

## Using the Online Patching Readiness Report in Oracle E-Business Suite Release 12.2 (Doc ID 1531121.1)

This document introduces the Global Standards Compliance Checker (GSCC) and Readiness Report, and outlines how it is used to help prepare for using Oracle E-Business Suite Release 12.2 with online patching, as described in the applicable version of *Oracle E-Business Suite Upgrade Guide*, available in [Oracle E-Business Suite Documentation Web Library Release 12.2](#).

This document contains the following sections:

- [Section 1: Prerequisites for Oracle E-Business Suite Online Patching Readiness and GSCC Report Patch](#)
- [Section 2: Using the Global Standards Compliance Checker \(GSCC\) and Readiness Report](#)
- [Section 3: Check for Data Dictionary Corruption / Time Stamp Mismatch](#)
- [Section 4: Run Script to Reduce Upgrade Time](#)
- [Section 5: Patch Readme](#)
- [References](#)

There is a [change log](#) at the end of this document.

### Section 1: Prerequisites for Oracle E-Business Suite Online Patching Readiness and GSCC Report Patch

You need to start by identifying and applying the correct Online Patching Readiness and GSCC Report patch for the codelevel from which you are upgrading.

Apply whichever of the following patches is relevant to your Oracle E-Business Suite release.

Oracle E-Business Suite Release	Online Patching Readiness and GSCC Report Patch
12.2.3 or later	<a href="#">Patch 33993716:R12.AD.C</a>
12.1	<a href="#">Patch 31026891:R12.AD.B</a>
12.0	<a href="#">Patch 31026891:R12.ADA</a>
11i	<a href="#">Patch 31026891</a>

The patch delivers a new set of Online Patching Development Standards for Oracle E-Business Suite Release 12.2, plus a set of reports that will help you analyze the dependencies between database objects as specified by the new rules used by the Edition Based Redefinition (EBR) feature that was introduced in Oracle Database Release 11gR2.

You can then proceed to use the Global Standards Compliance Checker (GSCC) and Readiness Report, as described in Section 2 below.

### Section 2: Using the Global Standards Compliance Checker (GSCC) and Readiness Report

This patch delivers scripts to automatically check that custom code complies with Oracle E-Business Suite online-patching coding standards. The implementation of online patching in Oracle E-Business Suite Release 12.2 relies on the Oracle Database 11gR2 EBR feature, and adds a new logical view over the database objects in Oracle E-Business Suite. Access to these database objects must be via the logical layer, and new coding standards help to ensure that code does this correctly. The implementation of the logical layer is such that the majority of application code already follows the new standards: however, this patch delivers scripts to scan for and identify many compliance issues that may exist.

The Global Standards Compliance Checker (GSCC) delivered in this patch consists of the main engine script, `$FND_TOP/bin/gsccl.pl`, plus a variety of enforcement code (EFC) modules in `$FND_TOP/perl/GSCC/OpenEFC` that check for common standards issues.

### Section 3: Check for Data Dictionary Corruption / Time Stamp Mismatch

If it is not already applied, apply the correct Online Patching Readiness and GSCC Report Patch for the codelevel from which you are upgrading. (Refer to [Section 1](#) and [Section 2](#) of this document.)

#### 3.1 Check for Data Dictionary Corruption

- Run the `$AD_TOP/patch/115/sql/adzddtsfix.sql` script, following the usage instructions provided.
- Run the `$AD_TOP/sql/ADZDDDBCC.sql` script to identify whether logical data dictionary corruption is present.
  - If no corruption is found, proceed with the upgrade.
  - If corruption is present, follow Section 3.2 below.

#### 3.2 Fix Data Dictionary Corruption (Conditional)

##### 3.2.1 Logical Data Dictionary Corruption (Missing Parent)

**Note:** Follow these steps **only** as a fix when logical data dictionary corruption (missing parent) is present.

- Copy the script `$AD_TOP/patch/115/sql/adzddmpfix.sql` to the database tier.
- On the database tier, connect to the database as `sysdba`.

**Note:** If you are on Oracle E-Business Suite Release 12.1.3 or Release 12.2 with a multitenant architecture, to connect to a multitenant architecture database as `sysdba`, run the following commands to set the database SID field to `<EBS_PDB_Name>`, the pluggable database name for your Oracle E-Business Suite database:

```
$ source <CDB_NAME>.<NODE_NAME>.env
$ export ORACLE_PDB_SID=<EBS_PDB_Name>;
$ sqlplus "/ as sysdba"
```

- Run the `$AD_TOP/patch/115/sql/adzddmpfix.sql` script.
- On the application tier, run the `$AD_TOP/sql/ADZDDDBCC.sql` script again to identify whether data dictionary corruption is still present.
  - If no corruption is found, proceed with the upgrade or patching cycle.
  - If corruption is still present, proceed to Section 3.3 below.

##### 3.2.2 Logical Data Dictionary Corruption (Time Stamp Mismatch)

**Note:** Follow these steps **only** as a workaround when logical data dictionary corruption (time stamp mismatch) is reported for the following Java classes in SYS on a RAC environment when adop is run.

```
com/sun/activation/viewers/ImageViewer
com/sun/activation/viewers/ImageViewerCanvas
```

Use the following steps to resolve the issue:

- Shut down all application tier services.
- On the database tier, connect to the database as `sysdba`.

**Note:** If you are on Oracle E-Business Suite Release 12.1.3 or Release 12.2 with a multitenant architecture, to connect to a multitenant architecture database as `sysdba`, run the following commands to set the database SID field to `<EBS_PDB_Name>`, the pluggable database name for your Oracle E-Business Suite database:

```
$ source <CDB_NAME>.<NODE_NAME>.env
$ export ORACLE_PDB_SID=<EBS_PDB_Name>;
$ sqlplus "/ as sysdba"
```

```
SQL> alter session set "_ORACLE_SCRIPT"=true;
SQL> CREATE OR REPLACE JAVA SYSTEM;
SQL> alter session set "_ORACLE_SCRIPT"=false;
```

- Start up all application tier services.
- On the application tier, run the `$AD_TOP/sql/ADZDDDBCC.sql` script again to identify whether data dictionary corruption is still present.
  - If no corruption is found, proceed with the upgrade or patching cycle.
  - If corruption is still present, proceed to Section 3.3 below.

#### 3.3 Additional Actions for Data Dictionary Corruption (Conditional)

**Note:** Follow these steps **only** when data dictionary corruption is present after following the steps in 3.2 above.

- On the database tier, go to the `$ORACLE_HOME/rdbms/admin` directory.
- Run the `utlirp.sql` script, following the usage instructions provided.
- Run the `utlirp.sql` script, following the usage instructions provided.

**Was this document helpful?**

Yes  
 No

---

**Document Details**

Type: ANNOUNCEMENT  
 Status: PUBLISHED  
 Last Major Update: 18-Nov-2022  
 Last Update: 06-Jun-2025  
 Language: English

---

**Related Products**

Oracle Applications DBA  
 Oracle Applications Utilities  
 Oracle E-Business Suite Technology Stack

---

**Information Centers**

Oracle Catalog: Information Centers and Advisors for All Products and Services [50.2]  
 Privacy and Security Feature Guidance for all Oracle Products (On Premise) [113.2]

---

**Document References**

No References available for this document.

---

**Recently Viewed**

Guidance for Integrating Custom and Third-Party Products With Oracle E-Business Suite Release 12.2 [1916149.1]

4. On the application tier, run the \$AD\_TOP/sql/ADZDDBCC.sql script again to identify whether data corruption is still present.
- If no corruption is found, proceed with the upgrade or patching cycle.
  - If corruption is still present, contact Oracle Support and request a bug to be logged.

#### Section 4: Run Script to Reduce Upgrade Time

To reduce the overall upgrade time, you can optionally run the ADZDMVREFRESHNEEDED.SQL script before starting the 12.2.0 upgrade driver.

#### Section 5: Patch Readme

The contents of the Readme (included with the patch) are as follows:

```
-----
RI222: STANDALONE READINESS REPORT AND GLOBAL STANDARDS COMPLIANCE CHECKER (GSCC)
-----

Update - 33993716
Product - Applications DBA
Release - R12
Platform - Generic Platform
Built - NOV-18-2022 07:32:47

Instructions For Applying This Patch
-----

Preparation Tasks
-----
The tasks in this section can be completed without taking any Applications
services or users offline.

There are no tasks to be performed in this section.

Pre-install Tasks
-----
For 12.0.X / 12.1.X / pre-upgrade patches (using adpatch), you must
shut down all Application tier services before performing the tasks in this
section.
For 12.2.X patches (using adop), you can perform the tasks in this
section without shutting down the Application tier services.

There are no tasks to be performed in this section.

Apply The Patch
-----

1. Apply patch
This patch contains the following unified driver file to be applied with
ADOP (AD Online patch):
u33993716.drv (This is the unified driver)

Post-install Tasks
-----
For 12.0.X / 12.1.X patches, you must complete the tasks in this
section before starting up Application tier services.
For 12.2.X patches, you may complete the tasks in this section at
any time after the update, without taking any services or users offline.

There are no tasks to be performed in this section.

Finishing Tasks
-----
You may complete the tasks in this section at any time after the update,
without taking any services or users offline.

There are no tasks to be performed in this section.

Additional Information
-----

There are no tasks to be performed in this section.

Description
-----

Oracle E-Business Suite Online Patching Readiness and GSCC Report Patch
-----
The Oracle E-Business Suite Release 12.2 includes a capability to apply
patches online. This feature relies on the Edition Based Redefinition
feature of the Oracle Database introduced in Release 11gR2. This database
feature added some new rules for implementing dependencies between database
objects. The purpose of this patch is to deliver a set of reports that will
help you analyze these dependencies as well as a new set of Online Patching
Development standards.

ADZDPSUM.sql
Lists schemas containing objects that reference EBS objects that are not
editioned. If the schemas are registered with the application, the Online
Patching Enablement patch will enable editioning on those schemas avoiding
many EBR rule violations (Non-Editioned objects referencing Editioned
objects). This report also provides instructions on how to register your
custom schemas, you are advised to re-run the script after registering your
schemas in order to reduce the number of violations in the next report.

ADZDFMAN.sql
Lists objects that violate EBR standards and that require manual intervention
in order to address the violations. It groups the objects per violation type
and provides instructions on how to address those violations. Make sure you
address all the violations in this report before proceeding with the next
reports.

ADZDPAUT.sql
Lists objects that violate Online Patching Enablement standards. The objects
in this report will be adjusted automatically by the Online Patching
Enablement patch. This report is provided for information purposes only, no
action is required on the output of this report.

ADZDDBCC.sql
Lists objects that violate Online Patching Development standards and that
require manual intervention in order to address the violations. It groups
the objects per violation type and provides instructions on how to address
those violations.

Apply patch
-----
1. Apply E-Business Suite:Online Patching Readiness and GSCC Report patch
applicable to your EBS instance's release version by referring to patches
listed in the Section 1: Prerequisites for Oracle E-Business Suite Online
Patching Readiness and GSCC Report Patch.

2. Once it is applied, Auto-patch will copy SQL files from patch to
$AD_TOP/sql/ADZDPSUM.sql
$AD_TOP/sql/ADZDPAUT.sql
$AD_TOP/sql/ADZDFMAN.sql
$AD_TOP/sql/ADZDDBCC.sql
$AD_TOP/sql/ADZDPCUST.sql location.

STEP#1: Run $AD_TOP/sql/ADZDPSUM.sql
-----
This script should be run by SYSTEM schema of ORACLE database.
For Example:

sqlplus system/<system password>@DB @$AD_TOP/sql/ADZDPSUM.sql

OR

sqlplus system@DB @$AD_TOP/sql/ADZDPSUM.sql

In latter case, sql script will prompt for SYSTEM schema password. The
execution-time of this script varies from machine to machine and may take
approximately 2-3 hours to complete.

Note:
The SQL script only queries the data dictionary, it does not query any
customer tables, or perform any data dictionary updates.

Make sure you review the instructions in the report, perform the
recommended actions and re-run the report until you have no more pending
actions.

STEP#2: and STEP#3:
-----
In the same way as STEP#1 has been performed for @$AD_TOP/sql/ADZDPSUM.sql,
it should be repeated for
@$AD_TOP/sql/ADZDFMAN.sql and
@$AD_TOP/sql/ADZDPAUT.sql also

STEP#4: Run ADZDDBCC.sql
-----
This script should be run by APPS schema of E-business Suite.

For Example:

sqlplus APPS/<apps-password>@DB @$AD_TOP/sql/ADZDDBCC.sql
```

NOTE: Currently these scripts share common objects names, hence to be run sequentially.

#### Review generated report

- The ADDPSUM.sql SQL script generates report file addpsum.txt. Each section of this report should be reviewed carefully and take appropriate actions.

- Review addpman.txt for detail and follow given instructions and may review addpaut.txt (optional) also but these violations would be fixed automatically when your instance is enabled for Online Patching.

- Review database compliance-checker report adddbcc.lst

#### Global Standards Compliance Checker (GSCC)

This patch delivers scripts to automatically check that custom code complies with E-Business Suite (EBS) online-patching coding standards.

The implementation of Online Patching in Oracle E-Business Suite Release 12.2 relies on the Oracle Database 11gR2 Editioning feature and adds a new logical view over the database objects in Oracle E-Business Suite. Attempts to access these database objects must go through the logical layer and new coding standards help to insure that code does this correctly. The implementation of the logical layer has been done such that the majority of application code already follows the new standards; however, this patch delivers scripts to scan and identify many compliance issues if they exist.

The Global Standards Compliance Checker (GSCC) delivered in this patch consists of the main engine script \$FND\_TOP/bin/gsccl.pl and a variety of enforcement code (EFC) modules in \$FND\_TOP/perl/GSCC/OpenEFC that check for common standards issues.

The following table lists the messages that can appear in the output of the GSCC script along with a reference to the applicable Development Standards sections which are included at the end of this file.

#### GSCC Error Associated Coding Standard Standards Section

File.Sql.80 VPD Policy must be on the Editioning View or VPD

Table Synonym, not the table.

File.Gen.34 Table Name must not use '#' character. Table (Ordinary)

File.Gen.35 Table must be owned by an EBS product schema Table (Ordinary)

and not APFS.

File.Gen.36 Base Column Name may only use '#' as the last Table (Ordinary)

character.

File.Xdf.4 Base Column name should be 28 bytes or less. Table (Ordinary)

File.Gen.37 Column Type must be a built-in type or a Table (Ordinary)

user-defined type owned by a non-editioned user.

File.Xdf.4 Column Type must not be LONG or LONG RAW. Table (Ordinary)

File.Xdf.4 Column Type should not be ROWID Table (Ordinary)

File.Gen.41 Query/DML statements must access Tables via the Table (Ordinary)

Table Synonym or Editioning View.

File.Gen.38 Do not modify application-managed tables in an Table (Ordinary)

Online Patch.

File.Sql.81 Patch the table definition using ODF or XDF. Table (Ordinary)

File.Gen.39 Index Name must contain an underscore ('\_'). Index

File.Sql.81 Deliver constraint definition using ODF or XDF. Integrity Constr.

File.Sql.82 Deliver Materialized View using XDF. Materialized View

#### Using GSCC

The Global Standards Compliance Checker (GSCC) delivered in this patch consists of the main, engine script \$FND\_TOP/bin/gsccl.pl and a variety of standards enforcement code (EFC) modules in \$FND\_TOP/perl/GSCC/OpenEFC/ that check for common standards issues.

After applying this patch to install the GSCC code, source the applications environment file and then run the GSCC on a set files under a directory tree like this:

```
cd /home/yourdir
$FND_TOP/bin/gsccl.pl -f '/home/yourdir/your-code/'
```

In this example, gsccl.pl will check all of the files located under the your-code/ directory using the EFC modules located in \$FND\_TOP/perl/GSCC/OpenEFC/ and generate a report named gsccl-out.log in the current working directory (/home/yourdir/ in this example). Invoking gsccl.pl without arguments will print a usage message with additional information.

#### Reading GSCC Output

GSCC output looks like this example (sans line numbers):

```
01: -----
02: GSCC Compliance Test Results:
03: -----
04: File: /home/yourdir/your-code/yourfile.xdf
05:
06: Errors (3):
07: File.Gen.34 (Code Reviewers: CODE_REVIEWER)
08: 341 - Table Name must not use '#' character.
09: File.Gen.35 (Code Reviewers: CODE_REVIEWER)
10: 340 - Table must be owned by an EBS product schema, not APFS.
11: File.Gen.36 (Code Reviewers: CODE_REVIEWER)
12: 387, 443, 518 - Base Column Name may only use '#' as last character
13:
14: -----
15: GSCC Totals:
16: Passes: 19
17: Fails: 3
18: Warnings: 0
19: Under Review: 0
20: GSCC Errors: 0
21:
22: --* Failed GSCC Compliance Test with Code Review Failures *--
```

Lines 07:, 09:, and 11: in the example report name the coding standards that GSCC identified as potential problems in the file being checked (line 04:, yourfile.xdf in this example).

Lines 08:, 10:, and 12: in the report then give the specific lines within yourfile.xdf (lines 341, 340, 387, 443, and 518) where potential standards compliance problems were found.

To correct the errors shown in this example, the table being created on line 341 of yourfile.xdf would need to be renamed to remove the '#' character from the table name; the table would need to be created in an EBS schema other than APFS; and the table columns created on lines 387, 443, and 518 of yourfile.xdf would need to be renamed to remove the '#' character from their names.

#### Sections of the Database Object Development Standards

The following sections of the Database Object Development Standards have been checked by automated GSCC standards.

#### Virtual Private Database (VPD) Policy

##### VPD Definition Standards:

VPD Policy must be on the Editioning View or Table Synonym, not the table.

VPD Dynamic DDL Standards: No special considerations

VPD Online Patching Compliance Standards: No special considerations.

VPD Tip: You can add or drop VPD policies by calling the FND\_ACCESS\_CONTROL\_UTIL package from a SQL script (4phase-plb).

For more information on Oracle Virtual Private Database, see: Oracle Database Security Guide 11g Release 2 (11.2).

#### Table (Ordinary)

An Ordinary Table is created/alterd/dropped during application patching, in contrast, a Dynamic Table is created/alterd/dropped by application runtime. The standards in this section only apply to Ordinary Tables.

In order to implement effectively-editioned support for ordinary tables, the online patching technology installs and maintains a new Editioning View Layer over each table. The editioning view maps logical column names used by the application to the actual storage columns used to hold those attributes in each edition. Developers must follow new procedures in order to create and alter ordinary tables in a development database, and patch those changes to other systems.

#### Table Definition Standards:

Ordinary Table is a table that is used by Oracle E-Business Suite runtime, and patched via Online Patching

Table Name must not use '#' character. (GSCC File.Gen.34)

Table Name must be unique within the first 29 bytes.

Table must be owned by an Oracle E-Business Suite product schema or custom product schema, not APPS. (GSCC File.Gen.35)

Base Column Name may only use '#' as last character. (GSCC File.Gen.36)

Base Column Name should be 28 bytes or fewer. (GSCC File.Xdf.4)

Note: Online Patching currently does not support patching columns that violate this standard. If a violating column must be patched, then it must be replaced with compliant column (shorter name) as part of the patch. Application code will need to be updated to reference the new shorter column name.

Column Type must be a built-in type or a user-defined type owned by a non-editioned user. (GSCC File.Gen.37)

Column Type must not be LONG or LONG RAW. For more information on this standard, see: Oracle Database SecureFiles and Large Objects Developer's Guide 11g Release 2 (11.2). Also see: LONG to CLOB Conversion Procedures. (GSCC File.Xdf.4)

Column Type should not be ROWID. (GSCC File.Xdf.4)

ROWID references can become invalid if the target table is patched, loaded, or rebuilt. It is not safe to store ROWID references across an online patching boundary.

Table Usage Standards:

Query/DML statements must access Tables via the Table Synonym or Editioning View. (GSCC File.Gen.41)

If you query, display, or store table column names in your application runtime code, you should use Logical Column Names rather than Physical Column Names in most cases.

Follow the Logical versus Physical Column Guidelines for runtime application code.

Warning: Some dictionary views (such as ALL\_TAB\_COLUMNS) contain information for both Logical and Physical table columns, depending on whether you query the editioning view or base table data. Consult Logical versus Physical Column Guidelines for details.

Review the section Oracle E-Business Suite Objects that Reference ALL\_TAB\_COLUMNS or DBA\_TAB\_COLUMNS and ensure that your query is getting the correct logical or physical column information for your purposes.

DDL statements such as TRUNCATE will not work on an APPS table synonym that points to an editioning view. To truncate a table, you must supply the actual base table (owner.table\_name) in the truncate command.

Table Dynamic DDL Standards:

Application-managed tables are tables that are created and maintained by application logic during normal application runtime:

Application-managed (dynamic) tables must not have an Editioning View.

Do not modify application-managed tables in an Online Patch. (GSCC File.Gen.38)

Ordinary tables are created and maintained by Online Patching (and will have an editioning view)

If the application logic modifies an ordinary table at runtime, it must use the AD\_DDL interface to execute the dynamic DDL.

Do not modify ordinary tables in the Run Edition while a Patch Edition exists

Table Online Patching Compliance Standards:

Patch the table definition using ODF or XDF. (GSCC File.Sql.81)

(Automatic) The table will be covered by an editioning view of the name: substr(table\_name, 1, 29)||'#'

(Automatic) The editioning view will map each logical column name to the latest version of that column.

Example: Table "FND\_PROFILE\_OPTIONS" has an editioning view called "FND\_PROFILE\_OPTIONS#"

Do not update existing columns in a way that is incompatible with the running application. Instead, create Revised Columns and store updated data in the Revised Columns:

A Revised Column Name has the form:  
<logical\_column\_name>#<version\_tag>

Version\_tag is a string of the form: [0-9A-Z]+

Version tags are compared using the SQL '>' operator. A "greater" tag is a later version.

Example: Column "AMOUNT" is replaced by revised column "AMOUNT#1"

Example: Column "AMOUNT#1" is replaced by revised column "AMOUNT#2"

Populate a Revised Column using a Forward Cross-Edition Trigger (FCET).

The FCET must be owned by APPS.

The FCET name should be of the form  
<table\_name>\_F<change\_number>.

Example: "FND\_PROFILE\_OPTIONS\_F3"

The FCET must be created with the "disable" option.

The FCET for the second and subsequent change to a given table must specify that it "follows <previous\_fcet\_name->" in the trigger definition.

For information on the FOLLOW clause, see: Oracle Database PL/SQL Language Reference.

The FCET must be created using a DBDRV command that executes the create script in phase=ccet.

The FCET must be applied using a DBDRV command that calls  
AD\_ZD\_TABLE\_APPLY <fcet\_name>. in phase=ccet.

Tip: Use the Forward Cross-Edition Trigger template.

Do not drop an existing table until the Cleanup phase of patch execution.

For an example of a deferred drop, see: Code Example of a Deferred Drop.

Do not drop an existing column. Columns that are replaced by new revised columns will be dropped automatically at cleanup. Dropping a logical column is not supported.

Do not rename an existing table.

For more information on guidelines for managing tables, see: Oracle Database Administrator's Guide 11g Release 2 (11.2) For more information on using XDF (XML Definition File) features, see: XML Definition File Utility.

Index

-----

Index Definition Standards:

Index Name must contain an underscore ('\_'). (GSCC File.Gen.39)

The Unique Index on a Seed Data Table must include ZD\_EDITION\_NAME.

Note: This will be implemented automatically when you call "ad\_zd\_seed\_upgrade" on your seed data table, but if you add a new unique index to an existing seed data table you will need to include the ZD\_EDITION\_NAME column in your index definition.

The Unique Index on a Seed Data Table should have at least one not-null column besides ZD\_EDITION\_NAME.

Note: If the unique index has all nullable columns, then we expect each row in the table to have at least one non-null column value for the indexed columns. You must ensure that this is true as part of the Oracle E-Business Suite 12.2 upgrade (select rows where all indexed columns are null and either delete or update as needed to meet this standard).

Index Key Length should be less than 3125 bytes.

Index Key Length is the sum of the column lengths for each column in the index. The index key length is 3125 bytes for each column in the index.

If the Index Key Length is greater than 3125 bytes, then the index cannot be revised using "online" index definition, and a full table lock will be held during index revision.

A function-based index must not reference editioned Oracle E-Business Suite objects (built-in database functions such as "UPPER()") are acceptable).

Index Dynamic DDL Standards:

"CREATE INDEX ... ON ..." statement must specify the fully qualified table name, not the APPS Table synonym.

Good: create index SOME\_TABLE\_N1 on SCHEMA.SOME\_TABLE ...

Bad: create index SOME\_TABLE\_N1 on SOME\_TABLE ...

Index Online Patching Compliance Standards:

Deliver the index definition using ODF or XDF.

(Automatic) New or revised indexes will be initially created disabled and with an alternate name. These indexes will be enabled and renamed during cutover processing.

Do not drop an existing index until the Cleanup phase of patch execution.

For an example of a deferred drop, see: Code Example of a Deferred Drop.

For more information on indexes, see: Oracle Database Concepts 11g Release 2 (11.2).

For more information on using XDF (XML Definition File) features, see: XML Definition File Utility.

Integrity Constraint

-----

Integrity Constraint Definition Standards:

Constraint Name must contain an underscore ('\_').

Foreign Key Constraint should not be created, and must not reference a Seed Data Table.

Integrity Constraint Dynamic DDL Standards:

If you dynamically create a primary or unique constraint against a Seed Data Table, the constraint must include the ZD\_EDITION\_NAME column.

Integrity Constraint Online Patching Compliance Standards:

Deliver constraint definition using ODF or XDF. (GSCC File.Sql.81)

New or revised constraints will be initially created as disabled, and will be enabled during Cutover.

For more information on integrity constraints, see: Oracle Database Concepts 11g Release 2 (11.2).

For more information on using XDF (XML Definition File) features, see: XML Definition File Utility.

Materialized View (MV)

-----

The Materialized View is a non-editioned object type, and therefore a materialized view cannot directly reference editioned objects. To avoid this limitation, Oracle E-Business Suite Online Patching technology implements a new Effectively Editioned Materialized View compound object. Application developers create and maintain the Materialized View Definition (query) in an ordinary view. The Online Patching technology then automatically maintains a corresponding Materialized View Implementation that is legal for editioned databases.

MV Definition Standards:

A Materialized View Name must be unique within the first 29 bytes.

A Materialized View Definition must be stored in an ordinary view called MV\_NAME||'%'.

Create or replace the Materialized View Definition as an ordinary view called mv\_name||'%'.

Test the MV Definition for accuracy before generating the MV Implementation.

For example:

```
create or replace view FND_EXAMPLE_MV# as select ... ;
select * from fnd_example_mv#;
```

The Materialized View Implementation is automatically generated from the MV Definition using the AD\_ZD\_MVIEW.UPGRADE procedure.

The syntax is exec ad\_zd\_mvview.upgrade(<MV\_OWNER>, <MV\_NAME>)

Do not attempt to directly create or replace the Materialized View Implementation Query. To recreate an MV Implementation, call the AD\_ZD\_MVIEW.UPGRADE procedure.

A Materialized View Definition must specify a column alias for each item in the select list.

Failure to specify a column alias may cause the error  
ORA-00998 "must name this expression with a column alias"

Example: change select sum(EMP.SALARY), ...  
to select sum(EMP.SALARY) SUM\_EMP\_SALARY, ...

A Materialized View Query must not reference editioned PL/SQL functions.

If the MV definition references an editioned PL/SQL function, the MV Implementation will fail to generate and the MV will be unusable.

For examples of replacing PL/SQL function calls with equivalent SQL in Materialized Views, see: Examples of SQL Replacements for PL/SQL Functions

A Materialized View should use 'REFRESH FORCE' instead of 'REFRESH FAST'. The 'FORCE' option allows the MV to fall back to using a complete refresh in situations where the fast refresh is not possible.

See: Oracle Database SQL Language Reference 11g Release 2 (11.2) for more information on the "REFRESH FORCE" option.

If the MV Implementation content must be automatically refreshed after patching, include the /\*AUTOREFRESH\*/ comment tag in the MV Definition query.

Do not specify the /\*AUTOREFRESH\*/ tag for large MVs that will take a long time to refresh. For these cases use a concurrent program to refresh the MV after patching cutover.

Example: create or replace view FND\_EXAMPLE\_MV# as select  
/\*AUTOREFRESH\*/ ... ;

MV Usage Standards:

Do not assume that Fast Refresh is always possible. After an online patch, Complete Refresh may be required. When refreshing a Materialized View, use the 'FORCE' clause instead of 'FAST'.

See: Oracle Database SQL Language Reference 11g Release 2 (11.2) for more information on the 'FORCE' option.

MV Dynamic DDL Standards:

Use AD\_MV to execute Dynamic DDL for materialized views. Here is an example of creating a materialized view using the AD\_MV package:

```
--
-- Code Example: Create a materialized view using AD_MV interface.
--
-- Note:
-- when executed in the Run Edition, the MV is created immediately.
-- when executed in the Patch Edition, the MV is generated at CUTOVER.
--
begin
-- Create MV
ad_mv.create_mv('FND_EXAMPLE_MV',
'create materialized view FND_EXAMPLE_MV ||
' tablespace '||ad_mv.g_mv_data_tablespace||' ||
' build deferred refresh on demand as ||
'select /*AUTOREFRESH*/ (||
' upper(oracle_username) USERNAME ||
', decode(read_only_flag, 'C', 'pub', 'E', 'appls', 'U', 'apps')
USERTYPE ||
' from fnd_oracle_userid ||
' where read_only_flag in ('C', 'E', 'U') ');
```

```

-- End of Code Example.

MV Online Patching Compliance Standards:

Deliver Materialized View using XDF. (GSCC File.Sql.82)

Do not drop an existing materialized view until the Cleanup phase of patch
execution.

Do not attempt to upgrade, refresh or access a Materialized View
Implementation in the Patch Edition. Although the MV Implementation is
visible to the Patch Edition, it continues to implement the Run Edition of
the definition until the cutover phase. MV implementations are
automatically regenerated as needed at the cutover phase.

If an online patch must manually refresh the MV Implementation
contents, submit a concurrent request to do the refresh. The concurrent
request will be run after cutover and therefore after the MV
implementation has been regenerated.

If the MV Definition specifies the /AUTOREFRESH/ comment tag, then
the MV Contents will be automatically refreshed whenever the MV
implementation is regenerated.

Do not drop an obsolete materialized view until the Cleanup phase of patch
execution.

For more information on materialized views, see: Oracle Database Concepts 11g
Release 2 (11.2).

For more information on using XDF (XML Definition File) features, see:
XML Definition File Utility.

Examples of SQL Replacements for PL/SQL Functions
-----

To "Editions-enable" the APPS schema, non-Editionable objects must not depend
on Editionable objects (NE !-> E). To meet this requirement, the database
object development standards specify that Materialized Views (Materialized
Views, or MVs, are non-Editionable) must not call PL/SQL functions (which are
Editionable).

The examples below demonstrate how to replace frequently-used Oracle
Applications Technology PL/SQL function calls with an equivalent SQL in
Materialized Views. You may continue to call built-in PL/SQL functions such as
"upper()". fnd_profile.value() replaced with a SQL sub-select:

Before:

fnd_profile.value('MSC_HUB_REGION_INSTANCE')

After:

(select profile_option_value
 from fnd_profile_option_values
 where level_id = 10001
 and (profile_option_id, application_id) =
 (select profile_option_id, application_id
  from fnd_profile_options
  where profile_option_name = 'MSC_HUB_REGION_INSTANCE'))

Notes:

This replacement is valid ONLY in a Materialized View. For other uses of
fnd_profile.value(), continue using the normal PL/SQL call.

The general case for fetching profile option values is very complex, that
is why there is a PL/SQL package dedicated to doing it. But materialized
views results have to be valid in any context, so profile options
referenced in materialized views should only have site-level values, and
the replacement SQL only needs to support fetching the site level value.

This replacement SQL will only use the profile option value set at the site
level.

fnd_message.get_string() replaced with a SQL sub-select

Before:

fnd_message.get_string('MSC','MSC_HUB_UNASSIGNED')

After:

(select substrb(REPLACE(message_text, 's', 'S'),1,2000)
 from fnd_new_messages m, fnd_application a
 where m.message_name = 'MSC_HUB_UNASSIGNED'
 and m.language_code = 'US'
 and a.application_short_name = 'MSC'
 and m.application_id = a.application_id)

Notes:

This replacement is valid ONLY in a Materialized View. For other uses of
fnd_message.get_string(), continue using the normal PL/SQL call.

This replacement SQL will only retrieve the US language message text and is
not sensitive to any session language settings.

MV queries cannot contain a sub-SELECT within the main SELECT clause;
therefore, the replacement SQL is a bit trickier if the function call was
used in the MV SELECT clause.

Before:

select fnd_message.get_string('FND', 'CANCEL')
from dual
where 1=1
/

After:

select fmsg.result
from dual
, (select substrb(REPLACE(message_text, 's', 'S'),1,2000) result
 from fnd_new_messages m, fnd_application a
 where m.message_name = 'CANCEL'
 and m.language_code = 'US'
 and a.application_short_name = 'FND'
 and m.application_id = a.application_id) fmsg
where 1=1
/

fnd_global.lookup_security_group() replaced with a SQL sub-select

Before:

fnd_global.lookup_security_group('INTEREST_STATUS', 279)

After:

(select nvl(max(lt.security_group_id), 0)
 from fnd_lookup_types lt
 where lt.view_application_id = 279
 and lt.lookup_type = 'INTEREST_STATUS'
 and lt.security_group_id in (
 0,
 to_number(decode(substrb(userenv('CLIENT_INFO'),55,1),
 ' ', '0',
 null, '0',
 substrb(userenv('CLIENT_INFO'),55,10))))))

Note: This replacement is valid ONLY in a Materialized View. For other uses of
fnd_global.security_group(), continue using the normal PL/SQL call.

Example of a Deferred Drop

Here is a code example of a deferred drop. This example is for a table:

REM dbdrv: sql -PROD -PATH -FILE \
REM dbdrv: none none none sqlplus sphase=last \
REM dbdrv: checkfile:PROD:-PATH:-FILE $un_fnd

-- Example logic to drop a table under online patching
--
-- Note: This script drops example table "APPLSYS.FND_EXAMPLE_TABLE".
-- To use this logic for another table, you must substitute "$un_fnd" with the
-- actual table owner token ("&un_capp short_name"), and
-- "FND_EXAMPLE_TABLE" with the actual table name.
--
-- Usage
-- @TABLE_DROP_SCRIPT <table_owner>

drop synonym FND_EXAMPLE_TABLE;
drop view $1..FND_EXAMPLE_TABLE;
exec ad_ed.load_dsl ('CLEANUP', 'drop table $1..FND_EXAMPLE_TABLE')

```

## References

For more information on the operation of Oracle E-Business Suite Online Patching, refer to Chapter 3, [Patching Procedures](#), in *Oracle E-Business Suite Maintenance Guide*.

## Change Log

Date	Description of Change
2025-06-06	<ul style="list-style-type: none"> <li>Added a new Section 3.2.2.</li> </ul>
2022-11-18	<ul style="list-style-type: none"> <li>Replaced Patch 31026891:R12.AD.C with new Patch 33993716:R12.AD.C for Release 12.2.3 and later.</li> </ul>
2020-11-17	<ul style="list-style-type: none"> <li>Made several updates to Section 3 for clarity.</li> </ul>
2020-08-10	<ul style="list-style-type: none"> <li>Replaced references to Patch 27962989 with new Patch 31026891.</li> </ul>
2020-08-07	<ul style="list-style-type: none"> <li>Updated GSCC Patch Report requirements in Section 1.</li> <li>Clarified multitenant-related instructions in Section 3.2.</li> </ul>
2020-02-06	<ul style="list-style-type: none"> <li>Updated GSCC Patch Report requirements in Section 1.</li> </ul>
2020-02-06	<ul style="list-style-type: none"> <li>Updated GSCC Patch Report requirements in Section 1.</li> </ul>
2019-12-11	<ul style="list-style-type: none"> <li>Updated patch number in Section 4 patch readme.</li> <li>Made various structural changes to enhance navigation.</li> </ul>
2016-09-14	<ul style="list-style-type: none"> <li>Added instructions (Section 3.4) for using ADZDMVREFRESHNEEDED.sql.</li> </ul>
2014-10-21	<ul style="list-style-type: none"> <li>Added information on how to fix data corruption.</li> </ul>
2013-10-21	<ul style="list-style-type: none"> <li>Initial publication.</li> </ul>

My Oracle Support Knowledge [Document 1531121.1](#) by Oracle E-Business Suite Development.

[Copyright © 2013, 2025, Oracle and/or its affiliates.](#)

Didn't find what you are looking for?

[Ask in Community...](#)

Related

Products

- Oracle E-Business Suite > Applications Technology > Lifecycle Management > Oracle Applications DBA > Upgrade related issues > Pre-Upgrade Issues
- Oracle E-Business Suite > Applications Technology > Technology Components > Oracle Applications Utilities > Patch Application Issues > Readme/Instructions Issues
- Oracle E-Business Suite > Applications Technology > Technology Components > Oracle E-Business Suite Technology Stack > Patch Application Issues > Readme/Instructions Issues

Keywords

DOCUMENTATION; E-BUSINESS; FAMILY PACK; INSTALL; README; RELEASE NOTES; UPGRADE

Errors

AR-2011

Translations

- English Source
- Japanese 日本語

[Back to Top](#)