatically performs patch validation when you run the `fapmgr apply` command. The steps for validating a patch are provided here because Oracle recommends that you validate every patch before applying it, especially those patches that contain updates to SOA composites. You can reduce downtime and potential failures during patching by validating SOA composite patches because you can resolve validation issues before you apply the patch. For more information about resolving these issues, see Section 11.4.2, "Troubleshooting SOA Composite Validation Failures".

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

Syntax

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][-online] [-taxonomyurl
hostname:portnumber]
[-logfile log_file_name][-loglevel log level]

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

Options

Table 3–2 lists the options available for the `validate` command.

Table 3–2 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 My Oracle Support patch plan XML file	Yes, when applying patches in a patch plan.
`online`	Validates patch in online mode so the status of impacted servers is checked.	No, default value is not online.
`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 you want override the value present in the environment properties file and when using the online option.
`logfile`	Overrides the default log file name and sends the processing information to the file you specify, under the *`APPLICATIONS_CONFIG/lcm/logs/11.1.7.0.0/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/11.1.7.0.0/FAPMGR`* using this naming convention: `FAPatchManager_validate_`*`timestamp`*`.log`.
`loglevel`	Records messages in the log file at the level you specify. See Section 11.1, "Oracle Fusion Applications Patch Manager Logging".	No, default value is `INFO`.
`help`	Displays help.	No.

3.4 Applying Patches

You apply patches to an Oracle Fusion Applications environment by running the `fapmgr apply` command. Only one patching session can be active at a time.

Syntax

Use the following syntax for the `apply` command:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh apply -patchtop (or grouptop) path_to_
unzipped_patch [-patchingplan path_to_XML_file]
[-stoponerror] [-online][-taxonomyurl hostname:portnumber]
[-workers number_of_database_workers]
[-logfile log_file_name] [-loglevel level]

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd apply -patchtop (or grouptop path_
to_unzipped_patch [-patchingplan path_to_XML_file]
[-stoponerror] [-online] [-taxonomyurl hostname:portnumber]
[-workers number_of_database_workers]
[-logfile log_file_name] [-loglevel level]
```

Example

The following example applies a patch using 10 database workers in online mode:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh apply -patchtop path_to_unzipped_patch
[-stoponerror] -online -workers 10

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd apply -patchtop
path_to_unzipped_ patch [-stoponerror] -online -workers 10
```

Options

Table 3–3 lists the options available for the `apply` command.

Table 3–3 apply Command Options

Option	Description	Mandatory
`patchtop`	Identifies the directory where the patch is unzipped.	Yes, when applying an individual patch.
`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 My Oracle Support patch plan XML file	Yes, when applying patches in a patch plan.
`stoponerror`	Patching session fails after a post-apply error is reported. You can then manually resolve the issue and start the session again, using the same command.	Yes, when patching in online mode.
`workers`	Identifies the number of workers to use when running database tasks. If you provide a value for the number of workers that is outside the calculated range, you are prompted to provide a value that is within the optimal range. If you do not use the `workers` option, a calculated optimal value is used.	No, default value is calculated if patch contains database tasks. See "Worker Calculation" in Section 3.1.2, "Patching Database Artifacts".
`online`	Applies patch in online mode so that impacted servers are shut down and started and supported artifacts are deployed. See Section 3.1.4, "Online and Offline Patching".	No, default value is not online. Use this option only when applying online patches.

Table 3–3 (Cont.) apply Command Options

Option	Description	Mandatory
`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 you want to override the value present in the environment properties file and when using the online option.
`logfile`	Overrides the default log file name and sends the processing information to the file you specify, under the *`APPLICATIONS_CONFIG/lcm/logs/11.1.7.0.0/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/11.1.7.0.0/FAPMGR`* using this naming convention: `FAPatchManager_apply_`*`timestamp`*`.log`.
`loglevel`	Records messages in the log file at the level you specify. See Section 11.1, "Oracle Fusion Applications Patch Manager Logging".	No, default value is `INFO`.
`help`	Displays help.	No.

3.5 Running Patching Reports

You can generate Patch Manager reports by running the `fapmgr report` command. You can view patch-related information from different perspectives to plan your patching strategy. These reports provide information that can be useful both before and after you apply patches.

Table 3–4 describes the patching reports that can be generated by Patch Manager.

Table 3–4 Patching Reports

Report Name	Report Option	Description	Variations
Patch Impact Report	`-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	`-listcomps`	Displays a list of installed components (product families) and their versions.	You can specify a list of product families or you can see all product families.
Patches Applied Report	`-listpatches`	Displays information about patches and bug fixes that have been applied to your system.	You can specify a list of product families or you can see all product families.
Patch Status Report	`-isapplied`	Tells you whether specific patches or bug fixes were applied to your system.	None.
Diagnostics Report	`-patchprogress`	Displays the progress of a patching session that is currently running.	This report runs automatically after each patching session. You can also run it during a patching session.

The `fapmgr report` command requires an option to specify which report you want 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]

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd report -report_option
-mandatory parameters [optional parameters]
```

3.5.1 Patch Impact Report

The Patch Impact report compares the contents of the patch to be applied with the files that currently exist on your system. You get a complete picture of what file system changes will occur when you apply the patch. You can plan your system downtime by viewing the servers that will be impacted by the patch, along with any manual deployment actions that are required after the patch applies. This report reads the patch metadata, local patch inventory, and the current view snapshot. For more information, see Section 10.2, "Maintaining Snapshot Information". Note that you cannot run the Patch Impact report when you are applying multiple patches that were download in a patch plan.

The Patch Impact report displays the impact information about a patch in five sections: bug fixes, prerequisite bug fixes, impacted product families, servers impacted, and files included in the patch.

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 impacted 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

3.5.1.1 Running the Patch Impact Report

Before you run the Patch Impact report, ensure that the snapshot is current for the environment. For more information, see Section 10.2, "Maintaining Snapshot Information". Note that you cannot run the Patch Impact report when you are applying multiple patches that were download in a patch plan.

Use the following syntax to run the Patch Impact report:

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

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd report -patchimpact -patchtop
path_to_unzipped_patch [optional parameters]
```

The following table describes the 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 file you specify after this parameter. You cannot use an existing file name. If you do not use this parameter, no output file is created.

Parameter	Mandatory	Description
logfile	No	Overrides the default log file name and sends the processing information to the file you specify, under the *APPLICATIONS_CONFIG/lcm/logs/11.1.7.0.0/FAPMGR* directory. If you enter an existing file name, the output is appended to the file. If you do not use this parameter, the utility generates a log file under *APPLICATIONS_CONFIG/lcm/logs/11.1.7.0.0/FAPMGR* using this naming convention: FAPatchManager_report-patchimpact_*timestamp*.log
loglevel	No	Records messages in the log file at the level you specify. See Section 11.1, "Oracle Fusion Applications Patch Manager Logging".
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.

3.5.2 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. You can run the report for all product families or you can select specific product families. 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

3.5.2.1 Running the Product Families Report

Before you run the Product Families report, ensure that the snapshot is current for the environment. For more information, see Section 10.2, "Maintaining Snapshot Information".

Use the following syntax to run the Product Families report:

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

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd report -listcomps [optional
parameters]
```

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

Parameter	Mandatory	Description
comps	No	Supply a comma-separated list of product families (components) you want to see 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 you specify after this parameter. You cannot use an existing file name. If you do not use this parameter, no output file is created.

Parameter	Mandatory	Description
logfile	No	Overrides the default log file name and sends the processing information to the file you specify, under the *APPLICATIONS_CONFIG*/lcm/logs/11.1.7.0.0/FAPMGR directory. If you enter an existing file name, the output is appended to the file. If you do not use this parameter, the utility generates a log file under *APPLICATIONS_CONFIG*/lcm/logs/11.1.7.0.0/FAPMGR using this naming convention: FAPatchManager_report-listcomps_*timestamp*.log
loglevel	No	Records messages in the log file at the level you specify. See Section 11.1, "Oracle Fusion Applications Patch Manager Logging".
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.

3.5.2.2 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
(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd 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

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd report -listcomps -comps
oracle.fusionapps.crm, oracle.fusionapps.fin
-outputfile listproducts.txt -logfile \log\listproducts.log
```

3.5.3 Patches Applied Report

The Patches Applied report provides information about patches that have been applied to an environment. You can 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

3.5.3.1 Running the Patches Applied Report

Before you run the Patches Applied report, ensure that the snapshot is current for the environment. For more information, see Section 10.2, "Maintaining Snapshot Information".

Use the following syntax to run the Patches Applied report:

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

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd report -listpatches [optional
parameters]
```

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

Parameter	Mandatory	Description
comps	No	Supply a comma-separated list of product families (components) you want to see 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 you specify after this parameter. You cannot use an existing file name. If you do not use this parameter, no output file is created.
logfile	No	Overrides the default log file name and sends the processing information to the file you specify, under the *APPLICATIONS_CONFIG/lcm/logs/11.1.7.0.0/FAPMGR* directory. If you enter an existing file name, the output is appended to the file. If you do not use this parameter, the utility generates a log file under *APPLICATIONS_CONFIG*/lcm/logs/11.1.7.0.0/FAPMGR, using this naming convention: FAPatchManager_report-listpatches_*timestamp*.log
loglevel	No	Records messages in the log file at the level you specify. See Section 11.1, "Oracle Fusion Applications Patch Manager Logging".
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.

3.5.3.2 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

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd 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
(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd report -listpatches -comps
oracle.fusionapps.fin,oracle.fusionapps.crm
```

3.5.4 Patch Status Report

The Patch Status report tells you if specific patches or bug fixes have been applied to an environment. You provide a list of patch numbers or bug fixes when you run the report and the output indicates whether each patch or bug fix has been applied. This

report queries the local patch inventory and current view snapshot. For more information, see Section 10.2, "Maintaining Snapshot Information".

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.

3.5.4.1 Running 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]

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd 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.

Parameter	Mandatory	Description
`bug`	Yes, unless the `patch` parameter is used.	Supply a comma-separated list of bug fixes. If you want to 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. If you want to request language patches, append the language code to the patch number, for example, `123456:KO`.
`outputfile`	No	Sends the report output to the file you specify after this parameter. You cannot use an existing file name. If you do not use this parameter, no output file is created.
`logfile`	No	Overrides the default log file name and sends the processing information to the file you specify, under the *APPLICATIONS_CONFIG*`/lcm/logs/11.1.7.0.0/FAPMGR` directory. If you enter an existing file name, the output is appended to the file. If you do not use this parameter, the utility generates a log file under *APPLICATIONS_CONFIG*`/lcm/logs/11.1.7.0.0/FAPMGR` using this naming convention: `FAPatchManager_report-listpatches_`*timestamp*`.log`
`loglevel`	No	Records messages in the log file at the level you specify. See Section 11.1, "Oracle Fusion Applications Patch Manager Logging".
`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`.

3.5.5 Diagnostics Report

After each patching sessions ends, the Diagnostics report is automatically generated so that you can view the results of the session. You can also use this report to monitor a patching session that is currently running, by generating the report from the command line. 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/11.1.7.0.0/FAPMGR/directory. If the Diagnostics report is generated by RUP Installer, it is located in the *APPLICATIONS_CONFIG*/lcm/logs/11.1.7.0.0/FAPMGR directory. The file name is FAPMgrDiagnosticsSummary_*mode*_*timestamp*.html. If you are using this report to monitor a session, you can refresh the report output as the patch progresses.

The Diagnostics report contains three sections.

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 Section 11.2.1, "Log Summary".

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 detailed information about each task that produced a warning or failed. The following information is displayed for each task:

- 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 detailed information about each task executed by Patch Manager. The following information is displayed for each task:

- 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

The Tasks to be Completed section displays a summary of the tasks you must complete after the patch applies. 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.

3.5.5.1 Running the Diagnostics Report

Use the following syntax to run the Diagnostics report while a patch session is active:

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

```
(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd report -patchprogress [optional
parameters]
```

The following table describes the parameter used by the Diagnostics report.

Parameter	Mandatory	Description
`reportlocation`	No	Supply with the full directory path and name of the report output. The default value is *APPLICATIONS_CONFIG*`/lcm/logs/11.1.7.0.0/FAPMGR/FAPMgrDiagnosticsSummarydate:`*fapmgr_commandtimestamp*`.html`.

3.6 End-to-End Process for Applying Individual Patches

The end-to-end process of obtaining and applying individual patches using Patch Manager, includes the following steps. This process assumes that you apply patches in online mode. If you are applying patches in a patch plan, see Section 3.7, "End-to-End Process for Applying Multiple Patches".

> **Note:** As part of the patching process, customers have their own backup and recovery management process. Oracle recommends that you always have a current backup before applying a patch.

Step 1 Research Issue That Must be Resolved by a Patch

When you have an issue that can be resolved by a patch for Oracle Fusion Applications, research the issue on My Oracle Support:

`https://support.oracle.com`

Step 2 Check for Existing Patches

After you find patches that may resolve your issues, you should confirm whether you previously applied them to your system. Select one of these options for finding whether patches were applied:

- Run the Patch Status report to see if specific patches were applied, as described in Section 3.5.4, "Patch Status Report".
- Run the Patches Applied report to see all patches applied for a specific product family, as described in Section 3.5.3, "Patches Applied Report".

Step 3 Obtain and Unzip the Patches

Upon determining that you need new patches, download the patches from My Oracle Support. Unzip the patch ZIP files in your `PATCH_TOP` directory.

Step 4 Read the README File

Read the README file that accompanies each patch. This file contains important information and instructions that must be followed. If a patch contains preinstallation or postinstallation manual steps, they are described in the patch README file. If there

are patches listed under "Other Patches" in the README file, you must download and apply them before you apply the Oracle Fusion Applications patch.

Step 5 Run the Patch Impact Report

Run the Patch Impact report to see the artifacts and Managed Servers impacted by this patch. For more information, see Section 3.5.1, "Patch Impact Report".

An example of the `report` command follows:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh report -patchimpact -patchtop path_to_
unzipped_patch

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd report -patchimpact -patchtop path_
to_unzipped_patch
```

Step 6 Validate the Patch

Run the `fapmgr validate` command to alert you to any potential issues that could result in failure while applying the patch. You should 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. Note that if you validate a patch that contains BI Publisher artifacts, the BI OPMN control process, which is similar to a node manager, has to be up for online mode validation to succeed.

When you apply a patch, the patch validation runs again by default. For more information, see Section 3.3, "Validating Patches".

An example of the `validate` command follows:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh validate -patchtop path_to_unzipped_
patch -online
(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd validate -patchtop path_to_
unzipped_patch -online
```

Step 7 Prepare the System

To prevent locks on patched objects and other data issues during patching of database artifacts, review and perform the following checklist before patching the target environment:

1. Set the environment variables in preparation for running the Health Checker utility, as described in "How to Run Health Checker" in the *Oracle Fusion Applications Upgrade Guide*.
2. Run Health Checker to perform the Patching Readiness Health Checks. For more information, see "Patching Readiness Health Checks" in the *Oracle Fusion Applications Upgrade Guide*.

   ```
   (UNIX) FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_
   HOME/lcm/hc/config/PatchingReadinessHealthChecks.xml -DlogLevel=FINEST

   (Windows)FA_ORACLE_HOME\lcm\hc\bin\hcplug.sh -manifest FA_ORACLE_
   HOME\lcm\hc\config\PatchingReadinessHealthChecks.xml -DlogLevel=FINEST
   ```

3. Run Health Checker to perform the General System Health Checks. For more information, see "General System Health Checks" in the *Oracle Fusion Applications Upgrade Guide.*

   ```
   (UNIX) FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_
   HOME/lcm/hc/config/GeneralSystemHealthChecks.xml -DlogLevel=FINEST
   ```

```
(Windows)FA_ORACLE_HOME\lcm\hc\bin\hcplug.sh -manifest FA_ORACLE_
HOME\lcm\hc\config\GeneralSystemHealthChecks.xml -DlogLevel=FINEST
```

4. Manually shut down the Oracle Enterprise Scheduler Service (ESS) servers, especially when a patch contains a PL/SQL package, by performing the following steps:
   a. Stop the ESS request processor and dispatcher to prevent new requests from being processed. See "Starting and Stopping Oracle Enterprise Scheduler Service Components" in the *Oracle Fusion Applications Administrator's Guide* for more information.
   b. Cancel any in-progress requests. See "Cancelling Oracle Enterprise Scheduler Job Requests" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Enterprise Scheduler* for more information.
   c. Shutdown the ESS WebLogic Server Managed server. See "Starting and Stopping a Product Family Oracle WebLogic Server Domain" in the *Oracle Fusion Applications Administrator's Guide* for more information.
5. If the patch contains BI Publisher reports, ensure that you have your own versions of any customized 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. For more information, see "Before You Begin Customizing Reports" in the *Oracle Fusion Applications Extensibility Guide for Business Analysts*.
6. If the patch contains BI Publisher artifacts, the BI OPMN control process, which is similar to a node manager, has to be up for online mode validation to succeed.
7. This step is for the Windows operating system only. Ensure that there are no active files. If the lock is not released `fapmgr` will not be able to copy files.

Step 8 Apply the Patch

Apply the patch using the `fapmgr apply` command as described in Section 3.4, "Applying Patches". An example of the `apply` command follows:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh apply -patchtop path_to_unzipped patch
-online [-workers number_of_database_workers]

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd apply -patchtop
 path_to_unzipped patch -online
 [-workers number_of_database_workers]
```

Step 9 Monitor and Verify the Application of the Patch

Monitor the progress of the patching session and verify its successful completion by reviewing the Log Summary from a browser. For more information, see Section 11.2, "Monitoring Patching Sessions".

Step 10 Verify the Deployment of Middleware Artifacts

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. For more information, see Section 3.5.5, "Diagnostics Report".

For additional information about the deployment of Oracle Fusion Middleware artifacts, see Section 4.1, "Oracle Fusion Applications Patch Manager Middleware Artifact Support".

Step 11 Run Health Checker for Post Patching Health Checks

Run Health Checker to perform the Post Patching Health Checks. For more information, see "Post Patching Health Checks" in the *Oracle Fusion Applications Upgrade Guide*.

```
(UNIX) FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_
HOME/lcm/hc/config/PostPatchingHealthChecks.xml -DlogLevel=FINEST

(Windows)FA_ORACLE_HOME\lcm\hc\bin\hcplug.cmd -manifest FA_ORACLE_
HOME\lcm\hc\config\PostPatchingHealthChecks.xml -DlogLevel=FINEST
```

Step 12 Run Health Checker for General System Health Checks

Run Health Checker to perform the General System Health Checks. For more information, see "General System Health Checks" in the *Oracle Fusion Applications Upgrade Guide.*

```
(UNIX) FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_
HOME/lcm/hc/config/GeneralSystemHealthChecks.xml -DlogLevel=FINEST

(Windows)FA_ORACLE_HOME\lcm\hc\bin\hcplug.sh -manifest FA_ORACLE_
HOME\lcm\hc\config\GeneralSystemHealthChecks.xml -DlogLevel=FINEST
```

3.7 End-to-End Process for Applying Multiple Patches

The end-to-end process of obtaining and applying multiple patches using Patch Manager, assumes that you create and download patches from My Oracle Support, create a patch plan using the patch plan utility, and you apply these patches in online mode. A set of patches can be applied in one execution, rather than individually, only if servers can be shutdown at the beginning of the apply session and none of the included patches require the servers to be available. See Section 3.1.5, "Applying Multiple Patches Using a Patch Plan" for a list of artifact types that allow multiple patches to be applied in one execution. If one patch fails to apply, the entire process stops.

If any of the patches in the patch plan contain any other artifact types, then Patch Manager applies each patch in the plan sequentially, one patch at a time. In this case, the server shutdown and restart occurs multiple times, as required by each patch.

This process of applying multiple patches in a patch plan includes the following steps.

> **Note:** As part of the patching process, customers have their own backup and recovery management process. Oracle recommends that you always have a current backup before applying a patch.

Step 1 Research Issue That Must be Resolved by a Patch

When you have an issue that can be resolved by one or more patches for Oracle Fusion Applications, research the issue on My Oracle Support:

```
https://support.oracle.com
```

Step 2 Check for Existing Patches

After you find patches that resolve your issues, you should confirm whether you previously applied them to your system. Select one of these options for finding whether patches were applied:

- Run the Patch Status report to see if specific patches were applied, as described in Section 3.5.4, "Patch Status Report".
- Run the Patches Applied report to see all patches applied for a specific product family, as described in Section 3.5.3, "Patches Applied Report".

Step 3 Obtain and Unzip the Patches

Upon determining that you need new patches, create a patch plan and download the patch plan from My Oracle Support. Unzip the patch ZIP files in your `PATCH_TOP` directory.

Step 4 Read the README File

Read the README file that accompanies each patch. This file contains important information and instructions that must be followed. If a patch contains preinstallation or postinstallation manual steps, they are described in the patch README file. If there are patches listed under "Other Patches" in the README file, you must download and apply them before you apply the Oracle Fusion Applications patch.

Step 5 Create a Patch Plan

Create a patch plan by running the Perl script, `adCreateMosPlan.pl` for Oracle Fusion Applications patches. The `adCreateMosPlan.pl` script reads the patch metadata from Oracle Fusion Applications patches to generate the patch plan file, `mosdownload.xml` and is located in *`APPLICATIONS_BASE`*`/fusionapps/applications/lcm/ad/bin`

To run this script, use the Perl executable from *`APPLICATIONS_BASE`*`/dbclient/perl/bin` for UNIX platforms and *`APPLICATIONS_BASE`*`\dbclient\perl\5.8.3\bin\MSWin32-x64-multi-thread` for Windows.

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

```
(UNIX)
setenv PATH /u01/APPLTOP/dbclient/perl/bin:$PATH
setenv PERL5LIB 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

$APPLICATIONS_BASE/dbclient/perl/bin/perl
APPLICATIONS_BASE/fusionapps/applications/lcm/ad/bin/adCreateMosPlan.pl download_
location_for_Oracle_Fusion_Applications_patches_only

(Windows)
set PATH /u01/APPLTOP/dbclient/perl/bin;PATH
SET PERL5LIB=APPLICATIONS_BASE\dbclient\perl\5.8.3;APPLICATIONS_
BASE\dbclient\perl\site\5.8.3\;APPLICATIONS_BASE\dbclient\perl\site

%APPLICATIONS_BASE%\dbclient\perl\5.8.3\bin\MSWin32-x64-multi-thread\perl
%APPLICATIONS_BASE%\fusionapps\applications\lcm\ad\bin\adCreateMosPlan.pl
download_location_for_Oracle_Fusion_Applications_patches_only
```

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>
```

Step 6 Validate the Patch

Run the `fapmgr validate` command to alert you to any potential issues that could result in failure while applying the patches. You should 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. Note that if you validate a patch that contains BI Publisher artifacts, the BI OPMN control process, which is similar to a node manager, has to be up for online mode validation to succeed.

When you apply patches, the patch validation runs again by default. For more information, see Section 3.3, "Validating Patches".

An example of the `validate` command follows:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/fapmgr.sh validate -grouptop
path_to_unzipped_patches
-patchingplan path_to_patch_plan_xml_file -online

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd validate -grouptop
path_to_unzipped_patches
-patchingplan path_to_patch_plan_xml_file -online
```

The patch plan XML file is downloaded from My Oracle Support when you download the patch plan.

Step 7 Prepare the System

To prevent locks on patched objects and other data issues during patching of database artifacts, review and perform the following checklist before patching the target environment:

1. Set the environment variables in preparation for running the Health Checker utility, as described in "How to Run Health Checker" in the *Oracle Fusion Applications Upgrade Guide*.
2. Run Health Checker to perform the Patching Readiness Health Checks. For more information, see "Patching Readiness Health Checks" in the *Oracle Fusion Applications Upgrade Guide*.

   ```
   (UNIX) FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_
   ```

```
HOME/lcm/hc/config/PatchingReadinessHealthChecks.xml -DlogLevel=FINEST

(Windows)FA_ORACLE_HOME\lcm\hc\bin\hcplug.cmd -manifest FA_ORACLE_
HOME\lcm\hc\config\PatchingReadinessHealthChecks.xml -DlogLevel=FINEST
```

3. Run Health Checker to perform the General System Health Checks. For more information, see "General System Health Checks" in the *Oracle Fusion Applications Upgrade Guide.*

```
(UNIX) FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_
HOME/lcm/hc/config/GeneralSystemHealthChecks.xml -DlogLevel=FINEST

(Windows)FA_ORACLE_HOME\lcm\hc\bin\hcplug.sh -manifest FA_ORACLE_
HOME\lcm\hc\config\GeneralSystemHealthChecks.xml -DlogLevel=FINEST
```

4. Manually shut down the Oracle Enterprise Scheduler Service (ESS) servers, especially when a patch contains a PL/SQL package, by performing the following steps:
   a. Stop the ESS request processor and dispatcher to prevent new requests from being processed. See "Starting and Stopping Oracle Enterprise Scheduler Service Components" in the *Oracle Fusion Applications Administrator's Guide* for more information.
   b. Cancel any in-progress requests. See "Cancelling Oracle Enterprise Scheduler Job Requests" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Enterprise Scheduler* for more information.
   c. Shutdown the ESS WebLogic Server Managed server. See "Starting and Stopping a Product Family Oracle WebLogic Server Domain" in the *Oracle Fusion Applications Administrator's Guide* for more information.
5. If the patch contains BI Publisher reports, ensure that you have your own versions of any customized 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. For more information, see "Before You Begin Customizing Reports" in the *Oracle Fusion Applications Extensibility Guide for Business Analysts*.
6. If the patch contains BI Publisher artifacts, the BI OPMN control process, which is similar to a node manager, has to be up for online mode validation to succeed.
7. This step is for the Windows operating system only. Ensure that there are no active files. If the lock is not released `fapmgr` will not be able to copy files.

Step 8 Apply the Patches

Apply the patch using the `fapmgr apply` command as described in Section 3.4, "Applying Patches". An example of the `apply` command follows:

```
(UNIX) 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]

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd apply -grouptop path_to_unzipped_
patches
-patchingplan path_to_patch_plan_xml_file -online -stoponerror
[-workers number_of_database_workers]
```

Step 9 Monitor and Verify the Application of the Patch

Monitor the progress of the patching session and verify its successful completion by reviewing the Log Summary from a browser. For more information, see Section 11.1.2, "Log Files for Multi-apply Patch Manager Sessions".

Step 10 Verify the Deployment of Middleware Artifacts

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. For more information, see Section 3.5.5, "Diagnostics Report".

For additional information about the deployment of Oracle Fusion Middleware artifacts, see Section 4.1, "Oracle Fusion Applications Patch Manager Middleware Artifact Support".

Step 11 Run Health Checker for Post Patching Health Checks

Run Health Checker to perform the Post Patching Health Checks. For more information, see "Post Patching Health Checks" in the *Oracle Fusion Applications Upgrade Guide*.

```
(UNIX) FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_
HOME/lcm/hc/config/PostPatchingHealthChecks.xml -DlogLevel=FINEST

(Windows)FA_ORACLE_HOME\lcm\hc\bin\hcplug.cmd -manifest FA_ORACLE_
HOME\lcm\hc\config\PostPatchingHealthChecks.xml -DlogLevel=FINEST
```

Step 12 Run Health Checker for General System Health Checks

Run Health Checker to perform the General System Health Checks. For more information, see "General System Health Checks" in the *Oracle Fusion Applications Upgrade Guide*.

```
(UNIX) FA_ORACLE_HOME/lcm/hc/bin/hcplug.sh -manifest FA_ORACLE_
HOME/lcm/hc/config/GeneralSystemHealthChecks.xml -DlogLevel=FINEST

(Windows)FA_ORACLE_HOME\lcm\hc\bin\hcplug.sh -manifest FA_ORACLE_
HOME\lcm\hc\config\GeneralSystemHealthChecks.xml -DlogLevel=FINEST
```

4

Patching Oracle Fusion Applications Artifacts

This chapter describes how Oracle Fusion Applications Patch Manager (Patch Manager) supports middleware and database artifacts. It also provides detailed steps for the manual deployment of artifacts. For more information about applying patches, see Section 3.6, "End-to-End Process for Applying Individual Patches" or Section 3.7, "End-to-End Process for Applying Multiple Patches".

This chapter contains the following topics:

- Patching Service-Oriented Architecture (SOA) Composites
- Patching SOAEXTENSION Artifacts
- Patching SOA Resource Bundles
- Patching Sales Prediction Engine (SPE) Inline Service Artifacts
- Patching Tree Artifacts

4.1 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 you apply the patch in offline mode, while others are always required. Before applying any patch, Oracle recommends that you 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 Section 3.5.1, "Patch Impact Report".

Table 4–1 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 your 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 you must perform when the patch includes the specified artifact. These actions are described in more detail in this chapter.

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

  This column describes the actions you must perform when the patch includes the specified artifact and you apply the patch in offline mode. If you apply the patch 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 you perform manual actions.

After you apply 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 Section 3.5.5, "Diagnostics Report". For more detailed information about manual actions for each artifact, refer to the relevant sections in this chapter.

Table 4–1 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 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 you want to implement the change	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
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

Table 4–1 (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
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 Section 4.12.1, "Prerequisites for the Deployment of IPM Artifacts"
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
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

4.2 Oracle Fusion Applications Patch Manager Database Artifact Support

Table 4–2 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.

Table 4–2 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 your 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.

4.3 Patching Oracle B2B Metadata

Oracle recommends that you 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.

4.3.1 Manually Deploying Trading Partner Agreements

If you choose to apply a patch containing Oracle B2B metadata updates in offline mode, and you want to implement the change, you must manually redeploy all Trading Partner Agreements that are affected by the metadata change. If you do not perform the redeployment, the runtime continues to use the older metadata. You can deploy the agreements using the B2B User Interface (UI) or by running the `b2bdeploy` utility from the command line.

4.3.1.1 Deploying Agreements from the User Interface

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

4.3.1.2 Deploying Agreements from the Command Line

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

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

   ```
   cd FA_ORACLE_HOME\soa\bin
   ```

   instead of this command:

   ```
   cd $ORACLE_HOME\bin
   ```

2. Export the entire repository for backup purposes.

   ```
   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, you must add the document definition. For more information, see "Adding Document Definitions" in the *Oracle Fusion Middleware User's Guide for Oracle B2B*.

For more information about the `b2bdeploy` command, see "Deploying Agreements" in the *Oracle Fusion Middleware User's Guide for Oracle B2B*.

4.4 Patching Oracle Business Intelligence Publisher Artifacts

When you patch Oracle Business Intelligence Publisher (BI Publisher) artifacts (Reports and Captions) 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, you must manually deploy the BI Publisher artifacts.

Oracle recommends you do not use offline mode when a patch contains BI Publisher artifacts. If you decide to apply a patch in offline mode, you must manually deploy the changes to the Oracle Business Intelligence repository 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.

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

4.4.1 Prerequisites for Manual BI Publisher Artifact Deployment

The following prerequisites are required before manual deployment of BI Publisher artifacts:

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 steps in Section 4.4.2, "Manually Deploying BI Publisher Artifacts", you must first determine exactly which files were actually copied to the Oracle home during the OPatch apply stage. You can review the Patch Impact report to get this list of files. You can also capture the list of files from the messages sent to the console and to the `FAPatchManager_apply_`*`timestamp`*.log file.

   After you have that list, you must 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, you have partially rolled back a previous patch.

4.4.2 Manually Deploying BI Publisher Artifacts

Follow these steps to manually deploy BI Publisher artifacts:

1. Shut down the BI Presentation server.

2. 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 Section 3.5.1, "Patch Impact Report".

3. Using the example in Step 2, go to the `C:\patch\custom\scripts` directory.

4. Locate the Catalog Manager *diff* files listed in the subdirectories under the directory in Step 3. These files have *.diff* extensions.

5. 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.

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

- *diff_file_location* refers to the file from Step 4.
- *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
  ```

  Example of *oracle-instance* on Windows:

  ```
  C:\APPLICATIONS_BASE\instance\BIInstance\bifoundation
  \OracleBIPresentationServicesComponent\coreapplication_
  obips1\catalogmanager
  ```

b. Apply the Catalog Manager patch file.

```
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.

6. Repeat Step 5 for the run-time catalog.
7. Restart the Oracle Business Intelligence system components using `opmnctl`:

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

4.5 Patching Oracle Business Process Management (Oracle BPM) Templates

Oracle recommends that you 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.

4.5.1 Manually Publishing Oracle BPM Templates

If you choose to apply a patch containing updates to Oracle BPM templates in offline mode, you must manually publish the new Oracle BPM Template to the Oracle MDS repository supporting the Oracle BPM Composer instance after you apply the patch. You 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 that you create. You must provide the location of the `mds-config.xml` configuration file as one of the input parameters of the `publish_template` command.

4.5.1.1 Creating the mds-config.xml Configuration File

Follow these steps to create the `mds-config.xml` configuration file:

1. Copy the `mds-config-template.xml` file from your SOA server installation to a local directory.

   ```
   cp SOA_ORACLE_HOME/bpm/config/mds-config-template.xml /local_
   directory/mds-config.xml
   ```

2. Modify the following properties in the file you just copied to the temporary directory:

   - Set `jdbc.userid` to the database user name of the SOA MDS database
   - Set `jdbc.passwd` to the database password of the SOA MDS database
   - Set `jdbc.url` to the connection URL of the SOA MDS database, for example, `jdbc:oracle:thin:@host2:1525:mds`
   - Set `partition.name` to `obpm`

4.5.1.2 Publishing the New Oracle BPM Template to the MDS Repository

Follow these 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 Section 3.5.5, "Diagnostics Report".
2. Expand the archive that contains the new BPM template, so the `publish_template` command can find the template.
   - Create or use an existing local directory.
   - 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.

   ```
   (UNIX) SOA_ORACLE_HOME/common/bin/wlst.sh
   (Windows) SOA_ORACLE_HOME\common\bin\wlst.cmd
   ```

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] )
   ```

   See "publish_template" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* for more information about the `publish_template` command syntax.

   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.

4.6 Patching 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. For more information, see "Starting and Stopping Oracle Enterprise Scheduler Service Components" in the *Oracle Fusion Applications Administrator's Guide*.

4.7 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. You must stop and restart all Managed Servers in all domains after you apply the patch.

4.8 Patching Diagnostic Testing Framework (DTF) JAR Files

No manual steps are required when patching DTF artifacts in either online or offline mode.

4.9 Patching E-Mail and Web Marketing (EWM) Artifacts

Oracle recommends you 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 Section 4.13, "Patching Java EE Artifacts".

4.10 Patching Flexfield Artifacts

Oracle recommends that you 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 you decide that you do not want to implement the changes to a flexfield, you can revert to a previous version of a flexfield. For more information, see Section 11.3.8, "Revert To a Previous Flexfield Definition After It Is Updated By a Patch".

4.10.1 Manually Deploying Patched Flexfields

Follow these steps when patching flexfields in offline mode to manually deploy the patched flexfields.

1. Ensure that the Administration Server and database are running.
2. Stop and start the `FNDSETUP` Managed Servers. For more information, see "Starting and Stopping" 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. Because you run this on a domain that hosts the `FndSetup` Application, you do not have to specify this application within the parentheses. 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 your Payroll application:

1. From your 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**.

4.10.2 Perform Flexfield NameSpaces Merge

If you apply a patch in offline mode that includes flexfield changes that require a `NameSpaces` merge, you must perform the manual steps in this section.

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.

4.11 Patching 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 you have any customizations on the template that the patched version will replace, you must manually incorporate the customizations in the new version of the template. If you have any settings or configurations that refer to the Group Space template name, ensure that you 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 that you patch Group Space templates in online mode. When updates to Group Space templates are introduced in a patch and you have not customized the template included in the patch, 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

4.11.1 Manually Deploying Group Space Templates

In offline mode, or in the case of failure, you must manually deploy the new Group Space template 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.

```
(UNIX) WC_ORACLE_HOME/common/bin/wlst.sh
(Windows) WC_ORACLE_HOME\common\bin\wlst.cmd
```

3. Deploy the Group Space template.

```
importGroupSpaces('appName', 'fileName')
```

The `appName` is always `webcenter` and the `fileName` is the name of the WebCenter archive file, from the patch, that you want to import. Refer to the Diagnostics report to get the full path and file name. For more information, see Section 3.5.5, "Diagnostics Report".

For more information, see "importGroupSpaces" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*. For more information about Group Space templates, see "Importing Space Templates" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter Portal*.

4.12 Patching Imaging and Process Management (IPM) Artifacts

Oracle recommends you 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, you must manually deploy the IPM artifacts.

4.12.1 Prerequisites for the Deployment of 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 your `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 your Oracle Fusion Payables Application and Expenses Application, all related searches, and inputs to a local file.

4.12.2 Manually Deploying IPM Artifacts

1. Review the Diagnostics report to find the location of the IPM artifacts that were copied to *FA_ORACLE_HOME*. For more information, see Section 3.5.5, "Diagnostics Report".

2. Access the WLST shell.

```
(UNIX) ECM_ORACLE_HOME/common/bin/wlst.sh
(Windows) ECM_ORACLE_HOME\common\bin\wlst.cmd
```

3. Deploy the IPM artifact.

```
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/i
pm/ApInvoiceIpmApp.xml',
action='Update', name='Payables Invoice Application','None')

importIPMInput(exportFile='/net/server01/fusionapps/applications/fin/ap/ipm/ApI
nvoiceIpmApp.xml',
action='Update', name='Payables Invoice Input')
```

4. If applicable, perform your customizations on the new file, based on the file you exported.

4.13 Patching Java EE Artifacts

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

- Administration Server
- Node manager
- Database

In offline mode, you must 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 you are applying, run the Patch Impact report. For more information, see Section 3.5.1.1, "Running the Patch Impact Report". For example, if the Patch Impact report indicates that the patch updates a Java EE artifact in the Financials Domain, then you must stop the Financials Domain, 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.

For more information about stopping and starting servers, see "Starting and Stopping a Product Family Oracle WebLogic Server Domain" in the *Oracle Fusion Applications Administrator's Guide*.

4.14 Patching Oracle Data Integrator (ODI) Artifacts

Oracle recommends that you 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 you patch ODI artifacts in offline mode, you must manually import the changed ODI content to the ODI repository. In both online and offline modes, the ODI repository import tool and the database must be running.

Note: Oracle Fusion Applications Provisioning does not install ODI Studio. You must install ODI Studio before manually importing ODI changes, for example, after you apply patches that deliver ODI changes in offline mode or when you need to manually retry a failed ODI import step in online mode. For more information, see "Installing Oracle Data Integrator" in the *Oracle Fusion Middleware Installation Guide for Oracle Data Integrator*.

4.14.1 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.

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 Section 3.5.5, "Diagnostics Report".
3. Start the ODI Studio.

   ```
   (UNIX) odi.sh
   (Windows) odi.exe
   ```

4. Access the ODI Studio.
   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.

   For more information, see "Connecting to a Work Repository" in the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.
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.
   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 you want to import.

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.

For more information, see "Exporting/Importing" in the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

7. Close the ODI Studio after importing all the files in the order specified in the patch README file.

4.15 Patching Oracle Forms Recognition and Oracle Document Capture Artifacts

Oracle Forms Recognition (OFR) and Oracle Document Capture (ODC) artifacts are used only by Windows. When you apply a patch that contains OFR and ODC artifacts, Patch Manager backs up your customized files, if any, before copying the new files to *FA_ORACLE_HOME*. You must then manually install the OFR and ODC artifacts that were delivered in the patch. If you have not installed OFR and ODC, refer to "Setting Up Oracle Document Capture and Oracle Forms Recognition" in the "Financials" chapter of the *Oracle Fusion Applications Post-Installation Guide* for the installation steps. Then return to this section for patching.

4.15.1 Install OFR Artifacts

If you previously customized the `AP_Packaged_Project_1004a.ini` file, then you should not install this file from the patch as the patch delivers it. Instead you should keep your existing file and copy the file from the patch to a directory with a different name. Then you should compare the files and manually update your existing file with the changes to preserve your customizations.

1. Review the Patch Impact report to find the location of the files related to OFR. For more information, see Section 3.5.1, "Patch Impact Report".

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 you see the existing `AP_Packaged_Projects_1004a.ini` file 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**.

Note: 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) that you previously created.

   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 you 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 your implementation.

f. Verify your changes.

g. Save and close the file.

4.15.2 Update ODC Expenses Metadata

To update ODC Expenses metadata:

1. Install the ODC Import-Export Utility from the Companion DVD, if you have not already done so.
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.

4.16 Patching Oracle Fusion Applications Patch Manager Artifacts

When updates to Oracle Fusion Applications Patch Manager are introduced in a patch, you must apply the patch 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 RUP Installer. The compatible version of Opatch is available on My Oracle Support under patch 14044793.

For more information about OPatch, see "Patching Oracle Fusion Middleware with Oracle OPatch" in the *Oracle Fusion Middleware Patching Guide*.

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 you 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]

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd bootstrap [-logfile log_file_name]
[-loglevel level]
```

4.17 Patching 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.

4.18 Patching 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 when you may need to follow the recovery steps include:

- You apply a security patch that introduces a set of new policies and LDAP GUIDs, after backing up the policy store and the applications database. You have not performed the GUID reconciliation with the applications database due to an unrelated database issue. To resolve the database issue, you 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.
- You 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:

- Patching Applications Policies (system-jazn-data.xml)
- Patching Data Security Grants
- Patching Data Role (RGX) Templates
- Patching Data Security Grants and Data Role (RGX) Templates
- Backing up the Data Security Store
- Recovering Data Security Seed Data from the Backup

Oracle recommends that you 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 Section 3.5.1, "Patch Impact Report".

For more information about general troubleshooting for Patch Manager, see Chapter 11, " Monitoring and Troubleshooting Patches". For more information about security in Oracle Fusion Applications, see "Securing Oracle Fusion Applications" in the *Oracle Fusion Applications Administrator's Guide*.

4.18.1 Patching 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 you (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 you apply the patch. If a change is considered *safe*, you can apply the patch in online mode. If a

change is considered to be a *conflict*, you must follow the steps to apply the patch in offline mode, which includes manually resolving conflicts. Table 4–3 describes a summary of changes that are safe and those that cause a conflict.

Table 4–3 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.
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.

For more information about what the `system-jazn-data.xml` file contains, see the "The OPSS Policy Model" chapter in the *Oracle Fusion Middleware Application Security Guide*. For more information about patching applications policies, see the "Upgrading Oracle Fusion Applications Policies" chapter in the *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*.

This section contains information about the following methods for patching applications policies:

- Prerequisites for Patching Applications Policies in Online Mode
- Patching Applications Policies in Offline Mode using APM

4.18.1.1 Prerequisites for Patching Applications Policies in Online Mode

Oracle recommends that you 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, see Section 3.3, "Validating Patches" and Table 4–3, " Changes to Applications Policies".

   If the validation reports any conflicts, then you can choose to apply all safe changes in online mode. Later, you must apply conflicting changes in offline mode, as described in Section 4.18.1.2, "Patching Applications Policies in Offline Mode using APM".

2. All domains that use the OPSS Policy store in Oracle Internet Directory for authorization policies must be shut down before the patch applies.

4.18.1.2 Patching Applications Policies in Offline Mode using APM

The following steps must be performed if you patch applications policies in offline mode using APM.

> **Note:** 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.

1. Run the Patch Impact report to see which artifacts are included in the patch. For more information, see Section 3.5.1, "Patch Impact Report". Note the location where `system-jazn-data.xml` is located in the patch, because you are prompted for this location 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. For more information, see Section 3.4, "Applying Patches".
3. Log in to Authorization Policy Manager.
4. Open the **Policy Upgrade Management** tab. Follow the steps in "The Policy Upgrade Management Tab" in the *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*. When you select the application to patch from the pull-down **Application** list, you should see choices such as the following:
   - `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. Follow the steps in "Analyzing Patch Differences" and "Resolving Patch Differences" in the *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*. If there are any errors during this step, restore the backup, as described in "Prerequisites to Patching Policies" in the *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*.
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.

4.18.2 Patching Data Security Grants

Oracle recommends that you patch data security grants in online mode. In both online and offline patching mode, you must ensure that the prerequisites are met.

4.18.2.1 Prerequisites for Patching Data Security Grants

1. Run the Patch Impact report to see which artifacts are included in the patch. For more information, see Section 3.5.1, "Patch Impact Report".

2. Back up the data security store by using the Oracle Database data pump export tool. For more information, see Section 4.18.5, "Backing up the Data Security Store".

4.18.2.2 Patching Data Security Grants

Follow these steps to patch data security steps:

1. Run Patch Manager to apply the seed data changes to the data security system. For more information, see Section 3.4, "Applying Patches". 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, you must manually run the `DSDataMigrator` utility as described in "Reconciling GUIDs" 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 you are unable to resolve the errors that occurred while applying the seed data changes, recover the seed data from the backup export file you created in Step 2 of the prerequisites. For more information, see Section 4.18.6, "Recovering Data Security Seed Data from the Backup".

4.18.3 Patching Data Role (RGX) Templates

Oracle recommends that you 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.

4.18.3.1 Manually Deploying Data Role (RGX) Templates

Follow the steps in this section when you apply a patch in offline mode that contains 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 deployed after you apply a patch that contains changes to data role templates, so that they remain synchronized with each other.

1. Run the Patch Impact report to see which artifacts are included in the patch. For more information, see Section 3.5.1, "Patch Impact Report". 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`*. For more information, see Section 3.4, "Applying Patches".
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')
```

For more information, see "createMetadataLabel" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

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.

   ```
   (UNIX) SOA_ORACLE_HOME/common/bin/wlst.sh
   (Windows) SOA_ORACLE_HOME\common\bin\wlst.cmd
   ```

   b. Connect to WebLogic Server.

   ```
   > 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 Section 3.5.5, "Diagnostics Report".

   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')
   ```

   Table 4–4 displays the parameters required by the `importMetadata` command.

Table 4–4 Parameters for the importMetadata WLST Command

Parameter Name	Description
application	Enter the value of `oracle.security.apm`
server	Enter the name of your 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 4–1 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

(Windows)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'

(Windows)
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 4–2 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

(Windows)
\\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/fndDataSecProvide
r/GeneralLedger.xml

(Windows)\\machine1\oracle\apps\oracle\fin\gl\rgx\dataSecPolicy\fndDataSecP
rovider\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/**'

(Windows) importMetadata(application='oracle.security.apm',
server='AdminServer',
fromLocation='\\machine1', docs='/oracle/apps/oracle/fin/gl/**'
```

For more information, see "Importing WebCenter Portal Service Metadata and Data (Framework Applications)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter Portal.*

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')
   ```

   For more information, see "promoteMetadataLabel" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

   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_
point')
```

7. Assuming Steps 3 through 5 are successful, Oracle recommends that you preview the execution of your changed data role templates. Run the preview from the **Summary** tab after you open the data role template from the APM console. For more information, see "Running a Template" in the *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*.

   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. For more information, see "Running a Template" in the *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition).*

9. If all steps are successful, delete the MDS label you 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')
   ```

4.18.4 Patching Data Security Grants and Data Role (RGX) Templates

Oracle recommends that you 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 you apply a patch, so they remain synchronized with each other.

To patch data security grants and data role templates:

1. Run the Patch Impact report to see which artifacts are included in the patch. For more information, see Section 3.5.1, "Patch Impact Report". 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 Section 4.18.5, "Backing up the Data Security Store".
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*. For more information, see Section 3.4, "Applying Patches".

   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, you must manually run the `DSDataMigrator` utility as described in "Reconciling 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. For general troubleshooting information, see Section 11.5, "Troubleshooting Patching Sessions for Database Content".

   If you are unable to resolve the errors that occurred while applying the seed data changes, recover the seed data from the backup export file you created in Step 2. For more information, see Section 4.18.6, "Recovering Data Security Seed Data from the Backup".

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')
   ```

   For more information, see "createMetadataLabel" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

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.

      ```
      (UNIX) SOA_ORACLE_HOME/common/bin/wlst.sh
      (Windows) SOA_ORACLE_HOME\common\bin\wlst.cmd
      ```

   b. Connect to WebLogic Server.

      ```
      > 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. For more information, see Section 3.5.5, "Diagnostics Report".

      ```
      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')
      ```

Table 4–5 displays the parameters required by the `importMetadata` command.

Table 4–5 Parameters for the importMetadata WLST Command

Parameter Name	Description
application	Enter the value of `oracle.security.apm`.
server	Enter the name of your 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 4–3 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

(Windows)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'

(Windows)
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 4–4 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

(Windows)
\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/fndDataSecProvide
r/GeneralLedger.xml

(Windows)\machine1\oracle\apps\oracle\fin\gl\rgx\dataSecPolicy\fndDataSecPr
ovider\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/**'

(Windows) importMetadata(application='oracle.security.apm',
server='AdminServer',
fromLocation='\\machine1', docs='/oracle/apps/oracle/fin/gl/**'
```

For more information, see "Importing WebCenter Portal Service Metadata and Data (Framework Applications)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter Portal*.

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 you created in Step 2. For more information, see Section 4.18.6, "Recovering Data Security Seed Data from the Backup".

   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')
   ```

   For more information, see "promoteMetadataLabel" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

   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_point')
   ```

9. Assuming Steps 2 through 7 are successful, Oracle recommends that you preview the execution of your changed data role templates. Run the preview from the **Summary** tab after you open the data role template from the APM console. For more information, see "Running a Template" in the *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*.

   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. For more information, see "Running a Template" in the *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*.

If the results are not correct, restore the database from the backup created in Step 2. For more information, see Section 4.18.6, "Recovering Data Security Seed Data from the Backup".

11. If all steps are successful, delete the MDS label you 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')
```

4.18.5 Backing 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 your Oracle Fusion Applications instance. You are prompted for the Oracle Fusion Applications database user name and password.

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

(Windows, TWO_TASK is known as LOCAL)
adsetenv.cmd
APPSORA.cmd
echo %LOCAL%
```

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
```

For more information about Oracle Data Pump, see:
http://www.oracle.com/technetwork/database/enterprise-edition/data-pump-overview-084963.html

4.18.6 Recovering 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 you previously created.

```
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
```

4.19 Patching 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:

- You have no SOA composite customizations in Oracle JDeveloper. If you do have customizations, follow the steps in Section 4.19.1, "Preserving SOA Composite JDeveloper Customizations Before Applying a Patch".
- You 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 you do not use offline mode when a patch contains SOA composites. If the patch fails while attempting to deploy a SOA composite, you may have to manually deploy the composite. For more information, see Section 4.19.2, "Manually Deploying SOA Composites".

For information about resolving validation errors, see Section 11.4.2, "Troubleshooting SOA Composite Validation Failures". For information about recovering from deployment errors, see Section 11.4.3, "Troubleshooting SOA Composite Deployment Failures".

If you customized SOA composites used by Oracle Fusion Applications in JDeveloper, you must preserve these customizations before you apply 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 you patch SOA composites

- 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`. You can 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. If the server is not running, see "Diagnosing Problems with Oracle WSM Policy Manager" in the *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

  For information about Fusion Applications Control, see "Accessing Fusion Applications Control" in the *Oracle Fusion Applications Administrator's Guide*.

4.19.1 Preserving SOA Composite JDeveloper Customizations Before Applying a Patch

If you performed JDeveloper customizations, not supported by OPatch, to a SOA composite and then you deploy the composite to the SOA runtime, subsequent patches are not directly deployable. The Patch Manager validation process returns the appropriate error, which instructs you 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.

Before applying the patch, review the recommendations in "Merging Runtime Customizations from a Previously Deployed Revision into a New Revision" in the *Oracle Fusion Applications Extensibility Guide for Developers* to ensure that your customizations will be merged successfully.

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 for you 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 Section 3.3, "Validating Patches".

   Note: You must run Patch Manager validation before applying every patch, especially patches that contain SOA composites. If your 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 you apply 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". For more information, see "Customizing SOA Composite Applications with JDeveloper" in the *Oracle Fusion Applications Extensibility Guide for Developers*.
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 you will need it in Step 8. You can 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 your file name is different, rename it to the original name.

10. Now you should be able to both validate and apply this patch successfully using Patch Manager in online mode.

For more information about customizing SOA composites, see "Customizing and Extending SOA Components" in the *Oracle Fusion Applications Extensibility Guide for Developers*.

4.19.2 Manually Deploying SOA Composites

If a customized SOA composite deployment fails during patching, you must manually deploy this composite using WLST commands. You must also manually deploy SOA composites if you apply a patch in offline mode that contains SOA composites.

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 to find the name and location of the `sca_*.jar` file that was copied to *`FA_ORACLE_HOME`* by Patch Manager. For more information, see Section 3.5.5, "Diagnostics Report".
2. If the previous revision contained JDeveloper customizations, ensure that you deploy the patched revision with the merged JDeveloper customizations. Using the `sca_*.jar` file from Step 1, follow the JDeveloper customization merge instructions that are described in Section 4.19.1, "Preserving SOA Composite JDeveloper Customizations Before Applying a Patch". Then use the merged sca_*.jar for Step 3.
3. Deploy the patched composite using the single patch composite command.

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

4.20 Patching SOAEXTENSION Artifacts

When updates to SOAEXTENSION artifacts are introduced in a patch, you must stop and restart all SOA-INFRA Managed Servers in all domains. Both online and offline patching require this step.

4.21 Patching SOA Resource Bundles

Oracle recommends you 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, you must manually deploy the SOA resource bundle and restart the SOA composites that depend on the SOA resource bundle.

The following must be running when you patch SOA resource bundles:

- Administration Server

- SOA-INFRA Managed Servers
- Node manager
- Database

After you apply the patch, refer to the Diagnostics report to get a complete list of composites that depend on each SOA resource bundle and also the domains. For more information, see Section 3.5.5, "Diagnostics Report".

4.21.1 Manually Deploying SOA Resource Bundle 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
   ```

   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 *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

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`, you must 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. For more information, see "Accessing Fusion Applications Control" in the *Oracle Fusion Applications Administrator's Guide*.

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**.

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 *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

4.22 Patching Sales Prediction Engine (SPE) Inline Service Artifacts

Oracle recommends that you 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.

- Prerequisites Required Before Manual SPE Artifact Deployment
- Manually Deploying SPE Artifacts After You Apply the Patch in Offline Mode

> **Note:** SPE artifacts are provisioned only when Oracle Fusion CRM Performance Management is provisioned. If CRM Performance Management is not provisioned in your environment, you should not deploy SPE artifacts.

4.22.1 Prerequisites Required Before Manual SPE Artifact Deployment

1. You must be running JDK 1.6 or later.
2. You must have 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`

   (Windows)
   `crm\components\crmPerformance\zsp\predictionmodule\inlineservice`

4.22.2 Manually Deploying SPE Artifacts After You Apply the Patch in Offline Mode

Oracle suggests you refer to the text file that was created when provisioning completed, which is a textual overview of your topology, as you follow these steps. For more information, see "Installation Process Flow" in the *Oracle Fusion Applications Installation Guide*.

1. Stop and start `bi_server1` to include the changes in `AdfZspPredictionModuleSupportUtilities.jar`.
2. 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 Section 4.22.1, "Prerequisites Required Before Manual SPE Artifact Deployment".
   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 that you have 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

(Windows) java -jar deploytool.jar -deploy -server server01.oracle.com
 -port 7001 -terminateSessions true D:\SPE\RTD\ILS\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 5
Inline 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"
```

4.23 Patching 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, you must 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 Section 3.5.1, "Patch Impact Report".

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 you need the version numbers to manually flatten the tree version changes.

   For more information about the Diagnostic report, see Section 3.5.5, "Diagnostics Report". For more information about log files, see Section 11.1.1, "Log Files for Single Patch Manager Sessions".

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 your 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 you want to make changes to the tree version, 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.

For more information about trees, see "Define Trees" in the "Maintain Common Reference Objects" chapter in the *Oracle Fusion Applications Common Implementation Guide*.

5

Introduction to Oracle Fusion Applications Languages

This chapter provides an introduction to the tools you use to install and maintain a set of languages in Oracle Fusion Applications.

This chapter includes the following topics:

- Installing and Upgrading Languages
- Maintaining Languages

5.1 Installing and Upgrading Languages

Oracle Provisioning installs only the English language. Upgrade Orchestrator upgrades all installed languages. To add a language, use Language Pack Installer. For more information about Language Pack Installer, see "Installing and Maintaining Oracle Fusion Applications Languages" in the *Oracle Fusion Applications Administrator's Guide.*

5.2 Maintaining Languages

Oracle Fusion Applications content is translated to different languages and fixes are made available as individual patches. If your environment uses multiple languages, whenever you apply a patch with translatable content, you may choose to also apply the associated translated patch for each of your installed languages. If a patch does not contain any translated content, such as a PL/SQL package, only the English patch is available.

If an Oracle Fusion Applications environment contains languages other than English, the recommended method for applying patches is to apply the English patch first and then apply the translation patch for each installed language.For example, after you apply a language pack for another language, such as Korean, whenever you apply a patch that involves translatable content, you must apply the base English patch and also the Korean patch for that fix.

For detailed information about how to apply a patch, see Section 3.4, "Applying Patches".

6

Patching Oracle Identity Management Artifacts

The Oracle Identity Management patching framework provides the tools to support updates to Oracle Identity Management software. This chapter introduces the Oracle Identity Management patching framework and its components.

This chapter contains the following topics:

6.1 Overview of the Oracle Identity Management Patching Framework

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.

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

6.1.1 Products Supported

Oracle Identity Management patching framework 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

6.2 Understanding the 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 your Oracle Identity Management environment, with complete information of 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

6.2.1 Oracle Identity Management Patch Manager

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

6.2.1.1 Patch Plan Generation

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 is generated by the

Oracle Identity Management Provisioning Wizard and 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.

6.2.2 Oracle Identity Management Patcher

The Oracle Identity Management Patcher is a per host patch execution engine. The Oracle Identity Management Patcher uses the patch plan generated by the Oracle Identity Management Patch Manager and executes the patch plan steps. These steps are applicable to the host where the Oracle Identity Management Patcher is running.

The Oracle Identity Management Patcher executes the steps until the next step in the patch plan is to be executed on a different host. If the steps in the patch plan are to be executed on a different host, then the Oracle Identity Management Patcher displays a message and exits. You may need to execute the Oracle Identity Management Patcher on a host multiple times if required, during the execution of a given patch plan, as different phases of the patch plan are executed.

6.2.2.1 Patch Plan Phases

The patch plan consists of the following three phases:

- Patch Apply Prerequisite Phase (All services will be up)

  The prerequisite checks are executed and no changes are made to the deployment. This phase can be executed before you plan your system downtime and apply patches. Any issues can be addressed immediately, which will enable the patches to be applied without any issues during downtime.

- Patch Pre-Apply Phase (All services will be down)

  All servers which must be down to apply patches are stopped. This is deployment-aware; for example, if the patch top consists solely of an Oracle Access Manager patch, you need not stop every server instance. Only Oracle HTTP Server and Oracle Identity Manager, which depend on Oracle Access Manager, and Oracle Access Manager itself will be stopped. Oracle Internet Directory will remain up during plan execution. This ensures that the downtime is minimized.

- Patch Apply Phase (Limited services will be available)

  Patches are applied, any artifact changes related to the patches are executed, and servers are started.

6.2.3 Oracle Identity Management Patching Framework Installation

The Oracle Identity Management patching framework is installed when you provision an Oracle Identity Management environment using the Oracle Identity Management Provisioning Wizard.

For more information about installing the Oracle Identity Management Provisioning Wizard and the Oracle Identity Management patching framework, see the *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity Management (Oracle Fusion Applications Edition)*.

The following sections detail the patch directory structure and property files you should verify to understand the different patch folders and ensure the correct variable values are set.

6.2.4 Directory Structure

When you provision an Oracle Identity Management environment using the Oracle Identity Management Provisioning Wizard, the Oracle Identity Management patching framework directory structure is created as described in Table 6–1.

`$IDM_LCM_TOP`: Root of the Oracle Fusion Middleware home where the Oracle Identity Management Provisioning Wizard and Oracle Identity Management patching framework is installed. `$IDM_LCM_TOP` contains the executables for the Oracle Identity Management provisioning and patching tools.

`$LCM_CONFIG`: Location where the Oracle Identity Management patching framework configuration files are located such as status files, logs and patch plan.

This directory exists under the `/config` directory which also holds the WebLogic IDM Domain, OID and OHS instance files. `/config` may or may not exist under the `$IDM_LCM_TOP`, depending on options selected during the Oracle Identity Management Provisioning Wizard installation.

patch top Directory: Select any directory for organizing the downloaded patches. Create different sub-directories under the selected top-level directory for storing the patches product-wise.

You should download the patches to these sub-directories according to the products they belong to. It is not necessary to unzip the files as both zipped and unzipped formats are supported.

The top-level directory containing these sub-directories is referred to as patch top. Figure 6–1 illustrates the example of a patch top directory structure.

Figure 6–1 Example of a patch top directory structure

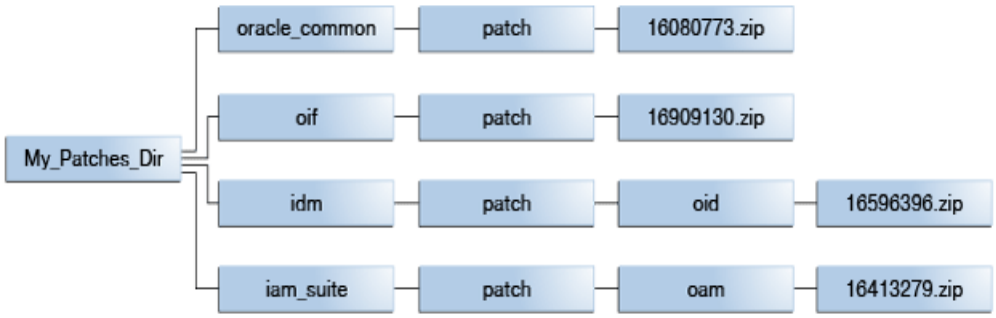


Table 6–1 details the Oracle Identity Management patching framework directory structure.

Table 6–1 Oracle Identity Management patching framework directory structure

Directory Structure	Description
`$IDM_LCM_TOP/patch`	Contains the patching framework executables and configuration files.
`$IDM_LCM_TOP/patch/bin`	Contains the Oracle Identity Management Patch Manager and Oracle Identity Management Patcher tools executable files (`.sh / .bat`).
`$IDM_LCM_TOP/patch/config`	Contains `env.properties`, `patchtop-content.properties`, `idmpatchmgr-logging.properties`, `idmpatch-logging.properties` that can be configured before running the patching tools.
`$IDM_LCM_TOP/patch/script`	Contains scripts, property files required by patching framework to start-stop services as well as for applying artifacts.

Table 6–1 (Cont.) Oracle Identity Management patching framework directory structure

Directory Structure	Description
`$LCMCONFIG/patch`	Contains status files, logs, patch bundles, and patch plan generated by the patching framework when a patch session is started.
`$LCMCONFIG/patch/patches`	Contains the patch bundles created by the Oracle Identity Management Patch Manager from the patch top provided. This patch bundle is used by the Oracle Identity Management Patch Manager tool to generate the patch plan.
`$LCMCONFIG/patch/session`	Contains the patch plan in a machine-readable format, and other session information that the Oracle Identity Management Patcher uses to execute the patching steps.
`$LCMCONFIG/patch/status`	Contains host-based files tracking the execution state of each patch plan step. Also contains the Oracle Identity Management Patch Manager and Oracle Identity Management Patcher log files and the patch plan.
`$LCMCONFIG/topology`	Contains the topology store file `topology.xml`, which provides detailed information regarding the Oracle Identity Management Provisioning environment. This is used by the Oracle Identity Management patching framework for generating the patch plan.

6.2.5 Configuration Files

The Oracle Identity Management patching framework requires correct values to be set in the `patchtop-contents.properties` file and the `env.properties` file.

6.2.6 Verify patchtop-contents.properties

The downloaded patches have to be organized in the following directory structure:

- You should have a top-level patch top directory containing different subdirectories for storing product-wise patches.
- The mapping between the products and the relative paths of the subdirectories under the patch top is stored in `patchtop-contents.properties`.

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

> **Note:** 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-wise patches correctly.

The `patchtop-contents.properties` file contents are detailed in Example 6–1.

Example 6–1 patchtop-content.properties file contents

```
#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
oim=iamsuite/patch/oim
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.

6.2.7 Verify env.properties

The `env.properties` file located at: `$IDM_LCM_TOP/patch/config/env.properties` contains all the environment variables required by the Oracle Identity Management patching framework as detailed in Table 6–2. 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 detailed in Table 6–2 are set.

Table 6–2 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.

Table 6–2 (Cont.) Environment Variables

Name	Value	Mandatory	Description
ANT_HOME	Ant Home	No	Absolute path pointing to the root directory of an Apache Ant distribution. It is only required to apply artifact changes, and only for some products. However, without this set, affected artifact changes may not complete. Hence it is recommended to set this value.
RETURN_MESSAGE_BUFFER_SIZE	This buffer size includes standard output and error messages stored in log files. Default value: 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: ■ B (byte) ■ KB (kilobyte) ■ MB (megabyte) ■ GB (gigabyte)
COMMAND_TIMEOUT	A number and unit. Default value: 3600s (1 hour)	No	Timeout value followed by unit. If command execution takes longer then it will be terminated. Permissible units are: ■ ms (milliseconds) ■ s (seconds) ■ m (minutes) ■ h (hours) ■ d (days)

The `env.properties` file is populated during the provisioning flow. However, if you are administering multiple `IDM_TOP`s using a single Oracle Identity Management provisioning and patching tools install, then you should delete the values 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 you delete the existing values from the `env.properties` file before setting the values. For example, if you are using a POSIX-compliant shell, use the following command:

- `export JAVA_HOME=<JDK absolute path>`

6.3 Using 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:

- Creating a Patch Plan
- Applying Patches
- Applying Artifact Changes

6.3.1 Creating a Patch Plan

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

- Running Oracle Identity Management Patch Manager
- Creating a Patch Plan

6.3.1.1 Running Oracle Identity Management Patch Manager

You run Oracle Identity Management Patch Manager by using the command line utility, `idmpatchmgr`, located in the `$IDM_LCM_TOP/patch/bin` directory (`$IDM_LCM_TOP\patch\bin for Windows`). Its shell script sets the environment and calls the utility. For UNIX, the shell script is `idmpatchmgr.sh` and for Windows, it is `idmpatchmgr.bat`. You can run `idmpatchmgr` with various commands and options.

> **Note:** You must run the Oracle Identity Management Patch Manager on the primordial host to create the patch plan.

The Oracle Identity Management Patch Manager maintains a stateful session to track the patch process coordination with the Oracle Identity Management Patcher tool.

> **Note:** A new patching session cannot be created until the existing session is completed or is aborted.

The 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 following status as detailed in Table 6–3:

The status `COMPLETE` and `INCOMPLETE` are the terminal states; whereas `FAILED` and `ABORTING` are recoverable states.

Table 6–3 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.

The following command shows the basic syntax for the **idmpatchmgr** utility:

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

(Windows) `$IDM_LCM_TOP\patch\bin\idmpatchmgr.bat command [-options]`

In the preceding example, the following variables are used:

- **command**: The `idmpatchmgr` utility manages patching-related activities by using one of the commands as described in Table 6–4.
- **options**: The `idmpatchmgr` commands accept options by using command-line arguments, as described in the specific sections for each `idmpatchmgr` command.

Table 6–4 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.
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`

(Windows) `$IDM_LCM_TOP\patch\bin\idmpatchmgr.bat command -help`

To display basic help for the `idmpatchmgr` command, enter `idmpatchmgr` with no options.

Note: In command syntax examples, the brackets ([]) indicate that the value inside the brackets is optional.

6.3.1.2 Creating a Patch Plan

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

Note: You must run the Oracle Identity Management Patch Manager on the primordial host to create the patch plan.

Syntax

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

(Windows): `$IDM_LCM_TOP\patch\bin\idmpatchmgr.bat apply -patchtop` *patch_top*

Oracle Identity Management Patch Manager performs the following tasks to create the patch plan:

- 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`

Options

Table 6–5 lists the options available for the `apply` command.

Table 6–5 apply Command Options

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

6.3.2 Applying Patches

You run Oracle Identity Management Patcher by using the command line utility, `idmpatch`, located in the `$IDM_LCM_TOP/patch/bin` directory (`$IDM_LCM_TOP\patch\bin` for Windows). Its shell script sets the environment and calls the utility. For UNIX, the shell script is `idmpatch.sh` and for Windows, it is `idmpatch.bat`.

The following command shows the basic syntax for the `idmpatch` utility:

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

(Windows) `$IDM_LCM_TOP/patch/bin/idmpatch.bat run`

Note: You can use the `prereq` option to run only the pre-requisites. This will not stop and start the services or apply and rollback patches.

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

(Windows) `$IDM_LCM_TOP/patch/bin/idmpatch.bat run` *-prereq*

The Oracle Identity Management Patcher `run` command performs the following tasks:

- The Oracle Identity Management Patcher `run` command 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 the Oracle Identity Management Patcher reports that the execution is not possible until execution is complete for the other host.
- It creates the following log file named `status` with the details:

  `$LCM_CONFIG/patch/status/`*currentSessionID*`/hosts/`*currentHostName*`/status`
- When the 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. The next time you use the `run` command, the Oracle Identity Management Patch Manager will display the results.
- On failure, the status log file will be 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`

6.3.3 Applying Artifact Changes

The Oracle Identity Management patching framework supports the application of post patch artifact changes, such as adding an entry within a configuration properties file, or invoking a product MBean. Such changes are optional, and most patches do not include them.

For patches which include them, the Oracle Identity Management Patcher automatically executes the changes after all binary patch application for a single product is completed.

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].

6.3.3.1 Prerequisites

- The post patch artifact changes require additional Perl libraries to perform certain actions such as connecting to the database and executing sql queries. Ensure that `DB.pm` is available in the Perl library.

If Perl is not found, set the Perl library location in the `$PATH` environment variable using the command and path to `DB.pm` as applicable for your deployment. For example:

```
export PATH=/usr/lib/perl5/5.14:$PATH
```

- Set the `ANT_HOME` environment variable in `env.properties` or as an environment variable.

6.3.3.2 Log File

The output of the artifact installation is written to the following log file:

`$LCM_CONFIG/patch/status/`*currentSessionID*`/hosts/`*hostname*`/log/<patch-id>-artifact.log`

If there is any other log file provided in the subroutine as a part of the automation logic, the logs will also be created in the specified location.

6.4 Oracle Identity Management Patching Options

The Oracle Identity Management patching framework supports the following scenarios for applying patches:

- Post Installation Patching
- Ongoing Patching
- Patching DMZ/Disconnected Hosts

6.4.1 Post Installation Patching

Post-installation patches are installed immediately after the install phase of provisioning. The Oracle Identity Management Provisioning Wizard invokes the Oracle Identity Management patching framework for installing the post installation patches using additional options. These are applicable only to patching during provisioning.

For example, patches are applied before any server instances are configured, so the Oracle Identity Management Provisioning Wizard is able to bypass the server stop and start steps. Such options are not supported for ongoing patching in this release.

6.4.2 Ongoing Patching

Ongoing patches are applied by the administrator to running, in-production deployment environment. These may be one-off patches addressing a given bug(s) or security issue(s) or "bundle patches" released for Oracle Identity Management products regularly.

The patch plan is generated by the Oracle Identity Management Patch Manager using the command:

`./idmpatchmgr.sh apply -patchtop` *patchtop location*

Then the Oracle Identity Management Patcher is run on the appropriate host using the following command:

```
./idmpatch.sh run
```

Error messages are displayed if you run the Oracle Identity Management Patcher on the wrong host.

6.4.3 Patching DMZ/Disconnected Hosts

You can deploy Web tier hosts within a DMZ. For this type of setup, a shared network location accessible to both the primordial host and the DMZ hosts, where the `$LCM_CONFIG` directory exists, may not be available.

In this specific case, use the following procedure for running the Oracle Identity Management patching framework on DMZ hosts:

> **Note:** Only Web tier server instances can be run from the DMZ host. Running and patching other servers is NOT supported.

1. Generate the patch plan using the Oracle Identity Management Patch Manager `apply` command on the primordial host.
2. Run the Oracle Identity Management Patcher on the non-DMZ hosts using the `run` command.
3. Before running the Oracle Identity Management Patcher on the DMZ host, run the Oracle Identity Management Patch Manager `progress` command to generate the patch bundle containing the latest session information for the DMZ hosts.
4. After you run the `progress` command, a patch bundle is generated under `$LCM_CONFIG/patch/status/`*session id*`/hosts/`*dmz host name*`/patchbundle.zip`. The `patchbundle.zip` file contains the latest session information for executing the Oracle Identity Management Patcher on the DMZ host.
5. Copy `patchbundle.zip` to the DMZ host.
6. On the DMZ host, extract the zip file under the PARENT directory of `$LCM_CONFIG`.
7. Run the Oracle Identity Management Patcher on the DMZ host using the `run` command.
8. After running the Oracle Identity Management Patcher, copy the status file on the DMZ host from `$LCM_CONFIG/patch/status/hosts/`*dmz hostname*`/status` to the primordial host `$LCM_CONFIG/patch/status/hosts/`*dmz hostname*`/status`.
9. Continue running the Oracle Identity Management Patcher on the appropriate host using the `run` command.

6.5 Monitoring and Troubleshooting

The following section describes how to monitor and troubleshoot the Oracle Identity Management patch sessions.

- Tracking Patch Progress
- Restarting a Failed Patch Session
- Rolling Back Patches
- Aborting a Patch Session

6.5.1 Tracking Patch Progress

The Oracle Identity Management Patch Manager `progress` command validates the existence of the patch session and displays the report with the required patch execution steps.

Syntax

(UNIX) `$IDM_LCM_TOP/patch/bin/idmpatchmgr.sh progress`

(Windows) `$IDM_LCM_TOP/patch/bin/idmpatchmgr.bat progress`

The Oracle Identity Management Patch Manager `progress` command performs the following tasks:

- The patch session status is printed on the console and log files.
- The Oracle Identity Management Patch Manager gets the current session ID from the `$LCM_CONFIG/patch/session/session` file.
- If there is an existing valid session, it scans through the list of Oracle Identity Management Patcher status files for that session under: `$LCM_CONFIG/patch/status/`*currentSessionID*`/hosts/`*hostname*`/status/status` and updates the patch session status accordingly.

  Table 6–7 describes the patch step status values and Table 6–8 describes the patch session status values.

Options

Table 6–6 lists the options available for the `progress` command.

Table 6–6 progress Command Options

Option	Description	Mandatory
`verbose`	Displays detailed status information for all tasks in the currently executing phase.	No
`all`	Displays detailed status information for all tasks.	No

Status

Table 6–7 describes the patch step status values.

Table 6–7 Patch Step Status

State	Description
`PLANNED`	The step has not been executed by the Oracle Identity Management Patcher.
`RUNNING`	The step is currently being executed by the Oracle Identity Management Patcher
`FAILED`	The step execution failed.
`COMPLETED`	The step execution completed successfully.

Table 6–8 describes the patch session status values.

Table 6–8 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.

6.5.2 Restarting a Failed Patch Session

If your patch session is in a HALTED state due to a failed execution step, you can use the retry command to run the session after you resolve the issue that caused the failure. Use the retry command, as shown in the following example:

(UNIX) $IDM_LCM_TOP/patch/bin/idmpatch.sh retry

(Windows) $IDM_LCM_TOP/patch/bin/idmpatch.bat retry

Note: You can use the prereq option to run only the pre-requisites. This will not stop and start the services or apply and rollback patches.

(UNIX) $IDM_LCM_TOP/patch/bin/idmpatch.sh retry *-prereq*

(Windows) $IDM_LCM_TOP/patch/bin/idmpatch.bat retry *-prereq*

The Oracle Identity Management Patcher retry command performs the following tasks:

- The Oracle Identity Management Patcher retry command validates the existence of the patch session with execution step in FAILED or RUNNING state for the host where the tool is running. The retry command identifies the step which is in FAILED status and starts the patching execution flow from the point of failure.
- It validates the existence of the status file under the $LCM_CONFIG/patch/status/*currentSessionID*/hosts/*currentHostName*/status folder.
- It updates the status log file with key = *step-id* and value = RUNNING and starts the step execution using the step executor. 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 then continues to the next step from the execution plan.
- If there are no steps to be executed for the current host, it halts the execution by sending an update to the administrator regarding the further steps to be executed.
- The retry command also updates the session status. The Oracle Identity Management Patch Manager is immediately notified of the results.

  However, only in case of a DMZ host, you should copy the status file to the primordial host to notify the DMZ host status to the Oracle Identity Management Patch Manager.
- On failure, the status log file will be updated with key = step-id and value = FAILED and the execution is halted.
- The retry 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

6.5.3 Rolling Back Patches

You rollback patches by running the `idmpatchmgr.sh rollback` command. The Oracle Identity Management Patch Manager `rollback` command validates the given patch top location and validates the existence of the patch session with the status `ACTIVE` or `FAILED`.

> **Note:** A new patching session cannot be created until the existing session is completed or aborted.

Syntax

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

(Windows) `$IDM_LCM_TOP/patch/bin/idmpatchmgr.bat rollback -patchtop` *patch_top*

The Oracle Identity Management Patch Manager `rollback` command performs the following tasks:

- The `rollback` command validates the given patch top location and 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 composite bundle patch from the provided patch top. This bundle patch is internally used to generate the patch plan. The composite bundle patch is created in the location: `$LCM_CONFIG/patch/patches`.
- If the `rollback` command is repeated or if the status of the patch session is `ACTIVE` or `FAILED`, the Oracle Identity Management Patch Manager displays an error message stating that it cannot execute the `rollback` command, since an active or failed patch session already exists. The current session output is displayed.
- A patch plan is generated with instructions for rolling back patches using the topology store information and composite bundle patch.
- The `rollback` command generates the patch plan in the following location in html and text formats:
  `$LCM_CONFIG/patch/status/`*current-sessionID*`/manager/log/PatchInstructions.html`

  `$LCM_CONFIG/patch/status/`*current-sessionID*`/manager/log/PatchInstructions.text`
- On successful patch plan generation, the Oracle Identity Management Patch Manager starts a new patch session with `ACTIVE` status and adds the execution step state for each host as a child to the patch session with status `PLANNED`.

  The patch session is stored in the `$LCM_CONFIG/patch/session/session` file. The steps information is stored in the $LCM_CONFIG/patch/session/step file.
- The rollback 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

6.5.4 Aborting a Patch Session

You can cancel a patch session by running the `abort` command. It validates the existence of the patch session with status of `ACTIVE/FAILED/ABORTING`. It changes the status of the patch session to `INCOMPLETE` and halts the execution.

Syntax

(UNIX) `$IDM_LCM_TOP/patch/bin/idmpatchmgr.sh abort`

(Windows) `$IDM_LCM_TOP/patch/bin/idmpatchmgr.bat abort`

The Oracle Identity Management Patch Manager `abort` command performs the following tasks:

- The `abort` command is a hard abort and you have to manually restore the product where the failure occurred. Depending on factors like your change control requirements and backup strategy you can choose to:
  - remove all traces of the patch session, by reverting all products.

  OR

  - continue running any products for which patches were successfully applied.
- The `progress` command will display the session status as `INCOMPLETE`.
- If you execute the `apply` or `rollback` command after the `abort` command, it starts a new session since the session status is `INCOMPLETE`.
- To completely delete session information, you can run the `abort` command twice or you can start a new session by using the `apply` or `rollback` command.

6.5.5 Troubleshooting

For more information about general troubleshooting scenarios for the Oracle Identity Management patching framework, see "Chapter 7, Troubleshooting Identity Management Provisioning" in the *Oracle Fusion Middleware Administrator's Guide for Identity Management Provisioning (Oracle Fusion Applications Edition)*.

7

Patching Oracle Fusion Middleware Extensions for Applications

This chapter describes how to apply patches to update Oracle Fusion Middleware Extensions for Applications (Applications Core).

This chapter contains the following topics:

7.1 Patching Applications Core Database Artifacts

The same set of patching related software and database tables is used by both Oracle Fusion Applications Patch Manager and Oracle Fusion Middleware Extensions for Applications (Applications Core). Oracle Fusion Applications Patch Manager and Applications Core each have their own separate Oracle home. Product specific shell scripts used for database patching exist in each corresponding Oracle home location. The separate scripts are provided to support each product's individual patching requirements. The scripts are uniquely defined to reference the appropriate Oracle home, set the patching configuration and environment, and then call the appropriate utility for patching.

Oracle Fusion Applications Patch Manager automatically calls Oracle Fusion Applications AutoPatch (AutoPatch) when it detects patch metadata that indicates a patch contains database content. However, you run AutoPatch directly to apply an applications database patch for Applications Core.

Oracle Fusion Applications AutoPatch is used to apply patches that contain fixes or updates to applications database artifacts. It gathers the necessary information about the applications system and performs the following tasks required to apply a patch:

- Reads patch metadata to determine patch dependencies and requirements
- Uploads patch information from a prior patch session to the database (if needed)
- Reads and validates the patch driver file
- Compares version numbers of object modules from the product libraries and version numbers of the existing files against the patch files
- Backs up all existing files that will be changed by the patch

- Copies files
- Updates database objects
- Saves patch information to the database
- Takes no action if a patch contains no new updates to files or database objects in your system
- Detects if there is a previously failed Oracle Fusion Applications AutoPatch session and attempts to recover that session

7.1.1 Artifacts Supported by Oracle Fusion Applications AutoPatch

Table 7–1 lists the types of artifacts that are supported by Oracle Fusion Applications AutoPatch.

Table 7–1 Database Artifacts Supported by Oracle Fusion Applications AutoPatch

Artifact Type	Description	Patching Recommendations
Applications Seed Data (XML,XLF files)	Examples include static lists of values, functional or error messages, and lookup values. Any non-transactional data values loaded into your 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 files)	Examples include tables, triggers, views, sequences, synonyms, queues, queue tables, policies, and contexts.	None
PL/SQL objects (PDQ, PCB 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

7.1.2 Running Oracle Fusion Applications AutoPatch

You run Oracle Fusion Applications AutoPatch by using the command line utility, `adpatch`, located in the *`ATGPF_ORACLE_HOME`*`/lcm/ad/bin` directory. This is the directory under MW_HOME that contains the Applications Core code. The utility's shell script sets the environment for *`ATGPF_ORACLE_HOME`* and calls the utility. For UNIX, the shell script is `adpatch.sh` and for Windows, it is `adpatch.exe`. You direct the way `adpatch` operates by adding arguments and options to the command. Command line arguments use the `token=value` format.

The following command shows the basic syntax for the `adpatch` utility:

```
(UNIX) ATGPF_ORACLE_HOME/lcm/ad/bin/adpatch.sh patchtop=complete_path_to_unzipped_
patch_directory \
driver=driver_name workers=number_of_workers [optional arguments=value]

(Windows) ATGPF_ORACLE_HOME\lcm\bin\adpatch.exe patchtop=complete_path_to_
unzipped_patch_directory \
driver=driver_name workers=number_of_workers [optional arguments=value]
```

Table 7–2 displays the arguments used by Oracle Fusion Applications AutoPatch:

Table 7–2 Arguments Used by Oracle Fusion Applications AutoPatch

Argument	Description	Mandatory	Default Value	Example
abandon	Abandons a failed patching session	No	n for no	abandon=y
apply	Applies the patch	No	y for yes	apply=n
backup	Allows you to override the backup directory for files that are copied	No	*ATGPF_ORACLE_HOME*/admin/pbackup	backup=*ATGPF_ORACLE_HOME*/admin
defaults file	File delivered by Oracle that contains the environment information required to apply a patch	Recommended	None	defaultsfile=*APPLICATIONS_CONFIG*/atgpf/admin/defaults.txt
driver	Name of the patch driver file	Yes	None	driver=u*patch_number*.drv
help	Accesses help for adpatch	No	n for no	help=y
interactive	Allows you to apply a patch non-interactively using information from the defaults file	Yes	y for yes	interactive=n
logfile	Name for log file	No	adpatch_apply.log	logfile=adpatch_apply_*patch_number*.log
loglevel	Records messages in the log file at the level you specify. Enter a value to override the default log level of `INFO`. See Section 11.1, "Oracle Fusion Applications Patch Manager Logging"	No	NOTIFICATION:16	loglevel=WARNING
logtop	Top level directory where log files are written	No	*ATGPF_ORACLE_HOME*/admin/FUSION/log	logtop=/tmp/admin/log
max_trial_time	The maximum time in minutes that deferred tasks are attempted	No	10 minutes	max_trial_time=5
parallel_index_threshold	Threshold block count in each table, which when exceeded for a tables, causes its indexes to be created using parallel slaves	No	10000	parallel_index_threshold=12000
patchtop	Top level directory for the unzipped patch	Yes	None	patchtop=/temp/patches/*patch_number*
printdebug	Displays additional debug information	No	n for no	printdebug=y
restart	Resumes a failed patching session	No	None	restart=y

Table 7–2 (Cont.) Arguments Used by Oracle Fusion Applications AutoPatch

Argument	Description	Mandatory	Default Value	Example
wait_on_ failed_job	Waits for user intervention when all tasks fail. If set to no, AutoPatch exits when all tasks assigned to workers have failed and require user intervention.	Recommended	n for no	wait_on_failed_job=y
workers	Number of workers to use for database patching tasks	Yes	None	workers=8

To access help for the `adpatch` command, type `adpatch help=y`. The output of the `help` argument provides a list of the options you can use to refine the operation of `adpatch`, along with a brief description of each option.

7.1.2.1 Applying Patches Non-interactively

Oracle Fusion Applications AutoPatch should be run non-interactively from the command line by using the `defaults.txt` file that contains the information required to apply the patch. During the installation and configuration for Applications Core, the `defaults.txt` file is created in this location: *`APPLICATIONS_CONFIG`*`/atgpf/admin/defaults.txt`.

7.1.2.2 End-to-End Patching Process

The end-to-end process of applying individual patches using AutoPatch, includes the following steps.

Note: As part of the patching process, customers have their own backup and recovery management process. Oracle recommends that you always have a current backup before applying a patch.

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

Step 1 Research Issue That Must be Resolved by a Patch

When you have an issue that must be resolved by a patch for Applications Core, file a service request with Oracle Support Services or research the issue on My Oracle Support:

`http://support.oracle.com`

Step 2 Obtain and Unzip the Patches

Upon determining that you need new patches, download the patches from My Oracle Support. Unzip the patch ZIP files in your `PATCH_TOP` directory. `PATCH_TOP` can be any location in your file system.

Step 3 Read the README File

Read the README file that accompanies each patch. This file contains important information and instructions that must be followed.

If a patch contains preinstallation or postinstallation manual steps, they are described in the patch README file.

Step 4 Prepare the System

To prevent locks on patched objects and other data issues during patching of database artifacts, review and perform the following checklist before patching the target database system:

- Confirm that the database system is in an idle state
- Confirm that there are no active jobs or processes running against the database
- Stop all background jobs, including jobs in the database and active processes
- Manually shut down the Oracle Enterprise Scheduler Service (ESS) servers, especially when a patch contains a PL/SQL package, by performing the following steps:

  a. Stop the ESS request processor and dispatcher to prevent new requests from being processed. See "Starting and Stopping Oracle Enterprise Scheduler Service Components" in the *Oracle Fusion Applications Administrator's Guide* for more information.

  b. Cancel any in-progress requests. For more information, see "Cancelling Oracle Enterprise Scheduler Job Requests" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Enterprise Scheduler* for more information.

  c. Shutdown the ESS WebLogic Server Managed server. See the "Starting and Stopping" table, specifically the "Managed Servers for the Oracle Fusion applications and Oracle Fusion Middleware components" row, in the *Oracle Fusion Applications Administrator's Guide* for more information.

Step 5 Apply the Prerequisite Patches

If any prerequisite patches are required, as described in the README file, apply those patches with Oracle Fusion Applications AutoPatch.

Step 6 Identify the correct Oracle home

Identify the Applications Core Oracle home (*ATGPF_ORACLE_HOME*). This is the directory under *MW_HOME* that contains the Applications Core code.

Step 7 Apply the Patch

Apply the patch using the `adpatch` command. An example of the `adpatch` command follows:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/adpatch.sh defaultsfile=APPLICATIONS_
CONFIG/atgpf/admin/defaults.txt \
patchtop=complete_path_to_unzipped_patch driver=upatch_number.drv \
workers=number_of_database_workers interactive=no [logfile=log_file_name]\
[logtop=ATGPF_ORACLE_HOME/admin/FUSION] [loglevel=level] wait_on_failed_job=yes

(Windows) FA_ORACLE_HOME\lcm\ad\bin\adpatch.exe defaultsfile=APPLICATIONS_
CONFIG/atgpf/admin\defaults.txt \
patchtop=complete_path_to_unzipped_patch driver=upatch_number.drv \
workers=number_of_database_workers interactive=no [-logfile=log_file_name]\
[logtop=ATGPF_ORACLE_HOME\admin\FUSION] [loglevel=level] wait_on_failed_job=yes
```

Step 8 Monitor the Application of the Patch

You can monitor the progress of the patching session and verify its successful completion by using the AD Controller utility. For more information, see Section 7.4, "Monitoring and Troubleshooting Applications Core Patching Sessions".

Step 9 Review the Log Files

Review the log files when the patch session is complete. For more information, see Section 7.3, "Log Files".

Step 10 Apply Postinstallation Patches or Run Manual Steps

If any postinstallation patches are required, as indicated in the README file, apply them. Run any manual steps that are described in the README file.

7.2 Patching Applications Core Middleware Artifacts

You use the OPatch utility to patch Applications Core middleware artifacts. All patches are available for download from My Oracle Support.

To patch Applications Core middleware artifacts

1. Identify the Applications Core Oracle home (*ATGPF_ORACLE_HOME*). This is the directory under *MW_HOME* that contains the Applications Core code.
2. Set the *ORACLE_HOME* to *ATGPF_ORACLE_HOME*.

   Example:

   ```
   setenv ORACLE_HOME /server01/mwhome/Oracle_atgpf
   ```

   Or:

   ```
   export ORACLE_HOME=/server01/mwhome/Oracle_atgpf
   ```

3. Create a temporary directory to stage the Applications Core patches.

   ```
   mkdir Temp_Directory
   ```

4. Unzip the OPatch patches in the temporary directory.
5. Review the README.txt file for information about any preinstallation and postinstallation steps.
6. Apply the patches using OPatch. Note that you must use the OPatch from *FA_ORACLE_HOME* instead of the one from the *ATGPF_ORACLE_HOME*.

   ```
   cd patch_top_directory
   FA_ORACLE_HOME/OPatch/opatch apply -invPtrLoc ORACLE_HOME/oraInst.loc
   ```

   For more information about OPatch, see "Patching Oracle Fusion Middleware with Oracle OPatch" in the *Oracle Fusion Middleware Patching Guide*.
7. Stop and start the WebLogic server. For more information, see "Starting and Stopping a Product Family Oracle WebLogic Server Domain" in the *Oracle Fusion Applications Administrator's Guide*.
8. Stop and start the Administration Server and all Managed Servers. For more information, see "Starting and Stopping" in the *Oracle Fusion Applications Administrator's Guide*.
9. Run the postinstallation steps, if applicable.

10. Apply the 'Deployment of Fusion APPS Attachments in UCM server OPatch' patch to your *ECM_ORACLE_HOME*. Follow the same instructions in Steps 1 through 9 to apply this patch.

7.2.1 Patching Global Menus in FndSetup

MAR artifacts that contain Global Menus are delivered in Applications Core OPatch patches and copied to this location when you apply the patch using OPatch:.

```
ATGPF_ORACLE_HOME/atgpf/applications/exploded/FndSetup.ear/FndAppsMenuData.mar
```

After you apply the patch with OPatch, you must copy this MAR artifact to a specific location in *FA_ORACLE_HOME* and start the corresponding Managed Server. Use the following example syntax to copy the file:

```
cp ATGPF_ORACLE_HOME/atgpf/applications/exploded/FndSetup.ear/FndAppsMenuData.mar
FA_ORACLE_HOME/fusionapps/applications/fs/deploy/FndSetup.ear
```

7.2.2 Patching the Flexfield SOA Synch Composite

Extensible flexfields require synchronization between the Service Oriented Architecture (SOA) Metadata Service (MDS) and Oracle Fusion Applications MDS, which is delivered in the `sca_UpdateSOAMDS_`*`rev_number`*`.jar` file. When updates to this jar are delivered in a patch, you must manually deploy the SOA composite after you apply the patch using OPatch. To deploy the SOA composite, log in to Oracle Fusion Applications Control as the WebLogic user, navigate to **SOAServer** and select **Deploy**. For more information, see "Accessing Fusion Applications Control" in the *Oracle Fusion Applications Administrator's Guide*.

7.3 Log Files

Oracle Fusion Applications AutoPatch creates log files in the *`ATGPF_ORACLE_HOME`*`/admin/FUSION` directory. You specify the name of the main log file by using the `logfile` argument when you run Oracle Fusion Applications AutoPatch. The related log files are named based on the name of the main log file. For example, if the name of the main log file is `adpatch_123456_apply.log`, then the worker logs for the session are named `adpatch_123456_apply_worker_`*`worker_number`*`.log`.

Table 7–3 displays the Oracle Fusion Applications AutoPatch log files.

Table 7–3 Log Files for Oracle Fusion Applications AutoPatch

Log file name	Log file description
adpatch.log	Main log file; Oracle recommends you override the default name by including the patch number in the name of the log file.
adpatch.lgi	Contains informational messages.
adlibin.log	Contains information about moving C object files into the C library.
adlibout.log	Contains information about moving C object files out of the C library.
adpatch_ worker*workernumber*.log	Worker log files for database tasks that run in parallel.
*language_filename*_ldt.log	Seed data loader files.

7.4 Monitoring and Troubleshooting Applications Core Patching Sessions

The AD Controller utility, `adctrl`, can monitor, determine the status, and control the progress of the workers called by AutoPatch to update database content. For more information about workers, see Section 3.1.2.1, "Worker Processes".

The AD Controller utility can monitor database patching sessions that were started by both Oracle Fusion Applications AutoPatch and Oracle Fusion Applications Patch Manager. To start `adctrl` to monitor Applications Core patching sessions, make sure you start the session in *ATGPF_ORACLE_HOME*.

The following sections offer steps for troubleshooting issues that may occur during patching sessions for database content.

- Starting AD Controller
- Reviewing Worker Status
- Determining Why a Worker Failed
- Restarting a Failed Worker
- Restarting a Failed Patching Session
- Abandoning a Failed Patching Session
- Applying a Patch to the Wrong Oracle Home

7.4.1 Starting AD Controller

Follow these steps to start AD Controller:

1. Start AD Controller with the `adctrl` command.

   ```
   (UNIX) FA_ORACLE_HOME/fusionapps/atgpf/lcm/ad/bin/adctrl.sh
   (Windows) FA_ORACLE_HOME\fusionapps\atgpf\lcm\ad\bin\adctrl.cmd
   ```

   It prompts you to:

   - Confirm the value of the Applications Core 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. You can press **Enter** at any time to return to the AD Controller main menu.

7.4.2 Reviewing 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. You can use AD Controller to determine the status of workers and manage worker actions.

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

1. Start AD Controller. For more information, see Section 7.4.1, "Starting AD Controller".
2. Review worker status.

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

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

  Table 7–4 describes the status that may be assigned to a worker.

Table 7–4 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	You fixed 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 Section 7.4.3, "Determining Why a Worker Failed" for assistance in fixing the problem so AutoPatch can complete its processing.

7.4.3 Determining Why a Worker Failed

When a worker fails its task, you do not have to wait until the other workers and the manager stop. You can review the worker log files to determine what caused the failure. Workers do not proceed to run tasks in the subsequent phase until all tasks in the current phase complete successfully. You must take action 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 from you.

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 runtime of the task is less than 10 minutes. If the deferred task failed a third time, or if its runtime is greater than 10 minutes, the task stays at a failed status and the worker waits for manual intervention. Action by you is then required because the worker stops any further processing.

Follow these steps to find out why a worker failed:

1. Follow the steps in Section 7.4.2, "Reviewing Worker Status" to find the worker number that failed.
2. Open and review the log file for the failed worker to determine the cause of the worker failure.

   Each worker logs the progress of tasks assigned to it in `adpatch_worker`*`workernumber`*`.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 Section 7.3, "Log Files".

3. Determine how to fix the problem that caused the failure.

   Resolve the error using the information provided in the log files. 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 you resolve the problem that caused the failure, restart the failed worker.

   Select **Tell worker to restart a failed job** from the AD Controller main menu to tell the worker to restart the failed task. For more information, see Section 7.4.4, "Restarting a Failed Worker".

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.

7.4.4 Restarting a Failed Worker

If a worker job failed or if you terminated a hanging worker process, you must manually restart the worker. Some worker processes spawn other processes called child processes. If you terminate a child process that is hung, the worker that spawned the process shows *Failed* as the status. After you fix the problem, you must restart the failed worker. After the worker restarts, the associated child processes start as well.

Follow these steps to restart a failed worker:

1. Start AD Controller. For more information, see Section 7.4.1, "Starting AD Controller".
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 Section 11.5.5, "Terminating a Hung Worker Process".

   The worker will restart its assigned tasks and spawn the associated child processes.

7.4.5 Restarting a Failed Patching Session

If your patch session failed, you can restart the session after you resolve the issue that caused the failure. Use the `restart` argument, as shown in the following example:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/adpatch.sh defaultsfile=APPLICATIONS_
CONFIG/atgpf/admin/defaults.txt \
patchtop=/mypatches/1213456 driver=u123456.drv workers=8 \
interactive=n logfile=adpatch_apply_123456.log loglevel=WARNING]\
logtop=ATGPF_ORACLE_HOME/admin/FUSION wait_on_failed_job=y restart=y

(Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd defaultsfile=APPLICATIONS_
CONFIG/atgpf/admin\defaults.txt \
patchtop=\mypatches\123456 driver=u123456.drv workers=8 \
interactive=no logfile=adpatch_apply_123456.log loglevel=WARNING \
logtop=ATGPF_ORACLE_HOME\admin\FUSION wait_on_failed_job=yrestart=y
```

7.4.6 Abandoning a Failed Patching Session

If your patch session failed and you do not want to restart it, you must abandon the session to update the database tables, allowing you to start a new patch session. Use the `abandon` argument, as shown in the following example:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/adpatch.sh defaultsfile=APPLICATIONS_
CONFIG/atgpf/admin/defaults.txt \
patchtop=/mypatches/1213456 driver=u123456.drv \
interactive=no logfile=adpatch_abandon_123456.log loglevel=WARNING\
logtop=ATGPF_ORACLE_HOME/admin/FUSION abandon=y

(Windows) FA_ORACLE_HOME\lcm\ad\bin\adpatch.cmd defaultsfile=APPLICATIONS_
CONFIG/atgpf/admin\defaults.txt \
patchtop=/mypatches/1213456 driver=u123456.drv \
interactive=no logfile=adpatch_abandon_123456.log loglevel=WARNING\
logtop=ATGPF_ORACLE_HOME\admin\FUSION abandon=y
```

Note: There can be only one patching session active for Oracle Fusion Applications and Oracle Middleware Extensions for Applications at a time. If there is a failed Oracle Fusion Applications Patch Manager session, see Section 11.3.2, "Abandoning a Failed Patching Session".

7.4.7 Applying a Patch to the Wrong Oracle Home

You must apply Applications Core patches to the correct Oracle home. For example, if you try to apply an Applications Core patch to the Oracle Fusion Applications Oracle home, you receive this error message:

```
Error: Incompatible Patch.
 Cannot apply ATGPF patch on FA_ORACLE_HOME
```

To resolve this issue, ensure that you are applying the patch to *ATGPF_ORACLE_HOME*.

7.5 Performing System Maintenance Tasks

AD Administration is a standalone utility that performs administration maintenance tasks for the products in your Applications Core Oracle home. The general purpose of the maintenance tasks is to keep your Applications Core files and database objects up-to-date. Some maintenance tasks should be performed systemwide on a regular basis, while others are required infrequently.

Run AD Administration by using the command-line utility, `adadmin`. This utility has a shell script that sets the environment and calls the utility. For UNIX, the shell script is `adadmin.sh` and for Windows it is `adadmin.cmd`. To run AD Administration for Applications Core, use this command syntax:

```
(UNIX) FA_ORACLE_HOME/fusionapps/atgpf/lcm/ad/bin/adadmin.sh [argument]
(Windows) FA_ORACLE_HOME\fusionapps\atgpf\lcm\ad\bin\adadmin.cmd [argument]
```

Refer to Chapter 10, "Performing System Maintenance Tasks" for how to use AD Administration. This chapter describes using AD Administration for both Applications Core and Oracle Fusion Applications Patch Manager. The only difference is the location of the command you use to start AD Administration.

8

Patching Oracle Fusion Functional Setup Manager

This chapter describes how to apply patches to update Oracle Fusion Functional Setup Manager.

This chapter contains the following topics:

8.1 Introduction to Oracle Fusion Functional Setup Manager

Oracle Fusion Functional Setup Manager enables rapid and efficient planning, implementation, and deployment of Oracle Fusion Applications through self-service administration. The code for Functional Setup Manager is maintained in the `ATGPF_ORACLE_HOME` for Oracle Fusion Applications

8.2 Patching Functional Setup Manager Database Artifacts

The same set of patching related software and database tables is used by both Functional Setup Manager and Oracle Fusion Middleware Extensions for Applications (Applications Core). You run Oracle Fusion Applications AutoPatch (AutoPatch) directly to apply an applications database patch for Functional Setup Manager. For more information, see Section 7.1, "Patching Applications Core Database Artifacts". The patch README file for Functional Setup Manager patches contains information to assist you in performing the patching steps correctly.

8.2.1 Database Artifacts Supported for Functional Setup Manager

Table 8–1 lists the types of artifacts that are supported for Functional Setup Manager.

Table 8–1 Database Artifacts Supported for Functional Setup Manager

Artifact Type	Description	Patching Recommendations
Applications Seed Data (XML,XLF files)	Examples include static lists of values, functional or error messages, and lookup values. Any non-transactional data values loaded into your 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 files)	Examples include tables, triggers, views, sequences, synonyms, queues, queue tables, policies, and contexts.	

8.2.2 Patching the Functional Setup Manager Database

To apply a Functional Setup Manager database patch, follow the steps in Section 7.1.2, "Running Oracle Fusion Applications AutoPatch".

8.3 Patching Functional Setup Manager Middleware Artifacts

Functional Setup Manager requires middleware artifacts that are deployed on the WebLogic Server. You use OPatch to apply a patch to update Functional Setup Manager artifacts.

8.3.1 Middleware Artifacts Required by Functional Setup Manager

Functional Setup Manager requires certain middleware artifacts that may need to be patched. Table 8–2 displays these middleware artifacts along with actions that need to be performed manually after you apply the patch and what must be running while you apply the patch and perform the manual actions.

Table 8–2 Oracle Fusion Middleware Artifacts Required by Functional Setup Manager

Artifact Type	Actions to Be Performed Manually	What Must Be Running During Patching and Manual Actions
Oracle ADF JAR	Start and stop the relevant servers that host the Java EE application.	Administration Server, node manager, database, LDAP.
Oracle Business Intelligence Publisher (Reports and Captions)	Shut down the BI Presentation server before patching, deploy to Business Intelligence repository, and start the BI Presentation server after patching.	Shut down the BI presentation server before patching. See Section 4.4, "Patching Oracle Business Intelligence Publisher Artifacts".
Oracle Enterprise Scheduler Service MAR	Stop and start the relevant servers that host the Java EE application.	Administration Server, node manager, database.

8.3.2 Patching Functional Setup Manager Middleware Artifacts

You use the OPatch utility to patch Functional Setup Manager middleware artifacts. The patch README file for Functional Setup Manager patches contains information to assist you in performing the patching steps correctly.

For more information, see Section 7.2, "Patching Applications Core Middleware Artifacts".

8.3.3 How to Patch Applications Policies (jazn-data.xml)

The following steps must be performed if you patch applications policies.

1. Back up function security policies in the Oracle Internet Directory (OID) Policy store by following the steps in "Prerequisites to Patching Policies" in the *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*.
2. Deploy on the FSCM policy stripe using Oracle Authorization Policy Manager. The JAZN files are available in *ATGPF_ORACLE_HOME*`/setup` and *ATGPF_ORACLE_HOME*`/setupEss` directories. For more information, see "Upgrading Oracle Fusion Applications Policies" in *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*. Note that FSM is only one of the components in the FSCM stripe and there are other components in the same stripe that you should not update.

8.4 Log Files

AutoPatch creates log files in the *APPLICATIONS_CONFIG*`/atgpf/logs` directory. For more information, see Section 7.3, "Log Files".

8.5 Monitoring and Troubleshooting Patching Sessions

The AD Controller utility, `adctrl`, can monitor and control the progress of the workers called by AutoPatch to update database content. For more information, see Section 7.4, "Monitoring and Troubleshooting Applications Core Patching Sessions".

8.6 Performing System Maintenance Tasks

AD Administration is a standalone utility that performs administration maintenance tasks for the products in *ATGPF_ORACLE_HOME*. The general purpose of the maintenance tasks is to keep your *ATGPF_ORACLE_HOME* files and database objects up-to-date. Some maintenance tasks should be performed system-wide on a regular basis, while others are required infrequently. For more information, see Section 7.5, "Performing System Maintenance Tasks".

9

Patching Oracle Fusion Applications Functional Core

This chapter describes how to apply patches to update Oracle Fusion Applications Functional Core.

This chapter contains the following topics:

9.1 Introduction to Oracle Fusion Applications Functional Core

Oracle Fusion Applications Functional Core enables the integration of other Oracle Fusion applications with Oracle Fusion Functional Setup Manager. It provides:.

- APIs and web services to handle the registration of topology objects that are used by the Provisioning software to provision offerings
- The shared libraries required for developing Task Execution Programs for Setup Tasks and Export and Import Services to extend Oracle Fusion Functional Setup Manager content
- APIs and Web services for allowing integration with Oracle Fusion Functional Setup Manager

9.2 Patching Oracle Fusion Applications Functional Core Database Artifacts

The database patches for Applications Functional Core contain Oracle SQL*PLUS scripts. To apply Applications Functional Core database changes, run the SQL*PLUS scripts against the FUSION schema. The patch README file for Applications Functional Core patches contains information to assist you in performing the patching steps correctly.

9.3 Patching Oracle Fusion Applications Functional Core Middleware Artifacts

Oracle Fusion Applications Functional Core uses Oracle ADF Jar files, which may need to be patched.

To patch Oracle Fusion Applications Functional Core middleware artifacts

1. Ensure that the Administration Server, node manager, and database are running.
2. Use the OPatch utility to patch Oracle Fusion Applications Functional Core middleware artifacts. The patch README file for Oracle Fusion Applications Functional Core patches contains information to assist you in performing the patching steps correctly. Follow the steps in Section 7.2, "Patching Applications Core Middleware Artifacts".
3. Stop and start the relevant servers that host the impacted Java EE application.

9.4 Log Files

Oracle Fusion Applications AutoPatch creates log files in the *APPLICATIONS_CONFIG*`/atgpf/logs` directory. For more information, see Section 7.3, "Log Files".

9.5 Monitoring and Troubleshooting Patching Sessions

The AD Controller utility, `adctrl`, can monitor and control the progress of the workers called by Oracle Fusion Applications AutoPatch (AutoPatch) to update database content. For more information, see Section 7.4, "Monitoring and Troubleshooting Applications Core Patching Sessions".

9.6 Performing System Maintenance Tasks

AD Administration is a standalone utility that performs administration maintenance tasks for the products in *ATGPF_ORACLE_HOME*. The general purpose of the maintenance tasks is to keep your *ATGPF_ORACLE_HOME* files and database objects up-to-date. Some maintenance tasks should be performed system-wide on a regular basis, while others are required infrequently. For more information, see Section 7.5, "Performing System Maintenance Tasks".

10

Performing System Maintenance Tasks

This chapter describes the process of running Oracle Fusion Applications utilities to perform routine system maintenance tasks.

This chapter contains the following topics:

10.1 Introduction to AD Administration

AD Administration is a standalone utility that performs administration maintenance tasks for the products in your Oracle Fusion Applications Oracle home and your Applications Core Oracle home. The general purpose of the maintenance tasks is to keep your Oracle Fusion Applications files and database objects up-to-date. Some maintenance tasks should be performed systemwide on a regular basis, while others are required infrequently.

Run AD Administration by using the command-line utility, `adadmin`. This utility has a wrapper script that sets the environment and calls the utility. For UNIX, the wrapper is `adadmin.sh` and for Windows it is `adadmin.cmd`. This brings you to the AD Administration main menu, where you select from a list of submenus that contain these task groups:

- Maintain Snapshot Information
- Maintain Applications Database Entities

After the utility starts, it requires that you enter system information by responding to a series of prompts.

10.1.1 AD Administration Main Menu

The AD Administration main menu is the gateway to submenus, where you select the individual maintenance tasks. For example, selecting **Maintain Snapshot Information** takes you to the Maintain Snapshot Information submenu, where you can run tasks related to snapshots of the application-related files of a given Oracle home.

To select a submenu, enter the number of the submenu at the prompt.

10.1.2 Valid Command-Line Arguments

You may want to add arguments to the `adadmin` command to customize its operations. For example, you can add `defaultsfile=`*`defaultsfile_location`* to indicate that you want to create a defaults file for use in running maintenance tasks non-interactively, without the need to respond to prompts. For more information, see Section 10.4, "Running Maintenance Tasks Noninteractively".

10.1.3 Prompts and Password Security

When you run AD Administration, it prompts for information about your system. Prompts typically include a description of the information needed, and usually, a default answer is enclosed in brackets. Enter a response to the prompt, or press Enter to accept the default. AD Administration prompts include:

- Confirmation of your Oracle home location
- Log file name
- Batch size
- Confirmation of the correct database

Some submenu options may trigger additional system prompts associated with that task.

For increased security, the `adadmin` command does not prompt for passwords, nor does the program store passwords in clear text or obfuscated format in the restart or defaults file. Instead, it uses the Credential Store Framework (CSF) to store and retrieve passwords from the CSF wallet. For more information, see Section 2.4, "Oracle Fusion Applications Patching and the Security Model".

10.1.4 Starting AD Administration

Follow these steps to start the AD Administration utility:

1. From any directory, start AD Administration with this command, adding any arguments that apply to this session:

   ```
   (UNIX) FA_ORACLE_HOME/lcm/ad/bin/adadmin.sh [argument]
   (Windows) FA_ORACLE_HOME\lcm\ad\bin\adadmin.cmd [argument]
   ```

   AD Administration starts and displays the first prompt.

2. Respond to each prompt by either entering a value or pressing Enter to accept the default value. When you have responded to all of the prompts, the AD Administration main menu appears.
3. Select a menu or submenu option by entering its number in the brackets at the bottom of the screen.
4. From the next menu, select any task by entering its number.
5. You can exit AD Administration from any submenu screen by entering the option number for **Exit AD Administration**.

10.2 Maintaining Snapshot Information

A snapshot represents each unique combination of Oracle home, product, and version at any given point of time. It includes information on each file maintained by Oracle Fusion Applications AutoPatch (AutoPatch) and a summary of database patches that have been applied to that Oracle home. The information maintained for each file

includes the subdirectory where it is located, the name of the file and its version, and the file metadata. The files tracked in the snapshot are those files located in the following *product family* subdirectories: `components`, `db`, and `patch`. Because snapshots track only files for which there can be different versions as shipped by Oracle, they do not track C executable files or log files.

Each time you run Oracle Fusion Applications Patch Manager (Patch Manager) or AutoPatch, the snapshot is updated to include any new file copied to the Oracle home. When you apply a patch, Patch Manager performs version-checking against the file to determine which files must be replaced.

The snapshot can be either a current view or a named view.

Current View Snapshot

A **current view** snapshot is a copy of the current file system of the Oracle home. Patch Manager creates the current view. The current view is then updated every time a patch is applied and every time the Update Current View Snapshot process runs to maintain a consistent view. Patch Manager uses the current view snapshot to determine if all prerequisite patches have been applied to the Oracle home.

For more information, see Section 10.2.1.2, "Update Current View Snapshot".

Named View Snapshot

A **named view** snapshot is a copy of the current view snapshot that you create at a particular time and it is never updated. You can compare one or more named view snapshots to the current view snapshot or refer to previous states of your Oracle home, as needed. Complete file, file version information, and patch history is stored in a named view snapshot.

Properly maintained snapshots allow you to:

- Compare two different Oracle homes
- Compare the same Oracle home at different times
- Track the current status of an Oracle home

Snapshot Tables

Snapshot information is stored in several tables. They contain details such as:

- The name, type, and other attributes of each snapshot
- The list of files, file versions, and other attributes of each file recorded by the snapshot
- The list of patches and bug fixes recorded by the snapshot

Table 10–1 describes the tables related to snapshots.

Table 10–1 Snapshot Tables

Table Name	Description
`AD_SNAPSHOTS`	Records one entry for each snapshot.
`AD_SNAPSHOT_BUGFIXES`	Records all patches ever applied at the moment of snapshot creation for a specific snapshot.
`AD_SNAPSHOT_FILES`	Records all files associated with a particular snapshot.

Table 10–1 (Cont.) Snapshot Tables

Table Name	Description
AD_FILES	Records all files associated with every snapshot in the system. Contains one entry for each file, regardless of how many snapshots and snapshot versions exist. This table may contain files that are no longer in the system to save files for past snapshots.
AD_FILE_VERSIONS	Records all versions for each file that are present in any snapshot.
AD_BUGS	Records all patches that have been applied to the Oracle home.

10.2.1 Maintain Snapshot Information Menu

You can select a task from the Maintain Snapshot Information submenu. Select the last option in the list to return to the AD Administration main menu. The following tasks are available:

- List Snapshots
- Update Current View Snapshot
- Create a Named View Snapshot
- Export a Snapshot to a File
- Import a Snapshot from a File
- Delete a Named View Snapshot

10.2.1.1 List Snapshots

This action queries the database and lists all snapshots. It displays the name, type, and comments for each snapshot stored in the AD_SNAPSHOTS table.

10.2.1.2 Update Current View Snapshot

If your current view snapshot is not up-to-date, you can update it. An example of when you may need to run this task is when a snapshot may not have been updated as the result of an interruption during a patching session. All of the other snapshot maintenance tasks are based on the results of this task. You can choose to scan the entire Oracle Fusion Applications Oracle home for all installed product families to determine which component directories should be examined or you can choose a specific product family. For each of those components, it scans these *product family* subdirectories: components, db, and patch. It then updates the database as follows:

- Inserts and updates the current view snapshot whenever the Update Current View Snapshot runs.
- Modifies the file and file version information for the current view snapshot.
- Sets the RAN_SNAPSHOT_FLAG column in the AD_SNAPSHOTS table to Y to indicate that the Update Current View Snapshot ran.

10.2.1.3 Create a Named View Snapshot

This action lists the current view snapshot and prompts you to provide a name. It then performs the following actions:

- Creates the named view snapshot.

- Creates a file information list for this snapshot by copying from the current view snapshot.
- Copies the bug fix list from the current view snapshot.

10.2.1.4 Export a Snapshot to a File

This action lists the snapshots available for export. Then it copies the snapshot file details and snapshot bug fix information to an output file that you specify at the prompt.

10.2.1.5 Import a Snapshot from a File

This action prompts you for the full path and name of the file you want to import. It then updates the snapshot tables with the imported snapshot details.

10.2.1.6 Delete a Named View Snapshot

This action lists the snapshots available for deletion. For the snapshot you select, it deletes the file details and snapshot bug fix entries from the snapshot tables.

10.3 Maintaining Applications Database Entities

Database entities are objects or data related to Oracle Fusion Applications. During standard system use, some database entities may become invalid, for example, through user error or after you apply a large patch. Oracle recommends that you verify the integrity of database entities as a regular maintenance procedure, or whenever the activity of your system indicates that database entities may need revalidation.

AD Administration contains several tasks designed to manage database entities. You access them from the Maintain Applications Database Entities submenu. Some tasks in this menu report on issues, or potential issues, and others actually resolve the issues.

The following tasks are available:

- Compiling Invalid Objects
- Running the Health Check
- Recreating Grants and Synonyms
- Maintaining Multi-lingual Tables

10.3.1 Compiling Invalid Objects

As you apply patches and make manual changes to the database, the number of invalid objects can increase. This process compiles invalid database objects in all schemas in the database. As soon as the compile process is complete, AD Administration generates an HTML report that contains information about any objects that remain invalid.

10.3.2 Running the Health Check

The health check process is a centralized set of validity checks that you can perform to ensure that the database is compatible with Oracle Fusion Applications. It validates the schemas and the database objects they contain to verify that they are valid.

The checks performed are as follows:

- Confirms the presence and validity of ADM_DDL in the FUSION_RUNTIME schema
- Identifies grants given to Oracle Fusion Applications and grants given by way of roles. Ensures that no grant is given with the admin option set to Yes. This task grants privileges to roles used by the FUSION_RUNTIME schema.
- Performs various checks on Oracle Fusion Application such as ensuring that synonyms point to valid, existing tables.
- Verifies that privileges are given to roles used by the FUSION_RUNTIME schema.
- Generates a list of invalid objects, compiles the invalid objects, and issues a report that includes the commands to recompile the objects that remain invalid.

The health check process produces a report with its findings, named ad_health_check_report.html. It also creates two scripts that contain a set of corrective actions that you can take to fix the issues found by the analysis. The scripts may contain steps for you to follow to resolve an issue and they may also contain SQL actions that are executed when you run the script. The HTML file and both scripts are located in the *FA_ORACLE_HOME* /admin/*SID*/log directory.

The following scripts are created:

- ad_health_fix_sys.sql: Run this script as sysdba.
- ad_health_fix_fus.sql: Run this script as fusion.

10.3.3 Recreating Grants and Synonyms

This task grants privileges to roles used by the FUSION_RUNTIME schema.

10.3.4 Maintaining Multi-lingual Tables

The standard Oracle Fusion Applications Multilingual Support (MLS) table architecture requires that the multilingual tables (those ending with _TL) be fully populated for all active languages. For each row in the base table, there must be a corresponding row in the MLS table for each active language. The Maintain Multilingual Tables utility standardizes and centralizes the synchronization logic for all tables in Oracle Fusion Applications. It relies on the Application/XDF dictionary data to know which tables to synchronize and inserts records only when they are missing for a given language. When you install a language, this task automatically runs. You can also run it from the AD Administration menu when you have reason to believe that the data in the _TL table is not synchronized. For more information, see "Installing and Maintaining Oracle Fusion Applications Languages" in the *Oracle Fusion Applications Administrator's Guide*.

10.4 Running Maintenance Tasks Noninteractively

The previous sections have so far described how to run AD Administration interactively - you start the adadmin utility and respond to system prompts, select a submenu from the main menu, and then select the task you want to run from that submenu. You can also run AD Administration noninteractively by specifying a defaults file that contains the information necessary to run one of the maintenance tasks. The first time you create the defaults file, you run the adadmin command interactively, and then you can reuse the defaults file for the task that you performed.

This section describes the steps necessary to create a defaults file and save it for when you want to run one of the AD Administration tasks without user intervention. You

may want to use the non-interactive functionality for long-running tasks, such as compiling the schema.

10.4.1 Creating a Defaults File

To create a defaults file, you add the option `defaultsfile=`*`defaults file location`* to your `adadmin` command. The following example syntax shows how to create a defaults file for compiling the schemas:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/adadmin.sh defaultsfile=APPLICATIONS_
CONFIG/atgpf/admin/default_compile.txt

(Windows) FA_ORACLE_HOME\lcm\ad\bin\adadmin.cmd defaultsfile=APPLICATIONS_
CONFIG\atgpf\admin\default_compile.txt
```

The next time you want to perform this same task non-interactively, call the `adadmin` command in non-interactive mode and specify the defaults file. Then run the `adadmin` command interactively to record the history of the session in the specified defaults file.

The following example specifies the defaults file:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/adadmin.sh defaultsfile=APPLICATIONS_
CONFIG/atgpf/admin/default_compile.txt interactive=no

(Windows) FA_ORACLE_HOME\lcm\ad\bin\adadmin.cmd defaultsfile=APPLICATIONS_
CONFIG\atgpf\admin\default_compile.txt interactive=no
```

When you use the defaults file in non-interactive mode, your `adadmin` session reads the contents of the defaults file to respond to the prompts.

10.4.2 Selecting a Menu Option Noninteractively

You can select a specific AD Administration menu option by specifying the menu name when you call the `adadmin` command. This functionality allows you to perform specific tasks somewhat noninteractively without using a defaults file. You must respond to the initial `adadmin` prompts. The following example shows how to run `adadmin` to compile the schemas:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/adadmin.sh menu_option=CMP_INVALID

(Windows) FA_ORACLE_HOME\lcm\ad\bin\adadmin.cmd menu_option=CMP_INVALID
```

If you select a menu option that contains additional prompts, such as List Snapshots, then you must respond to those prompts interactively. Table 10–2 provides a list of menu names that you can specify with `menu_option` when running the `adadmin` command.

Table 10–2 AD Administration Menu Options

Menu Name	Description
LIST-SNAPSHOT	Lists snapshots
UPDATE_CURRENT_VIEW	Updates the current view snapshot
UPDT_PROD_CURR_VIEW	Updates the current view snapshot for a product
CREATE_SNAPSHOT	Creates a named view snapshot
EXPORT_SNAPSHOT	Exports a snapshot to a file
IMPORT_SNAPSHOT	Imports a snapshot from a file

Table 10–2 (Cont.) AD Administration Menu Options

Menu Name	Description
DELETE_SNAPSHOT	Deletes a named view snapshot
CMP_INVALID	Compiles schemas
HEALTH_CHECK	Performs the health check analysis
CREATE_GRANTS	Re-creates grants and synonyms for the FUSION_RUNTIME schema
MAINTAIN_MLS	Maintains multi-lingual tables

10.4.3 Selecting a Menu Option While Using a Defaults File

You can combine the functionality provided by adding menu_option to the adadmin command along with specifying a defaults file. This allows you to bypass responding to prompts interactively. However, if you add the menu_option selection to the adadmin command while referencing an existing defaults file, your session executes only that menu choice from the defaults file. For example, if your defaults file was created to compile schemas and update the current view snapshot, the following command would update only the current view snapshot:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/adadmin.sh menu_option=update_current_view
defaultsfile=APPLICATIONS_CONFIG/atgpf/admin/mydefaults.txt

(Windows) FA_ORACLE_HOME\lcm\ad\bin\adadmin.cmd menu_option=update_current_view
defaultsfile=APPLICATIONS_CONFIG\atgpf\admin\mydefaults.txt
```

10.5 Running the HomeChecker Utility

The HomeChecker utility verifies the correctness of any Oracle Fusion Applications Oracle home directory. It verifies that the duplicated Oracle Applications Development Framework (Oracle ADF) libraries within Oracle Fusion Applications are synchronized with each other. While separate copies of the libraries are maintained, they are treated logically as a unit and this utility verifies that the libraries are still consistent with each other. HomeChecker also ensures that the artifacts within the Oracle home match the central Oracle Inventory, to confirm that the artifacts were laid down either as part of the initial installation or by a subsequent patch. For more information, see Section 2.1.5, "Oracle Universal Installer (OUI) Inventory".

The HomeChecker output displays the following information:

- Libraries that are not synchronized
- Oracle Fusion Applications artifacts that are not synchronized
- Libraries that are synchronized
- Oracle Fusion Applications artifacts that are synchronized

You would most likely run the HomeChecker utility when you are working with Oracle Support Services to diagnose issues with your environment.

To run the HomeChecker utility

Run the following command:

```
(UNIX) FA_ORACLE_HOME/lcm/ad/bin/homechecker.sh -validate FA_ORACLE_HOME path
[-logfile log file name] [-prodfamily comma-separated list of product families]
[-reportfile
```

```
report file path] [-loglevel log level]

(Windows) FA_ORACLE_HOME\lcm\ad\bin\homechecker.cmd -validate FA_ORACLE_HOME path
[-logfile log file name] [-prodfamily comma-separated list of product families]
[-reportfile
report file path] [-loglevel log level]
```

Table 10–3 HomeChecker utility parameters

Parameter	Description	Mandatory
validate	Specify the path to *FA_ORACLE_HOME*	Yes
logfile	Overrides the default log file name and sends the processing information to the file you specify, under the *APPLICATIONS_CONFIG*/lcm/logs/11.1.7.0.0/FAPMGR directory	No
prodfamily	Supply a comma-separated list of product families you want to see on the report	No
reportfile	Supply the directory path and file name for report output	No, the default is the current directory
loglevel	Records messages in the log file at the level you specify. Enter a value to override the default log level of INFO. See Section 11.1, "Oracle Fusion Applications Patch Manager Logging".	No

11

Monitoring and Troubleshooting Patches

This chapter describes how to monitor and troubleshoot Oracle Fusion Applications patching and AD Administration processing sessions.

This chapter contains the following topics:

11.1 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 you in diagnosing issues. When you use Patch Manager, you can specify the level of logging detail. The following log 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 that you should review.
- 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 you select a more detailed level of logging, the log files also include the lower level of details. For example, if you choose to see INFO messages in your log file, WARNING and SEVERE messages also appear in the log files. For more information, see "Standard Logging Levels" in the *Oracle Fusion Applications Administrator's Guide*.

11.1.1 Log Files for Single Patch Manager Sessions

You can examine the activities performed during patching sessions by reviewing the associated log files. Patch Manager consolidates log files for each patching session under the directory, *APPLICATIONS_CONFIG*/lcm/logs/11.1.7.0.0/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, you can view the Log Summary HTML file from a browser, which provides links to individual log files. You can periodically refresh the Log Summary HTML file to view the progress of the current patching session. If a task fails, you can access the links to the associated log files to assist in diagnosing the failure. For more information, see Section 11.2.1, "Log Summary".

When a patching session completes, its log files are archived in *FA_ORACLE_HOME*/admin/FUSION/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 *FA_ORACLE_HOME*/admin/FUSION/logarchive/fapmgr_command/*timestamp*.

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

Table 11–1 contains a list of log files created by Patch Manager during patching activities.

Table 11–1 Log Files for Single Oracle Fusion Applications Patch Manager Patching Activities

Log file or directory name under *APPLICATIONS_CONFIG*/lcm/logs/11.1.7.0.0/FAPMGR	**Log file generated from...**
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
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

Table 11–1 (Cont.) Log Files for Single Oracle Fusion Applications Patch Manager Patching Activities

Log file or directory name under *APPLICATIONS_CONFIG*/lcm/logs/11.1.7.0.0/FAPMGR	Log file generated from...
FAPMgrDiagnosticsSummary.html	The Diagnostic report when it is run on demand using this command syntax: `fapmgr.sh report -patchprogress`. For more information, see Section 3.5.5.1, "Running the Diagnostics Report".
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

11.1.2 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 master log file and diagnostics report for the overall multi-apply session. For more information, see Section 3.1.5, "Applying Multiple Patches Using a Patch Plan".

You can 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/11.1.7.0.0/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, you can view this log summary HTML file from a browser, which provides links to individual log files. You can periodically refresh the log summary HTML file to view the progress of the current patching session. If a task fails, you can access the links to the associated log files to assist in diagnosing the failure. For more information, see Section 11.2.1, "Log Summary".

Multi-apply sessions create a master 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.

Table 11–2 contains a list of log files created by Patch Manager during multi-apply patching activities.

Table 11–2 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	Master log file from a multi-apply patching session
FAPMgr_Multiapply_DiagnosticsSummary_*timestamp*.html	Master Diagnostics report from a multi-apply patching session in HTML format

Table 11–2 (Cont.) Log Files For Multi-apply Patch Sessions

Log file name or directory under FA_ORACLE_HOME/admin/FUSION/log	Log file description
FAPMgr_Multiapply_DiagnosticsSummary_*timestamp*.xml	Master 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

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
```

11.1.3 Timing Reports

For information about the duration of patching tasks, refer to the Diagnostics Report, in Section 11.2.2, "Diagnostics Report".

11.2 Monitoring Patching Sessions

Patch Manager coordinates patching activities by assigning tasks based on information from the patch metadata file. OPatch runs the tasks for updating middleware artifacts and AutoPatch runs the database tasks. Each of these tools generates one or more log files containing informational and error messages generated during patching. For more information, see Section 11.1.1, "Log Files for Single Patch Manager Sessions". If a task fails, the exact failure information for a given task is included in the log file. You can view the progress of the patching session from a browser, including the details of a failed task, by reviewing the Log Summary.

11.2.1 Log Summary

The Log Summary is created automatically whenever you start an Patch Manager session. 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 your browser and periodically refresh the page to see updated links to log files as they are created. You can open those links and monitor the progress of the session by refreshing the browser.

11.2.2 Diagnostics Report

After each patching sessions ends, the Diagnostics report is automatically generated so that you can view the results of the session from a browser. You can also 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 more information, see Section 3.5.5, "Diagnostics Report".

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

11.3 General Troubleshooting for Oracle Fusion Applications Patching

This section contains the following general troubleshooting scenarios for patching:

- Starting a New Patching Session After the Previous Session Failed

- Abandoning a Failed Patching Session
- Recovering from an Interrupted Patching Session
- Avoiding a Lost Connection During the Patching Session
- Resolving Components Locked by a Singleton Patch
- Resolving a Webcat Patch File Creation Failure
- Resolving an EditTimedOutException Error
- Finding Artifact Versions
- Backing Out Patches After They Have Been Successfully Applied

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 Section 4.18, "Patching Security Artifacts". For troubleshooting information that is specific to patching SOA composites, see Section 11.4, "Troubleshooting Patching Sessions for SOA Composites". For troubleshooting information that is specific to patching database content, see Section 11.5, "Troubleshooting Patching Sessions for Database Content".

11.3.1 Starting a New Patching Session After the Previous Session Failed

The previous patching session failed, and when you attempt to start a new patching session, a message appears about a previous session having failed.

The previous patching session could be in various states after its failure. The following scenarios are possible:

- To abandon the previous session and start a new session, see Section 11.3.2, "Abandoning a Failed Patching Session".
- In some cases, the patch tables do not correctly reflect the failed state of the patching session and may still show a patch task as running. In this case, you must use the `forcefail` command to fail the session. Then you can abandon the current patching session. For more information, see Section 11.3.3, "Recovering from an Interrupted Patching Session".
- 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 Section 7.4.6, "Abandoning a Failed Patching Session".

11.3.2 Abandoning a Failed Patching Session

The previous patching session failed and you want to start a new patching session. Only one patching session can be running at a time. If aborted, the patching session cannot be restarted and any pending patching actions, such as deployment actions, must be performed manually.

Always make sure that processes associated with the previous patching session do not exist. You can abandon a previously failed session by running the `fapmgr abort` command so that you can start a new patching session. 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. You cannot abandon a session that is actively running.

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

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

Table 11–3 describes the options available for the `abort` command.

Table 11–3 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 Section 11.1, "Oracle Fusion Applications Patch Manager Logging".	No, default value is INFO
help	Displays help.	No

If the `fapmgr abort` command errors with a message, such as, "Another APPLY session is already running", you may need to use the `fapmgr forcefail` command first. For more information, see Section 11.3.3, "Recovering from an Interrupted Patching Session". Also confirm that the table FND_INSTALL_PROCESSES does not exist.

11.3.3 Recovering from an Interrupted Patching Session

The patching session was interrupted by a system failure when Patch Manager and AutoPatch were running. The patching-related database tables still show the patching session as running, but no patching-related processes are actually running.

Use the `fapmgr forcefail` command to update the patching tables. Confirm that no patching processes are running before you use this command.

1. Confirm that all processes related to this patching session are no longer active. From your operating system, check for processes that are running `fapmgr`, `javaworker`, `adpatch`, `adadmin`, and `adworker`. If any processes are running, you must stop them using the command appropriate for your operating system.
2. Use the `fapmgr forcefail` command to update the patching tables.

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

   (Windows) FA_ORACLE_HOME\lcm\ad\bin\fapmgr.cmd forcefail [-logfile log file
   name] [-loglevel level]
   ```

3. From this point you can either abandon or restart the failed session.
   - To abandon the failed patching session, follow the steps in Section 11.3.2, "Abandoning a Failed 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.

Note: Patch Manager, Functional Setup Manager, and Applications Core all use AutoPatch for database patching. If a patching session is hung or incomplete, you may potentially need to consider the impact of an Applications Core or Functional Setup Manager patching session. Only one patching session can be active at a time.

11.3.4 Avoiding a Lost Connection During the Patching Session

If you initiate a patching session from a terminal server, such as a laptop, you may lose the connection 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.

11.3.5 Resolving Components Locked by a Singleton Patch

If you recently applied a one-off patch to a middleware artifact, you may encounter the following error the next time you apply a standard patch:

```
The incoming patch(es) [patch_number] target components that are locked by
singleton patch(es) [patch_number].
OPatch cannot proceed. Please refer to log file for more details.
```

A one-off middleware patch sets a lock on the component that the one-off patch updates. After you apply a one-off patch, you must subsequently apply the related standard patch that contains the same updates as the one-off to release the lock. The error message tells you which one-off patch is locking the component. For more information, see Section 2.2.1, "Impact of a One-off Patch".

Note: OPatch uses the term *singleton* patch, which is equivalent to a *one-off* patch, as used by Patch Manager.

11.3.6 Resolving a Webcat Patch File Creation Failure

If you apply a patch that contains BI Publisher artifacts, the BI Presentation servers should not be running. The following error occurs if the BI Presentation servers are running during the deployment of BI Publisher artifacts:

```
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. For more information, see "fastartstop Syntax" in the *Oracle Fusion Applications Administrator's Guide*.

11.3.7 Resolving an EditTimedOutException Error

If you receive the following error during patch validation,

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

you may have to manually release the edit session. This situation can occur when a domain is already in "edit" mode during patching, such as when the server crashes when Patch Manager tries to stop and restart it.

Follow these steps to manually release the edit session:

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.

11.3.8 Revert To a Previous Flexfield Definition After It Is Updated By a Patch

After you apply a patch that contains flexfield changes and you decide you are not ready to implement those flexfield changes, you have the option to 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. For your reference, the name of the label is included along with the Flex Deployment report in the patching log file. To use a previous version of a flexfield definition, use the WLST command `promoteMetadataLabel`. For more information, see "promoteMetadataLabel" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

To delete all previous MDS labels for a flexfield, after you confirm that you can use the changes delivered by a patch, use the WLST command `deleteFlexPatchingLabels`. Note that keeping old MDS labels adversely impacts performance. For more information, see "Using the WLST Flexfield Commands to Deploy Flexfields" in the *Oracle Fusion Applications Developer's Guide*.

11.3.9 Resolving 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, 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, you must ensure that the BI OPMN control process is up and running.

11.3.10 Finding 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 you are working with Oracle Support Services and you need to provide an artifact version.

Use this command syntax to generate the report:

```
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, you must 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/Ad
fPjfIntMspUi.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
```

For more information about OPatch, see the *Oracle Fusion Middleware Patching Guide*.

11.3.11 Backing Out Patches After They Have Been Successfully Applied

You should always test the application of a patch and the functionality related to the patch on a test system. After the patch is successfully tested, apply the patch on the production system. There is no automated method of backing out patches. Oracle strongly recommends that you work with Oracle Support Services if you must back out a patch.

11.4 Troubleshooting Patching Sessions for SOA Composites

The information provided in this section describes the troubleshooting process for errors that can occur when patching Service-Oriented Architecture (SOA) composites. These processes assume that you validate the patch and then apply the patch 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 you do not resolve validation errors before applying

the patch, the patch fails and you must manually deploy the SOA composites after you 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 Troubleshooting for SOA Composite Failures
- Troubleshooting SOA Composite Validation Failures
- Troubleshooting SOA Composite Deployment Failures
- Troubleshooting Complex Failures during SOA Patching

11.4.1 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 you validate or apply a patch that contains SOA composites, review the Diagnostics report for errors.

   Patch Manager generates the Diagnostics report after every validation and patching session. 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 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, you can review log files by accessing the link to the Log Summary. For more information, see Section 11.2.1, "Log Summary".
- 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. For more information, see "Troubleshooting Oracle SOA Suite" in the *Oracle Fusion Applications Administrator's Troubleshooting Guide*.
5. Determine if you need to manually restore the system back 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 condition in this message:

     ```
     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 you need to restore the system, follow the steps in Step 6.
6. If a failed deployment leaves a composite in an inconsistent state, you must restore the system to its original state. Follow these steps to use WLST commands to restore the system. If you prefer to use Oracle Fusion Applications Control to restore the system, see Step 7.
   a. Find out what the active revision of the composite was 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 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:

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

To assign the default composite:

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

7. Follow these steps to restore the system to its original state using Oracle Fusion Applications Control. For more information, see "Accessing Fusion Applications Control" in the *Oracle Fusion Applications Administrator's Guide*.

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 example, 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. For more information, see "Undeploying SOA Composite Applications" in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

c. Mark the previously active revision of the composite as active and as a default revision. For more information, see "Managing the State of All Applications at the SOA Infrastructure Level" in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

8. Manually deploy the SOA composites included in the patch by following the steps in Section 4.19.2, "Manually Deploying SOA Composites".

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 "Table 10-2 SOA Log Information for Oracle Support Services" in the *Oracle Fusion Applications Administrator's Troubleshooting Guide*.

11.4.2 Troubleshooting SOA Composite Validation Failures

This section describes common problems and solutions for SOA composite validation failures. Errors that occur during the validation phase must be resolved before applying the patch. If you encounter these errors during patch application, you must manually deploy the SOA composites after you 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:

- Oracle JDeveloper Customization Error
- SOA Server Not Available
- Administration Server Not Available
- SOA-Infra Server Is Ready

11.4.2.1 Oracle JDeveloper Customization Error

An error that is related to a JDeveloper customizations occurs when you customized a SOA composite and did not save the customizations. You must save these customizations before you apply a patch that includes the next revision of the composite. Follow the steps in Section 4.19.1, "Preserving SOA Composite JDeveloper Customizations Before Applying a Patch" to resolve this error.

11.4.2.2 SOA Server Not Available

If the SOA server is down or not available for patching, patch validation succeeds, but you receive 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 that you wait until the SOA server is completely up and running, with all composites initialized.

For more information, see "SOA Server Does Not Start" in the *Oracle Fusion Applications Administrator's Troubleshooting Guide* and "Accessing Fusion Applications Control" in the *Oracle Fusion Applications Administrator's Guide*.

11.4.2.3 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 "Accessing Fusion Applications Control" in the *Oracle Fusion Applications Administrator's Guide*.

11.4.2.4 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. If you are using a cluster, you must perform this check for all SOA-infra servers in the cluster. For more information, see "Accessing Fusion Applications Control" in the *Oracle Fusion Applications Administrator's Guide*.

11.4.2.5 Composite with Identical Revision Is Already Deployed

If you 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.

You can resolve this error either by not applying the current patch or by undeploying the composite before applying the patch. Note that if you undeploy the composite, you lose any customizations you may have made to that composite. 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*.

11.4.3 Troubleshooting SOA Composite Deployment Failures

This section describes common problems and solutions for SOA composite deployment failures during patching. 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 you resolve the cause of the error, you must deploy the composite manually.

This section contains the following topics:

- Failed to Make New Composite Revision the Default
- Failed to Retire Previous Composite Revision
- Custom Metadata and Key Flexfield Changes Are Not Propagated Across Clusters

11.4.3.1 Failed to Make New Composite Revision the Default

If you receive an error in making the new composite the default, you can 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. If it is not, you must then manually deploy the composite that failed, by following the steps in Section 4.19.2, "Manually Deploying SOA Composites".

For more information, see "sca_assignDefaultComposite" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

11.4.3.2 Failed to Retire Previous Composite Revision

If you 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, you must then manually deploy the composite that failed, by following the steps in Section 4.19.2, "Manually Deploying SOA Composites".

For more information, see "sca_retireComposite" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

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

Custom metadata and key flexfield changes are not propagated across clusters after applying a patch.

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. To analyze,

export, and import the metadata, follow the steps in "Task: Synchronizing Customized Flexfields in the MDS Repository for SOA" in the *Oracle Fusion Applications Extensibility Guide for Developers*.

11.4.4 Troubleshooting Complex Failures during SOA Patching

The following failures may require contacting Oracle Support Services. If you are unable to resolve the failure after following the steps in Section 11.4.1, "Basic Troubleshooting for SOA Composite Failures", file a service request with Oracle Support Services and provide the logs and information as described in "SOA Log Information for Oracle Support Services" in the *Oracle Fusion Applications Administrator's Guide*.

11.4.4.1 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.
```

11.4.4.2 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.
```

11.4.4.3 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.
```

11.5 Troubleshooting 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. With AD Controller, you can determine the status of the workers and control their actions. For more information about workers, see Section 3.1.2.1, "Worker Processes".

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

- Starting AD Controller
- Reviewing Worker Status
- Determining Why a Worker Failed
- Restarting a Failed Worker
- Terminating a Hung Worker Process
- Shutting Down the Manager
- Reactivating the Manager

- Resolving the Error, "Unable to start universal connection pool"
- Resolving a Worker Blocked by a Session
- Resolving an Error During Conflict Checking
- Resolving an Error During Upload of Flexfield Data
- Setting the Environment for Troubleshooting Database Issues

11.5.1 Starting AD Controller

Follow these steps to start AD Controller:

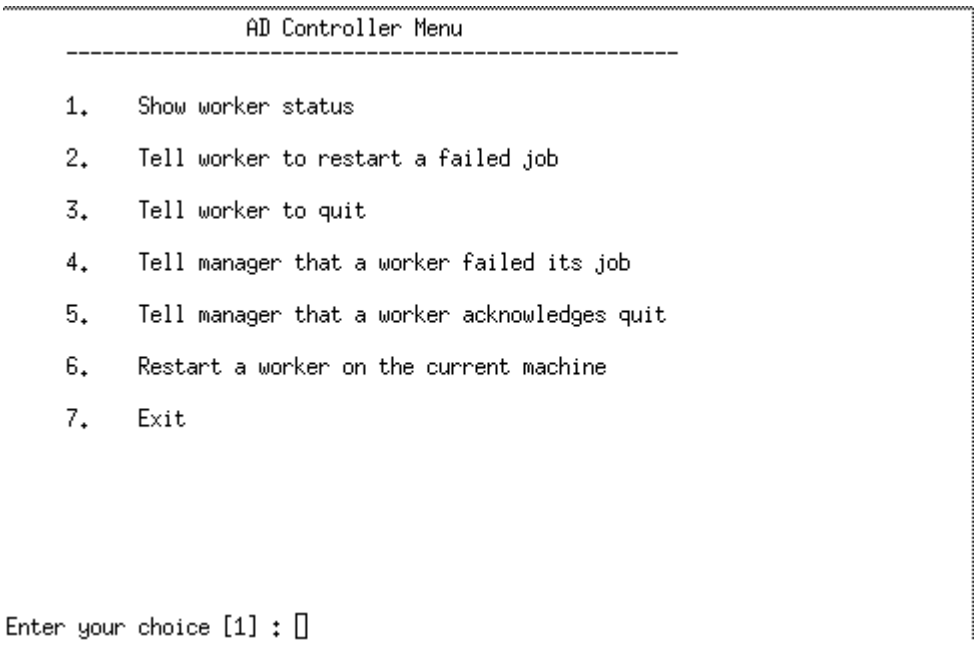
1. Start AD Controller with the `adctrl` command.

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

   It prompts you to:

   - 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. You can press Enter at any time to return to the AD Controller main menu.



11.5.2 Reviewing 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. You can use AD Controller to determine the status of workers and manage worker actions. You can also find the status of workers by reviewing the Log Summary. For more information, see Section 11.2.1, "Log Summary".

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

1. Start AD Controller. For more information, see Section 11.5.1, "Starting AD Controller".
2. Review worker status.

   Select **Show worker status** from the AD Controller main menu. AD Controller displays a summary of current worker activity. 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

     Table 11–4 describes the status that may be assigned to a worker.

Table 11–4 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	You fixed 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 Section 11.5.3, "Determining Why a Worker Failed" for assistance in fixing the problem so Patch Manager can complete its processing.

11.5.3 Determining Why a Worker Failed

When a worker fails its task, you do not have to wait until the other workers and the manager stop. You can review the worker log files to determine what caused the failure. Workers do not proceed to run tasks in the subsequent phase until all tasks in the current phase complete successfully. You must take action 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 from you.

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 by you 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 Section 11.2.1, "Log Summary".
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 Section 11.1.1, "Log Files for Single Patch Manager Sessions".
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 you resolve the problem that caused the failure, restart the failed worker.

   Select **Tell worker to restart a failed job** from the AD Controller main menu. For more information, see Section 11.5.4, "Restarting a Failed Worker".
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.

11.5.4 Restarting a Failed Worker

If a worker job failed or if you terminated a hanging worker process, you must manually restart the worker. Some worker processes spawn other processes called child processes. If you terminate a child process that is hung, the worker that spawned the process shows *Failed* as the status. After you fix the problem, you must restart the failed worker. After the worker restarts, the associated child processes start as well.

Follow these steps to restart a failed worker:

1. Start AD Controller. For more information, see Section 11.5.1, "Starting AD Controller".
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 Section 11.5.5, "Terminating a Hung Worker Process".

11.5.5 Terminating a Hung Worker Process

When running AD utilities, there may be situations when a worker process appears to hang, or stop processing. If this occurs, you may need to terminate the process manually. After you do that, you must also restart that process manually.

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

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

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

1. Start AD Controller. For more information, see Section 11.5.1, "Starting AD Controller".
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. You can also verify whether the process is consuming CPU resources from your operating system.

   For more information, see Section 11.1.1, "Log Files for Single Patch Manager Sessions".
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.

   ```
   (UNIX) ps -a| grep worker number
   (Windows) Start the Task Manager (Ctrl-Alt-Delete) to view processes.
   ```
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 you terminate a SQL*Plus session spawned by a worker, you just need to kill the SQL*Plus session. The worker immediately detects the FAILED state. For other processes, follow Steps 7 through 11.

   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.
10. Fix the issue that caused the worker to hang. Search the My Oracle Support site or file a service request with Oracle Support Services if you do not know how to fix the issue.
11. 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.

Note: Do not select **Restart a worker on the current machine** if the worker process is running. Doing so creates duplicate worker processes with the same worker ID and will cause incorrect results.

11.5.6 Shutting Down the Manager

There may be situations when you must shut down an AD utility while it is running. For example, you may want to shut down the database while Oracle Fusion Applications is running AutoPatch or during an AD Administration session. You should perform this shutdown in an orderly fashion so that it does not affect your data. The best way to do this is to shut down the workers manually.

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.

11.5.7 Reactivating the Manager

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 Section 11.5.1, "Starting AD Controller".
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 if you do not understand how to 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.

11.5.8 Resolving 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" in the *Oracle Fusion Applications Performance and Tuning Guide*.

11.5.9 Resolving 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. You can use the following SQL*Plus query to identify the sessions that are blocking patching sessions:

```
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>';
```

> **Note:** To minimize blocked sessions, ensure that you follow the steps in Step 7, "Prepare the System" before you apply a patch.

11.5.10 Resolving an Error During Conflict Checking

If you recently applied a one-off patch to a database artifact, you may encounter the following error the next time you apply a standard patch:

```
Error occurred during patch conflicts checking.
This may be due to infrastructure failure OR patch conflicts
You may check FA_ORACLE_HOME/admin/FUSION/out/patch_number_conflict_report.xml
file for any patch conflicts
You should check the file FA_ORACLE_HOME/admin/FUSION/log/adpatch.log for errors
```

A one-off database patch sets a lock on the artifact that the one-off patch updates. After you apply a one-off patch, you must subsequently apply the related standard patch that contains the same updates as the one-off to release the lock. To find out which one-off patch is locking the artifact, review the `conflict_report.xml` file. In the following example of this file, the one-off patch that created the lock was 909090.

```
<Patch_Conflict_Report>
  <instance>
    <appl_sys_name>FUSION</appl_sys_name>
    <appl_top>/server01/fusionapps</appl_top>
    <patch_number>909090</patch_number>
  </instance>
  <patch_type>one-off</patch_type>
  <conflict_details>
    <conflicts>
```

```
      <prod>HCM</prod>

<subdir>hcm/components/hcmPayroll/legconfig/setup/dbSchema/database/fusionDB/FUSIO
N</subdir>
      <filename>PAY_INSTALLED_LEGISLATIONS.table</filename>
      <bug>909090</bug>
    </conflicts>
```

To resolve this issue and remove the lock, obtain and apply the standard patch that delivers the same fix as the one-off patch.

11.5.11 Resolving 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. You can ignore these errors if the flexfield seed data task succeeds on the retry. If the task remains in a failed state, you must use the AD Controller utility to retry the failed task.

For more information, see Section 11.5.4, "Restarting a Failed Worker".

11.5.12 Setting the Environment for Troubleshooting Database Issues

If you need to connect to the Oracle Fusion Applications database to troubleshoot database related issues, by running SQL*Plus, for example, you need to set up the appropriate environment variables. 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

(Windows, TWO_TASK is known as LOCAL)
adsetenv.cmd
APPSORA.cmd
echo %LOCAL%
```