

Oracle Banking Trade Finance

Multi-Tenant Patch-set Deployment



Release 14.8.1.0.0
G46016-01
October 2025

ORACLE®

Copyright © 2007, 2025, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Purpose	i
Audience	i
Documentation Accessibility	i
Critical Patches	i
Diversity and Inclusion	ii
Conventions	ii
Related Resources	ii
Screenshot Disclaimer	ii

1 Overview of Applications in an Application Container

1.1 Managing Applications in an Application Container	1
1.2 Application Maintenance	1
1.2.1 Application Installation	2
1.2.2 Application Upgrade	2

2 Patch-set Application Steps

2.1 Application Upgrade	1
2.1.1 Purpose	1
2.1.2 Steps to be Followed	1
2.1.2.1 Start Application Upgrade	2
2.1.2.2 Compiling Incremental Units	2
2.1.2.3 Recompilation of invalids	3
2.1.2.4 End Application upgrade	3
2.1.2.5 Start Application upgrade	3
2.1.2.6 Application Root Objects Conversion for New Objects	3
2.1.2.7 Application Root Objects Conversion for Existing Objects	4
2.1.2.8 Recompilation of Invalids	4
2.1.2.9 End Application Upgrade	4
2.1.3 Purpose	5
2.1.4 Steps to be Followed	5

3 Step by Step Execution

3.1	Pre-Requisites	1
3.2	Patch-set Application Step by Step with Screenshots	2

Index

Preface

- [Purpose](#)
- [Audience](#)
- [Documentation Accessibility](#)
- [Critical Patches](#)
- [Diversity and Inclusion](#)
- [Conventions](#)
- [Related Resources](#)
- [Screenshot Disclaimer](#)

Purpose

Major changes to an application constitute application upgrades. During the upgrade, Oracle Database automatically clones the application root, and the application PDBs point to the clone.

Application upgrade can be performed in the application root only, and application PDBs applies the changes in the upgrade when they synchronize with the application.

Audience

This guide is intended for anyone responsible for installing Oracle Banking Application.

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 customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at [Critical Patches, Security Alerts and Bulletins](#). All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by [Oracle Software Security Assurance](#).

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

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.
<i>italic</i>	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.

Related Resources

For more information, see these Oracle Banking Trade Finance resources:

- [Oracle Banking Trade Finance Release Notes](#)
- [Oracle Banking Trade Finance Install & Upgrade](#)

Screenshot Disclaimer

Personal information used in the interface or documents is dummy and does not exist in the real world. It is only for reference purposes.

1

Overview of Applications in an Application Container

- [Managing Applications in an Application Container](#)
In an application container, an application is the named, versioned set of application common objects stored in the application root. In this context, “application” means “application back-end.” Application common objects include user accounts, tables, PL/SQL packages, and so on. An application can be shared with the application PDBs that belong to the application root.
- [Application Maintenance](#)
Application maintenance refers to installing, uninstalling, upgrading, or patching an application.

1.1 Managing Applications in an Application Container

In an application container, an application is the named, versioned set of application common objects stored in the application root. In this context, “application” means “application back-end.” Application common objects include user accounts, tables, PL/SQL packages, and so on. An application can be shared with the application PDBs that belong to the application root.

On performing application changes, application PDBs can synchronize with the application in the application root. The application container also manages the versions of the application and the patches to the application:

- While installing an application, user must specify the application version number.
- While upgrading an application, user must specify the old application version number and the new application version number.

As the application evolves, the application container maintains all of the versions that are applied.

1.2 Application Maintenance

Application maintenance refers to installing, uninstalling, upgrading, or patching an application.

Perform application installation, upgrade, and patching operations using an ALTER PLUGGABLE DATABASE APPLICATION statement.

The basic steps for application maintenance are as follows:

1. Log in to the application root.
2. Begin the operation with an ALTER PLUGGABLE DATABASE APPLICATION ... BEGIN statement in the application root.
3. Execute the application maintenance statements.
4. End the operation with an ALTER PLUGGABLE DATABASE APPLICATION ... END statement.

These statements can be issued in the same user session or in different user sessions.

- [Application Installation](#)
- [Application Upgrade](#)

An application upgrade is a major change to an installed application.

1.2.1 Application Installation

An application installation is the initial creation of a master application definition. A typical installation creates user accounts, tables, and PL/SQL packages.

Refer **Multi-Tenant_Deployment.pdf** for more details on the application installation.

1.2.2 Application Upgrade

An application upgrade is a major change to an installed application.

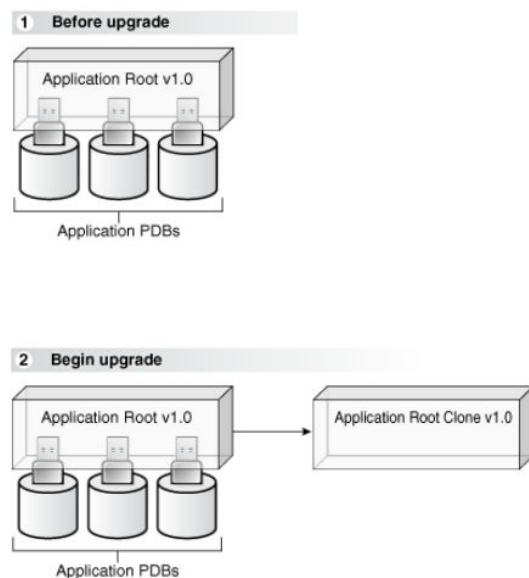
Typically, an upgrade changes the physical architecture of the application. For example, an upgrade might add new tables, and packages, or alter the definitions of existing objects.

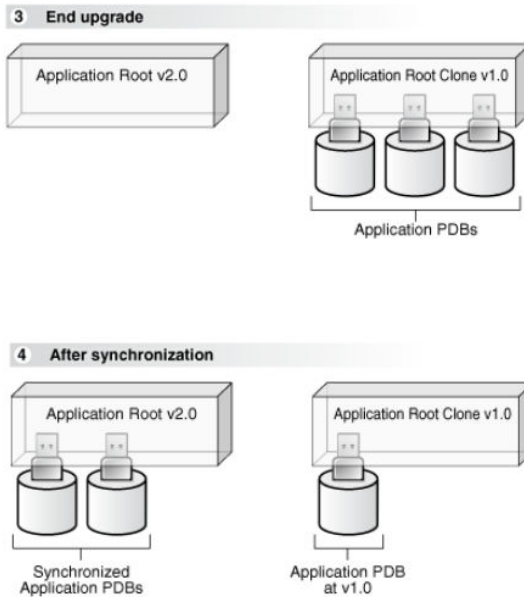
To upgrade the application, specify the following in the ALTER PLUGGABLE DATABASE APPLICATION statement:

- Name of the application
- Old application version number
- New application version number

During an application upgrade, the application remains available. To make this availability possible, Oracle Database clones the application root.

The following figure gives an overview of the application upgrade process.





When an application is upgraded, Oracle Database automatically clones the application root.

During the upgrade, application PDBs point to the clone and applications continue to run during the upgrade. Application PDBs can perform DML on metadata-linked and tables and views and query data-linked tables.

After the upgrade, the application root clone remains and continues to support any application PDB that still uses the pre-upgrade version of the application in the clone.

Application PDBs that are synchronized are pointed to the upgraded application root. Application PDBs that are not synchronized might continue to use the clone.

2

Patch-set Application Steps

Multi entity application root/PDB based setup has to be available to perform 18c database application upgrade for applying the patch-set. Refer Multi-Tenant_Deployment.docx for the deployment and installation steps.

Patch-set can be applied by following below steps in sequential order, and detail of each steps explained as separate sections subsequently.

- Application Upgrade
- Synchronize application PDBs

Patch-set Deployment Pre-requisites:

- Download the required patch-set zip file and unzip it in a local path.
- Verify whether the property files (fcubs.properties and env.properties) have the application root schema details where the application is available, if not update the approot schema details through installer (Refer OBTF_Property_File_Creation.docx for more details) and re-generate the files.
- Make sure to set the flag PATCHSET_INSTALLATION to 'Y'.
- [Application Upgrade](#)

2.1 Application Upgrade

- [Purpose](#)
Major changes to an application constitute application upgrades. During the upgrade, Oracle Database automatically clones the application root and the application PDBs point to the clone.
- [Steps to be Followed](#)
- [Purpose](#)
- [Steps to be Followed](#)

2.1.1 Purpose

Major changes to an application constitute application upgrades. During the upgrade, Oracle Database automatically clones the application root and the application PDBs point to the clone.

Application upgrade can be performed in the application root only, and application PDBs apply the changes in the upgrade when they synchronize with the application.

2.1.2 Steps to be Followed

Below steps to be followed to initiate application upgrade:

- Start Application upgrade
- Compiling Incremental Units

- Recompilation of invalids
- End Application upgrade
- Start Application upgrade
- Application Root objects conversion for new objects
- Application Root objects conversion for existing objects
- Recompilation of invalids
- End Application upgrade
- [Start Application Upgrade](#)
An ALTER PLUGGABLE DATABASE APPLICATION statement has to be issued to upgrade an application in the application root.
- [Compiling Incremental Units](#)
- [Recompilation of invalids](#)
- [End Application upgrade](#)
- [Start Application upgrade](#)
- [Application Root Objects Conversion for New Objects](#)
- [Application Root Objects Conversion for Existing Objects](#)
- [Recompilation of Invalids](#)
- [End Application Upgrade](#)

2.1.2.1 Start Application Upgrade

An ALTER PLUGGABLE DATABASE APPLICATION statement has to be issued to upgrade an application in the application root.

Each upgrade must be associated with an application name, starting version number, and ending version number.

- The common user must have the DBA privilege, and the privilege must be commonly granted in the application root.
- The application root must be in open read/write.
- Run the below script for initiating an application upgrade. This will initiate the application from current version to the next version (patch-set version).

[Start Upgrade](#)

Input sample for the script:

Spool Path	<< Any local path>>
Application next version	14.4.0.0.0

2.1.2.2 Compiling Incremental Units

Patch-set objects have to be loaded using bat file [E.g.: SMSDBCompileRun.bat, TFDBCompileRun.bat] by silent installer for respective product processor.

Compile the incremental SMS units using /INSTALLER/SOFT/SMSDBCompileRun.sh for UNIX installations or /INSTALLER/SOFT/SMSDBCompileRun.bat for Windows installations.

Compile the incremental OBTF units using `/INSTALLER/SOFT/TFDBCompileRun.sh` for UNIX installations or `/INSTALLER/SOFT/TFDBCompileRun.bat` for Windows installations.

2.1.2.3 Recompilation of invalids

As the sharing property of most of the objects are modified other than NONE, recompilation of objects is not allowed outside an application.

Recompilation of objects will be initiated inside the application upgrade for sanity with zero invalids with the below script:

[Recompilation of invalids](#)

2.1.2.4 End Application upgrade

Application upgrade can be performed in the application root only and end of the upgrade is performed with an `ALTER PLUGGABLE DATABASE APPLICATION END UPGRADE` statement.

Run the below script for ending an application upgrade for patch-set.

[End Application upgrade](#)

And run the invalid script by connecting to the common user in approot outside the upgrade.

[Invalid Recompilation Outside Upgrade.](#)

2.1.2.5 Start Application upgrade

Run the below script for initiating another application upgrade for object conversion. This will initiate the application from current version to the next version (patch-set version).

[Start Application upgrade](#)

Input sample for the script:

Spool Path	<< Any local path>>
Application next version	14.4.0.0.0

2.1.2.6 Application Root Objects Conversion for New Objects

As part of patch-set when there are new tables added which has to be converted as DL or when there is a new function id which is identified to be an approot function is provided, otherwise no conversion will happen as part of this step

Below script takes care of converting the new DL objects during patch-set based on the deployment model of the application during installation.

[New Object Conversion](#)

Input sample for the script:

Spool Path	<< Any local path>>
Application next version	14.4.0.0.0

2.1.2.7 Application Root Objects Conversion for Existing Objects

Various Sharing types of objects during installation:

- A static table will hold the information of selected table sharing as Data link. Other tables will be treated as Meta Data Link
- Sharing of object types such as INDEX, LOB, TABLE PARTITION, SEQUENCE, and DYNAMIC PACKAGES will remain as NONE.
- All other object types such as Packages, Procedures, Functions, and Synonyms would be converted as Meta Data Link sharing.

Sharing during upgrade:

Sharing of existing database objects will remain the same.

Below script takes care of converting the modified MDL objects when there is a re-creation [objects with Create or Replace command during creation] happens during patch-set.

[Application Root Objects Conversion](#)

Input sample for the script:

Spool Path	<< Any local path>>
Application next version	14.4.0.0.0

When there are new tables introduced as part of patch-set which has to be converted into DL will be done separately. The recommendation for the same will be provided as part of patch-set instructions for this case.

2.1.2.8 Recompilation of Invalids

As the sharing property of most of the objects are modified other than NONE, recompilation of objects is not allowed outside an application.

Recompilation of objects will be initiated inside the application upgrade for sanity with zero invalids with the below script:

[Recompilation of Invalids](#)

2.1.2.9 End Application Upgrade

Application upgrade can be performed in the application root only and end of the upgrade is performed with an ALTER PLUGGABLE DATABASE APPLICATION END UPGRADE statement.

Run the below script for ending an application upgrade for patch-set.

[End Application Upgrade](#)

And run the invalid script by connecting to the common user in approot outside the upgrade.

[Invalids_Recompilation_Outside_Upgrade](#)

2.1.3 Purpose

Synchronizing an application updates the application in the application PDB to the latest version in the application root. When an application is upgraded in an application root, an application PDB that belongs to the application root is not changed until it is synchronized.

Application PDBs synchronize with an application by running an ALTER PLUGGABLE DATABASE statement with the SYNC clause.

2.1.4 Steps to be Followed

Prerequisites

- The current user must have ALTER PLUGGABLE DATABASE system privilege.
- Ensure that the current container is the application PDB.
- Run an ALTER PLUGGABLE DATABASE APPLICATION statement with the SYNC clause.
- Run the below script to synchronize the PDBs with the latest application changes in the application root.

[PDB_Sync](#)

3

Step by Step Execution

- [Pre-Requisites](#)
This topic provides systematic instructions for pre-requisites.
- [Patch-set Application Step by Step with Screenshots](#)
This topic provides systematic instructions to patch-set application step by step with screenshots.

3.1 Pre-Requisites

This topic provides systematic instructions for pre-requisites.

1. Before applying the patch-set, we have to make sure the release is updates with the base version of the patch-set.

For Example, If the first patch-set of 14.2 is yet to applied, the release has to be updated as '14.2.0.0.0'. It can be verified with the below queries.

```
select param_name, param_val from CSTB_PARAM WHERE PARAM_NAME = 'RELEASE';  
select module_group_id, release from SMTB_MODULES_GROUP;
```

2. Another significant parameter is the values of application name and deployment type in CSTB_PARAM.

This value will be updated from the installer during Approot Object Conversion utility as part of deployment.

```
select param_name, param_val from cstb_param where PARAM_NAME in ('MULTI_TENANT_APP_NAME', 'MULTI_TENANT_DEPLOYMENT_MODEL');
```

The Application name of multi-tenant deployment will be stored in CSTB_PARAM as

Param_Name	Param_Val
MULTI_TENANT_APP_NAME	OBTF

The type of object conversion will be stored in CSTB_PARAM as

Param_Name	Param_Val
MULTI_TENANT_DEPLOYMENT_MODEL	SA (or) SAUA (or) SASDD (or) SASDC

SA □ Shared Application

SAUA □ Shared Application User Authentication

SASDD □ Shared Application Shared Data - Default

SASDC □ Shared Application Shared Data – Custom

(Optional) Enter the result of the procedure here.

3.2 Patch-set Application Step by Step with Screenshots

This topic provides systematic instructions to patch-set application step by step with screenshots.

1. Start Application upgrade

- a. Login into the Approot Schema as Common user.
- b. Run 01_Start_Upgrade.sql for initiating the application upgrade.
- c. User input has to be inputted for the below:

Spool Path	<< Any local path>>
Application next version	14.4.0.0.0

- d. Script will be executed as in the screen shot below and keep the SQL Plus session open for upcoming steps.

Execution Screenshot:

```

SQL> SPOOL ON
SQL> SET SQLBLANKLINES ON
SQL> SET SERVEROUTPUT ON
SQL> SET ERRORLOGGING ON
SQL> SET ECHO ON
SQL> prompt Welcome to Application PDB Configuration
Welcome to Application PDB Configuration
SQL> SPOOL "&SPOOL_PATH"
Enter value for spool_path: D:\FCUBS_14.3\Upgrade_Patchset_Approach\Documents\Review\Attachment\AVAILS\01Spool.txt
SQL>
SQL> DECLARE
2     l_app_name          VARCHAR2(128);
3     l_app_currver       VARCHAR2(38);
4     l_sql               VARCHAR2(256);
5 BEGIN
6
7     BEGIN
8         SELECT app_name
9             INTO l_app_name
10            FROM dba_applications
11             WHERE app_implicit <> 'Y'
12             AND app_name = (SELECT param_val FROM cstab_param WHERE Param_name = 'MULTI_TENANT_APP_NAME');
13     EXCEPTION
14     WHEN NO_DATA_FOUND THEN
15         dbms_output.put_line('Error Nodata---'||SQLERRM);
16     WHEN OTHERS THEN
17         dbms_output.put_line('Error others---'||SQLERRM);
18     END;
19     SELECT MAX(app_version)
20         INTO l_app_currver
21        FROM dba_app_versions
22       WHERE app_name = l_app_name;
23
24     l_sql := 'ALTER PLUGGABLE DATABASE APPLICATION ' || l_app_name || ' BEGIN UPGRADE ''' || l_app_currver || ''' TO ''' || '&P_APPLICATION_NEXTVER' || '''';
25     dbms_output.put_line('l_sql: ' || l_sql);
26     EXECUTE IMMEDIATE l_sql;
27
28     l_sql := 'ALTER SYSTEM SET DEFAULT_SHARING = NONE';
29     dbms_output.put_line('l_sql: ' || l_sql);
30     EXECUTE IMMEDIATE l_sql;
31
32 EXCEPTION
33     WHEN OTHERS THEN

```

```

14      WHEN NO_DATA_FOUND THEN
15          dbms_output.put_line('Error Nodata-->'||SQLERRM);
16      WHEN OTHERS THEN
17          dbms_output.put_line('Error others-->'||SQLERRM);
18      END;
19      SELECT MAX(app_version)
20      INTO l_app_currver
21      FROM dba_app_versions
22      WHERE app_name = l_app_name;
23
24      l_sql := 'ALTER PLUGGABLE DATABASE APPLICATION ' || l_app_name || ' BEGIN UPGRADE ''' || l_app_currver || ''' TO ''' || '&p_APPLICATION_NEXTVER' || '''';
25      dbms_output.put_line('l_sql: ' || l_sql);
26      EXECUTE IMMEDIATE l_sql;
27
28      l_sql := 'ALTER SYSTEM SET DEFAULT_SHARING = NONE';
29      dbms_output.put_line('l_sql: ' || l_sql);
30      EXECUTE IMMEDIATE l_sql;
31
32  EXCEPTION
33      WHEN OTHERS THEN
34          dbms_output.put_line('Error -->'||SQLERRM);
35  END;
36  /
37
38  Enter value for p_application_nextver: 14.2.0.0.2
39  SQL>
40  SQL>
41  SQL>
42  SQL>
43  SQL>
44  SQL>
45  SQL>
46  SQL>
47  SQL>
48  SQL>
49  SQL>
50  SQL>
51  SQL>
52  SQL>
53  SQL>
54  SQL>
55  SQL>
56  SQL>
57  SQL>
58  SQL>
59  SQL>
60  SQL>
61  SQL>
62  SQL>
63  SQL>
64  SQL>
65  SQL>
66  SQL>
67  SQL>
68  SQL>
69  SQL>
70  SQL>
71  SQL>
72  SQL>
73  SQL>
74  SQL>
75  SQL>
76  SQL>
77  SQL>
78  SQL>
79  SQL>
80  SQL>
81  SQL>
82  SQL>
83  SQL>
84  SQL>
85  SQL>
86  SQL>
87  SQL>
88  SQL>
89  SQL>
90  SQL>
91  SQL>
92  SQL>
93  SQL>
94  SQL>
95  SQL>
96  SQL>
97  SQL>
98  SQL>
99  SQL>
100 SQL>
101 SQL>
102 SQL>
103 SQL>
104 SQL>
105 SQL>
106 SQL>
107 SQL>
108 SQL>
109 SQL>
110 SQL>
111 SQL>
112 SQL>
113 SQL>
114 SQL>
115 SQL>
116 SQL>
117 SQL>
118 SQL>
119 SQL>
120 SQL>
121 SQL>
122 SQL>
123 SQL>
124 SQL>
125 SQL>
126 SQL>
127 SQL>
128 SQL>
129 SQL>
130 SQL>
131 SQL>
132 SQL>
133 SQL>
134 SQL>
135 SQL>
136 SQL>
137 SQL>
138 SQL>
139 SQL>
140 SQL>
141 SQL>
142 SQL>
143 SQL>
144 SQL>
145 SQL>
146 SQL>
147 SQL>
148 SQL>
149 SQL>
150 SQL>
151 SQL>
152 SQL>
153 SQL>
154 SQL>
155 SQL>
156 SQL>
157 SQL>
158 SQL>
159 SQL>
160 SQL>
161 SQL>
162 SQL>
163 SQL>
164 SQL>
165 SQL>
166 SQL>
167 SQL>
168 SQL>
169 SQL>
170 SQL>
171 SQL>
172 SQL>
173 SQL>
174 SQL>
175 SQL>
176 SQL>
177 SQL>
178 SQL>
179 SQL>
180 SQL>
181 SQL>
182 SQL>
183 SQL>
184 SQL>
185 SQL>
186 SQL>
187 SQL>
188 SQL>
189 SQL>
190 SQL>
191 SQL>
192 SQL>
193 SQL>
194 SQL>
195 SQL>
196 SQL>
197 SQL>
198 SQL>
199 SQL>
200 SQL>
201 SQL>
202 SQL>
203 SQL>
204 SQL>
205 SQL>
206 SQL>
207 SQL>
208 SQL>
209 SQL>
210 SQL>
211 SQL>
212 SQL>
213 SQL>
214 SQL>
215 SQL>
216 SQL>
217 SQL>
218 SQL>
219 SQL>
220 SQL>
221 SQL>
222 SQL>
223 SQL>
224 SQL>
225 SQL>
226 SQL>
227 SQL>
228 SQL>
229 SQL>
230 SQL>
231 SQL>
232 SQL>
233 SQL>
234 SQL>
235 SQL>
236 SQL>
237 SQL>
238 SQL>
239 SQL>
240 SQL>
241 SQL>
242 SQL>
243 SQL>
244 SQL>
245 SQL>
246 SQL>
247 SQL>
248 SQL>
249 SQL>
250 SQL>
251 SQL>
252 SQL>
253 SQL>
254 SQL>
255 SQL>
256 SQL>
257 SQL>
258 SQL>
259 SQL>
260 SQL>
261 SQL>
262 SQL>
263 SQL>
264 SQL>
265 SQL>
266 SQL>
267 SQL>
268 SQL>
269 SQL>
270 SQL>
271 SQL>
272 SQL>
273 SQL>
274 SQL>
275 SQL>
276 SQL>
277 SQL>
278 SQL>
279 SQL>
280 SQL>
281 SQL>
282 SQL>
283 SQL>
284 SQL>
285 SQL>
286 SQL>
287 SQL>
288 SQL>
289 SQL>
290 SQL>
291 SQL>
292 SQL>
293 SQL>
294 SQL>
295 SQL>
296 SQL>
297 SQL>
298 SQL>
299 SQL>
300 SQL>
301 SQL>
302 SQL>
303 SQL>
304 SQL>
305 SQL>
306 SQL>
307 SQL>
308 SQL>
309 SQL>
310 SQL>
311 SQL>
312 SQL>
313 SQL>
314 SQL>
315 SQL>
316 SQL>
317 SQL>
318 SQL>
319 SQL>
320 SQL>
321 SQL>
322 SQL>
323 SQL>
324 SQL>
325 SQL>
326 SQL>
327 SQL>
328 SQL>
329 SQL>
330 SQL>
331 SQL>
332 SQL>
333 SQL>
334 SQL>
335 SQL>
336 SQL>
337 SQL>
338 SQL>
339 SQL>
340 SQL>
341 SQL>
342 SQL>
343 SQL>
344 SQL>
345 SQL>
346 SQL>
347 SQL>
348 SQL>
349 SQL>
350 SQL>
351 SQL>
352 SQL>
353 SQL>
354 SQL>
355 SQL>
356 SQL>
357 SQL>
358 SQL>
359 SQL>
360 SQL>
361 SQL>
362 SQL>
363 SQL>
364 SQL>
365 SQL>
366 SQL>
367 SQL>
368 SQL>
369 SQL>
370 SQL>
371 SQL>
372 SQL>
373 SQL>
374 SQL>
375 SQL>
376 SQL>
377 SQL>
378 SQL>
379 SQL>
380 SQL>
381 SQL>
382 SQL>
383 SQL>
384 SQL>
385 SQL>
386 SQL>
387 SQL>
388 SQL>
389 SQL>
390 SQL>
391 SQL>
392 SQL>
393 SQL>
394 SQL>
395 SQL>
396 SQL>
397 SQL>
398 SQL>
399 SQL>
400 SQL>
401 SQL>
402 SQL>
403 SQL>
404 SQL>
405 SQL>
406 SQL>
407 SQL>
408 SQL>
409 SQL>
410 SQL>
411 SQL>
412 SQL>
413 SQL>
414 SQL>
415 SQL>
416 SQL>
417 SQL>
418 SQL>
419 SQL>
420 SQL>
421 SQL>
422 SQL>
423 SQL>
424 SQL>
425 SQL>
426 SQL>
427 SQL>
428 SQL>
429 SQL>
430 SQL>
431 SQL>
432 SQL>
433 SQL>
434 SQL>
435 SQL>
436 SQL>
437 SQL>
438 SQL>
439 SQL>
440 SQL>
441 SQL>
442 SQL>
443 SQL>
444 SQL>
445 SQL>
446 SQL>
447 SQL>
448 SQL>
449 SQL>
450 SQL>
451 SQL>
452 SQL>
453 SQL>
454 SQL>
455 SQL>
456 SQL>
457 SQL>
458 SQL>
459 SQL>
460 SQL>
461 SQL>
462 SQL>
463 SQL>
464 SQL>
465 SQL>
466 SQL>
467 SQL>
468 SQL>
469 SQL>
470 SQL>
471 SQL>
472 SQL>
473 SQL>
474 SQL>
475 SQL>
476 SQL>
477 SQL>
478 SQL>
479 SQL>
480 SQL>
481 SQL>
482 SQL>
483 SQL>
484 SQL>
485 SQL>
486 SQL>
487 SQL>
488 SQL>
489 SQL>
490 SQL>
491 SQL>
492 SQL>
493 SQL>
494 SQL>
495 SQL>
496 SQL>
497 SQL>
498 SQL>
499 SQL>
500 SQL>
501 SQL>
502 SQL>
503 SQL>
504 SQL>
505 SQL>
506 SQL>
507 SQL>
508 SQL>
509 SQL>
510 SQL>
511 SQL>
512 SQL>
513 SQL>
514 SQL>
515 SQL>
516 SQL>
517 SQL>
518 SQL>
519 SQL>
520 SQL>
521 SQL>
522 SQL>
523 SQL>
524 SQL>
525 SQL>
526 SQL>
527 SQL>
528 SQL>
529 SQL>
530 SQL>
531 SQL>
532 SQL>
533 SQL>
534 SQL>
535 SQL>
536 SQL>
537 SQL>
538 SQL>
539 SQL>
540 SQL>
541 SQL>
542 SQL>
543 SQL>
544 SQL>
545 SQL>
546 SQL>
547 SQL>
548 SQL>
549 SQL>
550 SQL>
551 SQL>
552 SQL>
553 SQL>
554 SQL>
555 SQL>
556 SQL>
557 SQL>
558 SQL>
559 SQL>
560 SQL>
561 SQL>
562 SQL>
563 SQL>
564 SQL>
565 SQL>
566 SQL>
567 SQL>
568 SQL>
569 SQL>
570 SQL>
571 SQL>
572 SQL>
573 SQL>
574 SQL>
575 SQL>
576 SQL>
577 SQL>
578 SQL>
579 SQL>
580 SQL>
581 SQL>
582 SQL>
583 SQL>
584 SQL>
585 SQL>
586 SQL>
587 SQL>
588 SQL>
589 SQL>
590 SQL>
591 SQL>
592 SQL>
593 SQL>
594 SQL>
595 SQL>
596 SQL>
597 SQL>
598 SQL>
599 SQL>
600 SQL>
601 SQL>
602 SQL>
603 SQL>
604 SQL>
605 SQL>
606 SQL>
607 SQL>
608 SQL>
609 SQL>
610 SQL>
611 SQL>
612 SQL>
613 SQL>
614 SQL>
615 SQL>
616 SQL>
617 SQL>
618 SQL>
619 SQL>
620 SQL>
621 SQL>
622 SQL>
623 SQL>
624 SQL>
625 SQL>
626 SQL>
627 SQL>
628 SQL>
629 SQL>
630 SQL>
631 SQL>
632 SQL>
633 SQL>
634 SQL>
635 SQL>
636 SQL>
637 SQL>
638 SQL>
639 SQL>
640 SQL>
641 SQL>
642 SQL>
643 SQL>
644 SQL>
645 SQL>
646 SQL>
647 SQL>
648 SQL>
649 SQL>
650 SQL>
651 SQL>
652 SQL>
653 SQL>
654 SQL>
655 SQL>
656 SQL>
657 SQL>
658 SQL>
659 SQL>
660 SQL>
661 SQL>
662 SQL>
663 SQL>
664 SQL>
665 SQL>
666 SQL>
667 SQL>
668 SQL>
669 SQL>
670 SQL>
671 SQL>
672 SQL>
673 SQL>
674 SQL>
675 SQL>
676 SQL>
677 SQL>
678 SQL>
679 SQL>
680 SQL>
681 SQL>
682 SQL>
683 SQL>
684 SQL>
685 SQL>
686 SQL>
687 SQL>
688 SQL>
689 SQL>
690 SQL>
691 SQL>
692 SQL>
693 SQL>
694 SQL>
695 SQL>
696 SQL>
697 SQL>
698 SQL>
699 SQL>
700 SQL>
701 SQL>
702 SQL>
703 SQL>
704 SQL>
705 SQL>
706 SQL>
707 SQL>
708 SQL>
709 SQL>
710 SQL>
711 SQL>
712 SQL>
713 SQL>
714 SQL>
715 SQL>
716 SQL>
717 SQL>
718 SQL>
719 SQL>
720 SQL>
721 SQL>
722 SQL>
723 SQL>
724 SQL>
725 SQL>
726 SQL>
727 SQL>
728 SQL>
729 SQL>
730 SQL>
731 SQL>
732 SQL>
733 SQL>
734 SQL>
735 SQL>
736 SQL>
737 SQL>
738 SQL>
739 SQL>
740 SQL>
741 SQL>
742 SQL>
743 SQL>
744 SQL>
745 SQL>
746 SQL>
747 SQL>
748 SQL>
749 SQL>
750 SQL>
751 SQL>
752 SQL>
753 SQL>
754 SQL>
755 SQL>
756 SQL>
757 SQL>
758 SQL>
759 SQL>
760 SQL>
761 SQL>
762 SQL>
763 SQL>
764 SQL>
765 SQL>
766 SQL>
767 SQL>
768 SQL>
769 SQL>
770 SQL>
771 SQL>
772 SQL>
773 SQL>
774 SQL>
775 SQL>
776 SQL>
777 SQL>
778 SQL>
779 SQL>
780 SQL>
781 SQL>
782 SQL>
783 SQL>
784 SQL>
785 SQL>
786 SQL>
787 SQL>
788 SQL>
789 SQL>
790 SQL>
791 SQL>
792 SQL>
793 SQL>
794 SQL>
795 SQL>
796 SQL>
797 SQL>
798 SQL>
799 SQL>
800 SQL>
801 SQL>
802 SQL>
803 SQL>
804 SQL>
805 SQL>
806 SQL>
807 SQL>
808 SQL>
809 SQL>
810 SQL>
811 SQL>
812 SQL>
813 SQL>
814 SQL>
815 SQL>
816 SQL>
817 SQL>
818 SQL>
819 SQL>
820 SQL>
821 SQL>
822 SQL>
823 SQL>
824 SQL>
825 SQL>
826 SQL>
827 SQL>
828 SQL>
829 SQL>
830 SQL>
831 SQL>
832 SQL>
833 SQL>
834 SQL>
835 SQL>
836 SQL>
837 SQL>
838 SQL>
839 SQL>
840 SQL>
841 SQL>
842 SQL>
843 SQL>
844 SQL>
845 SQL>
846 SQL>
847 SQL>
848 SQL>
849 SQL>
850 SQL>
851 SQL>
852 SQL>
853 SQL>
854 SQL>
855 SQL>
856 SQL>
857 SQL>
858 SQL>
859 SQL>
860 SQL>
861 SQL>
862 SQL>
863 SQL>
864 SQL>
865 SQL>
866 SQL>
867 SQL>
868 SQL>
869 SQL>
870 SQL>
871 SQL>
872 SQL>
873 SQL>
874 SQL>
875 SQL>
876 SQL>
877 SQL>
878 SQL>
879 SQL>
880 SQL>
881 SQL>
882 SQL>
883 SQL>
884 SQL>
885 SQL>
886 SQL>
887 SQL>
888 SQL>
889 SQL>
890 SQL>
891 SQL>
892 SQL>
893 SQL>
894 SQL>
895 SQL>
896 SQL>
897 SQL>
898 SQL>
899 SQL>
900 SQL>
901 SQL>
902 SQL>
903 SQL>
904 SQL>
905 SQL>
906 SQL>
907 SQL>
908 SQL>
909 SQL>
910 SQL>
911 SQL>
912 SQL>
913 SQL>
914 SQL>
915 SQL>
916 SQL>
917 SQL>
918 SQL>
919 SQL>
920 SQL>
921 SQL>
922 SQL>
923 SQL>
924 SQL>
925 SQL>
926 SQL>
927 SQL>
928 SQL>
929 SQL>
930 SQL>
931 SQL>
932 SQL>
933 SQL>
934 SQL>
935 SQL>
936 SQL>
937 SQL>
938 SQL>
939 SQL>
940 SQL>
941 SQL>
942 SQL>
943 SQL>
944 SQL>
945 SQL>
946 SQL>
947 SQL>
948 SQL>
949 SQL>
950 SQL>
951 SQL>
952 SQL>
953 SQL>
954 SQL>
955 SQL>
956 SQL>
957 SQL>
958 SQL>
959 SQL>
960 SQL>
961 SQL>
962 SQL>
963 SQL>
964 SQL>
965 SQL>
966 SQL>
967 SQL>
968 SQL>
969 SQL>
970 SQL>
971 SQL>
972 SQL>
973 SQL>
974 SQL>
975 SQL>
976 SQL>
977 SQL>
978 SQL>
979 SQL>
980 SQL>
981 SQL>
982 SQL>
983 SQL>
984 SQL>
985 SQL>
986 SQL>
987 SQL>
988 SQL>
989 SQL>
990 SQL>
991 SQL>
992 SQL>
993 SQL>
994 SQL>
995 SQL>
996 SQL>
997 SQL>
998 SQL>
999 SQL>
1000 SQL>

```

2. Compiling Incremental Units

- a. Make sure that the fcubs.properties and env.properties are updated with approot schema details.
- b. Run the <Product Processor>DBCompileRun.bat from <Patchset>\INSTALLER\SOFT directory. DDL Compilation, Object Compilation and Static Data load will be done.

For Example: OBTF INSTALLATION

First load SMS objects first and then OBTF objects. i.e. Run SMSDBCompileRun.bat and after SMS object loading is completed, then initiate OBTF compilation Run TFDBCompileRun.bat

3. Recompilation of invalids

- a. Login into the Approot Schema as Common user
- b. Run 03_Invalids_Recompilation.sql for recompiling the invalids during application upgrade.
- c. No user input is required for this step.
- d. Script will be executed as in the screen shot below and keep the SQL Plus session open for upcoming steps.

Execution Screenshot:

```

C:\app\client\gribalac\product\18.0.0\client_1\bin\sqlplus.exe
SQL> prompt Welcome to Application Upgrade Invalids Recompilation
Welcome to Application Upgrade Invalids Recompilation
SQL> SPOOL "85POOL_PATH"
Enter value for spool_path: D:\FCUBS_14.3\Upgrade_Patchset_Approach\Documents\Review\Attachment\AVAILS\05Spool.txt
SQL>
SQL> DECLARE
2   inval_cnt NUMBER := 0;
3 BEGIN
4   WHILE inval_cnt < 3 LOOP
5     --SCRIPT
6     FOR i IN (Select 'alter ' || object_type || ' ' || object_name || ' compile' invalidobject1
7               FROM user_objects
8               WHERE status = 'INVALID'
9               AND object_type IN ('VIEW','SYNONYM','PROCEDURE','FUNCTION','PACKAGE','TRIGGER')
10              )
11     LOOP
12       DBMS_OUTPUT.PUT_LINE(chr(10));
13       EXECUTE IMMEDIATE i.invalidobject1;
14     END LOOP;
15     inval_cnt := inval_cnt + 1;
16   END LOOP;
17 END;
18 /

PL/SQL procedure successfully completed.

SQL>
SQL> DECLARE

```

```

C:\app\client\gribalac\product\18.0.0\client_1\bin\sqlplus.exe

PL/SQL procedure successfully completed.

SQL> select count(*) From user_objects Where status = 'INVALID';

COUNT(*)
-----
0
1 row selected.

SQL>
SQL> SET ERRORLOGGING OFF
SQL> SPOOL OFF
SQL>
SQL>
SQL>
SQL>
SQL>

```

4. End Application upgrade

- a. Login into the Approot Schema as Common user.
- b. Run 06_End_Upgrade.sql for recompiling the invalids during application upgrade.
- c. No user input is required for this step.
- d. Script will be executed as in the screen shot below.

Execution Screenshot:

```

C:\app\client\prdbase\product\18.0.0\client_1\bin\sqlplus.exe
SQL> prompt Welcome to Application PDB Configuration
Welcome to Application PDB Configuration
SQL> SPOOL "85POOL_PATH"
Enter value for spool_path: D:\FCUBS_14.3\Upgrade_Patchset_Approach\Documents\Review\Attachment\AVAILS\06Spool.txt
SQL>
SQL> DECLARE
2   l_app_name    VARCHAR2(128);
3   l_sql         VARCHAR2(256);
4 BEGIN
5
6     SELECT app_name
7     INTO l_app_name
8     FROM dba_applications
9     WHERE app_implicit <> 'Y'
10    AND app_name = (SELECT param_val FROM cstab_param WHERE param_name = 'MULTI_TENANT_APP_NAME');
11 EXCEPTION
12 WHEN NO_DATA_FOUND THEN
13     dbms_output.put_line('Error1 Nodata-->' || SQLERRM);
14 WHEN OTHERS THEN
15     dbms_output.put_line('Error1 others-->' || SQLERRM);
16 END;
17 l_sql := 'ALTER PLUGGABLE DATABASE APPLICATION ' || l_app_name || ' END UPGRADE ';
18 dbms_output.put_line('l_sql: ' || l_sql);
19 EXECUTE IMMEDIATE l_sql;
20
21 EXCEPTION
22 WHEN OTHERS THEN
23     dbms_output.put_line('Error --->' || SQLERRM);
24 END;
25 /
26
l_sql: ALTER PLUGGABLE DATABASE APPLICATION FCUBS END UPGRADE
PL/SQL procedure successfully completed.

SQL> SET ERRORLOGGING OFF
SQL> SPOOL OFF
SQL>

```

5. Start Application upgrade

- a. Login into the Approot Schema as Common user.
- b. Run 05_Start_Upgrade.sql for initiating the application upgrade.
- c. User input has to be inputted for the below:

Spool Path	<< Any local path>>
Application next version	14.4.0.0.0

- d. Script will be executed similar to step 1 above and keep the SQL Plus session open for upcoming steps.

6. Application Root objects conversion for new objects

- a. Login into the Approot Schema as Common user.
- b. Run 06_New_Object_Conversion.sql for converting new approot objects added during patch-set as DL
- c. User input has to be inputted for the below:

Spool Path	<< Any local path>>
Application next version	HUBUSER (common user name)

Execution Screenshot:

```

C:\app\client\gribale\product\18.0.0\client_1\bin\sqlplus.exe
SQL> prompt Welcome to Application PDB Configuration
Welcome to Application PDB Configuration
SQL> SPOOL "85POOL_PATH"
Enter value for spool_path: D:\VCUBS_14.3\Upgrade_Patchset_Approach\Documents\Review\Attachment\AVAILS\02Spool1.txt
SQL>
SQL> DECLARE
2   l_count NUMBER;
3   l_app_deployment VARCHAR2(30);
4 BEGIN
5   SELECT count(*)
6     INTO l_count
7   FROM user_objects
8   WHERE sharing = 'NONE' --to get the new set of DL approot objects if any
9     AND object_name IN (SELECT DISTINCT a.object_name
10                        FROM cstm_approot_objects a
11                        WHERE sharing = 'DL'
12                        AND UPPER(object_type) = 'TABLE'
13                        AND EXISTS (SELECT 1
14                                FROM user_objects b
15                                WHERE b.object_name = a.object_name)
16                        AND EXISTS (SELECT 1
17                                FROM cstm_approot_functions_menu c
18                                WHERE c.function_id = a.function_id
19                                AND c.modifiable IN ('Y', 'S')));
20 dbms_output.put_line('l_count: ' || l_count);
21 IF l_count > 0 THEN
22   dbms_output.put_line('New DL objects are available');
23   SELECT param_val
24     INTO l_app_deployment
25   FROM cstb_param
26   WHERE param_name = 'MULTI_TENANT_DEPLOYMENT_MODEL';
27   dbms_output.put_line('l_app_deployment: ' || l_app_deployment);
28 IF l_app_deployment IS NOT NULL AND l_app_deployment = 'SAUA' THEN
29   UPDATE sctb_menu_menu
30     SET menu.approot_flg = 'Y'
31     WHERE menu.function_id IN
32       (SELECT function_id
33        FROM cstm_approot_functions_menu
34        WHERE modifiable = 'S'
35        UNION
36        SELECT summary_fn_id
37        FROM cstm_approot_functions_menu
38        WHERE modifiable = 'S'
39        AND summary_fn_id IS NOT NULL) --SMS function id 'S'
40

```

```

function_id
90
91
92   LOOP
93     DBMS_OUTPUT.PUT_LINE(chr(10));
94     EXECUTE IMMEDIATE l.sqlobject;
95     DBMS_OUTPUT.PUT_LINE(l.sqlobject);
96   END LOOP;
97 EXCEPTION
98   WHEN OTHERS THEN
99     DBMS_OUTPUT.PUT_LINE('Error --->||SQLERRM');
100 END;
101 ELSE
102   dbms_output.put_line('No new DL objects available');
103 END IF;
104 EXCEPTION
105   WHEN OTHERS THEN
106     dbms_output.put_line('Error --->||SQLERRM');
107 END;
108 /
Enter value for p_approot_user: HUBUSER
old 77: FOR I IN (SELECT 'BEGIN ' || chr(10) || 'DBMS_PDB.SET_DATA_LINKED(''8P_APPROOT_USER'' ' ||
new 77: FOR I IN (SELECT 'BEGIN ' || chr(10) || 'DBMS_PDB.SET_DATA_LINKED(''HUBUSER'' ' ||
l_count: 1
New DL objects are available
l_app_deployment: SASOD

BEGIN
DBMS_PDB.SET_DATA_LINKED('HUBUSER','CSTM_CHECK2',1);
EXCEPTION
WHEN OTHERS then
DBMS_OUTPUT.PUT_LINE('ERROR ->' || SQLERRM);
END;

PL/SQL procedure successfully completed.

SQL>
SQL> SET ERRORLOGGING OFF
SQL> SPOOL OFF
SQL>

```

7. Application Root objects conversion for existing objects

- Login into the Approot Schema as Common user.
- Run 07_Object_Conversion.sql for initiating the application upgrade.
- User input has to be inputted for the below:

Spool Path	<< Any local path>>
Application next version	HUBUSER (common user name)

Execution Screenshot:

```

C:\app\client\gribalac\product\18.0.0\client_1\bin\sqlplus.exe
SQL> SET SERVEROUTPUT ON
SQL> SET ERRORLOGGING ON
SQL> SET ECHO ON
SQL> prompt Welcome to Upgrade object conversion
Welcome to Upgrade object conversion
SQL> SPOOL &SPOOL_PATH
Enter value for spool_path: D:\FCUBS_14.3\Upgrade_Patchset_Approach\Documents\Review\Attachment\AVAIL5\04Spool.txt
SQL>
SQL> DECLARE
  2   l_app_name          VARCHAR2(128);
  3 BEGIN
  4
  5     SELECT app_name
  6     INTO l_app_name
  7     FROM dba_applications
  8     WHERE app_implicit <> 'Y'
  9     AND app_name = (SELECT param_val FROM cstab_param WHERE Param_name = 'MULTI_TENANT_APP_NAME');
10
11    EXCEPTION
12    WHEN NO_DATA_FOUND THEN
13        dbms_output.put_line('Error1 Nodata-->'||SQLERRM);
14    WHEN OTHERS THEN
15        dbms_output.put_line('Error1 others-->'||SQLERRM);
16    END;
17
18    FOR I IN (SELECT 'BEGIN ' || chr(10) ||
19                'DBMS_PDB.SET_METADATA_LINKED(''&P_APPROOT_USER'' || ', '' || chr(10) ||
20                'Object Name || ''', '' || Namespace || '' ); ' || chr(10) ||
21                'EXCEPTION ' || chr(10) || 'WHEN OTHERS then ' || chr(10) ||
22                'DBMS_OUTPUT.PUT_LINE(''ERROR ->' || SQLERRM); ' || chr(10) ||
23                'chr(10) || 'END;' sqlobject
24            FROM user_objects
25            WHERE sharing = 'NONE'
26            AND object_type NOT IN ('INDEX', 'LOB', 'TABLE PARTITION', 'SEQUENCE')
27            AND (object_name NOT LIKE '%%' )
28            AND (created_appid = (select app_id from dba_applications where app_name = l_app_name) OR
29                modified_appid = (select app_id from dba_applications where app_name = l_app_name)) ) LOOP
30        DBMS_OUTPUT.PUT_LINE(chr(10));
31        EXECUTE IMMEDIATE l_sqlobject;
32    END LOOP;
33
34    EXCEPTION
35    WHEN OTHERS THEN
36        DBMS_OUTPUT.PUT_LINE('Error --->'||SQLERRM);
37    END;
38 /

```

```

C:\app\client\gribalac\product\18.0.0\client_1\bin\sqlplus.exe
5     SELECT app_name
6     INTO l_app_name
7     FROM dba_applications
8     WHERE app_implicit <> 'Y'
9     AND app_name = (SELECT param_val FROM cstab_param WHERE Param_name = 'MULTI_TENANT_APP_NAME');
10
11    EXCEPTION
12    WHEN NO_DATA_FOUND THEN
13        dbms_output.put_line('Error1 Nodata-->'||SQLERRM);
14    WHEN OTHERS THEN
15        dbms_output.put_line('Error1 others-->'||SQLERRM);
16    END;
17
18    FOR I IN (SELECT 'BEGIN ' || chr(10) ||
19                'DBMS_PDB.SET_METADATA_LINKED(''&P_APPROOT_USER'' || ', '' || chr(10) ||
20                'Object Name || ''', '' || Namespace || '' ); ' || chr(10) ||
21                'EXCEPTION ' || chr(10) || 'WHEN OTHERS then ' || chr(10) ||
22                'DBMS_OUTPUT.PUT_LINE(''ERROR ->' || SQLERRM); ' || chr(10) ||
23                'chr(10) || 'END;' sqlobject
24            FROM user_objects
25            WHERE sharing = 'NONE'
26            AND object_type NOT IN ('INDEX', 'LOB', 'TABLE PARTITION', 'SEQUENCE')
27            AND (object_name NOT LIKE '%%' )
28            AND (created_appid = (select app_id from dba_applications where app_name = l_app_name) OR
29                modified_appid = (select app_id from dba_applications where app_name = l_app_name)) ) LOOP
30        DBMS_OUTPUT.PUT_LINE(chr(10));
31        EXECUTE IMMEDIATE l_sqlobject;
32    END LOOP;
33
34    EXCEPTION
35    WHEN OTHERS THEN
36        DBMS_OUTPUT.PUT_LINE('Error --->'||SQLERRM);
37    END;
38 /
Enter value for p_approot_user: HUBUSER
old 18:          'DBMS_PDB.SET_METADATA_LINKED(''&P_APPROOT_USER'' || ', '' || chr(10) ||
new 18:          'DBMS_PDB.SET_METADATA_LINKED(''HUBUSER'' || ', '' || chr(10) ||
PL/SQL procedure successfully completed.
SQL>
SQL> SET ERRORLOGGING OFF
SQL> SPOOL OFF
SQL>

```

8. Recompilation of invalids

- Login into the Approot Schema as Common user.
- Run `08_Invalids_Recompilation.sql` for recompiling the invalids during application upgrade.
- No user input is required for this step.
- Script will be executed as in the screen shot below and keep the SQL Plus session open for upcoming steps.

Execution Screenshot:

```

C:\app\client\gribalac\product\18.0.0\client_1\bin\sqlplus.exe
SQL> prompt Welcome to Application Upgrade Invalids Recompilation
Welcome to Application Upgrade Invalids Recompilation
SQL> SPOOL "85POOL_PATH"
Enter value for spool_path: D:\FCUBS_14.3\Upgrade_Patchset_Approach\Documents\Review\Attachment\AVAILS\05Spool.txt
SQL>
SQL> DECLARE
2   inval_cnt NUMBER := 0;
3 BEGIN
4   WHILE inval_cnt < 3 LOOP
5     --SCRIPT
6     FOR j IN (Select 'alter ' || object_type || ' ' || object_name || ' compile' invalidobject1
7               FROM user_objects
8               WHERE status = 'INVALID'
9               AND object_type IN ('VIEW','SYNONYM','PROCEDURE','FUNCTION','PACKAGE','TRIGGER')
10              )
11     LOOP
12       DBMS_OUTPUT.PUT_LINE(chr(10));
13       EXECUTE IMMEDIATE j.invalidobject1;
14     END LOOP;
15     inval_cnt := inval_cnt + 1;
16   END LOOP;
17 END;
18 /

PL/SQL procedure successfully completed.
SQL>
SQL> DECLARE

```

```

C:\app\client\gribalac\product\18.0.0\client_1\bin\sqlplus.exe

PL/SQL procedure successfully completed.
SQL> select count(*) from user_objects where status = 'INVALID';
COUNT(*)
-----
0
1 row selected.
SQL>
SQL> SET ERRORLOGGING OFF
SQL> SPOOL OFF
SQL>
SQL>
SQL>
SQL>
SQL>

```

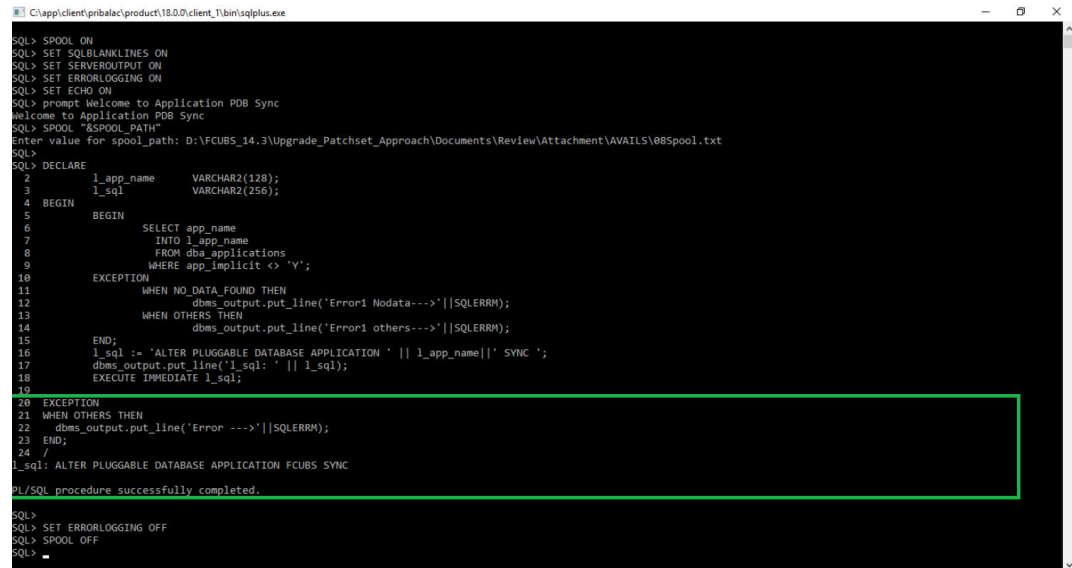
9. End Application upgrade

- a. Login into the Approot Schema as Common user.
- b. Run 06_End_Upgrade . sql **for recompiling the invalids during application upgrade.**
- c. No user input is required for this step.
- d. Script will be executed as that of step 4.

10. Synchronize application PDBs

- a. Login into the PDB Schema as Common user. For each PDB, this steps has to be done individually.
- b. Run 07_PDB_Sync . sql **for synching the application upgrade with PDBs.**
- c. No user input is required for this step.
- d. Script will be executed as in the screen shot below.

Execution Screenshot:



```
C:\app\client\gribalac\product\18.0.0\client_1\bin\sqlplus.exe

SQL> SPOOL ON
SQL> SET SQLBLANKLINES ON
SQL> SET SERVEROUTPUT ON
SQL> SET ERRORLOGGING ON
SQL> SET ECHO ON
SQL> prompt Welcome to Application PDB Sync
Welcome to Application PDB Sync
SQL> SPOOL "&SPOOL_PATH"
Enter value for spool_path: D:\FCUBS_14.3\Upgrade_Patchset_Approach\Documents\Review\Attachment\AVAIL5\08Spool.txt
SQL>
SQL> DECLARE
2   l_app_name      VARCHAR2(128);
3   l_sql           VARCHAR2(256);
4 BEGIN
5     BEGIN
6         SELECT app_name
7             INTO l_app_name
8             FROM dba_applications
9             WHERE app_implicit <> 'Y';
10    EXCEPTION
11    WHEN NO_DATA_FOUND THEN
12        dbms_output.put_line('Error1 Nodata-->' || SQLERRM);
13    WHEN OTHERS THEN
14        dbms_output.put_line('Error1 others-->' || SQLERRM);
15    END;
16    l_sql := 'ALTER PLUGGABLE DATABASE APPLICATION ' || l_app_name || ' SYNC ';
17    dbms_output.put_line('l_sql: ' || l_sql);
18    EXECUTE IMMEDIATE l_sql;
19
20 EXCEPTION
21 WHEN OTHERS THEN
22     dbms_output.put_line('Error -->' || SQLERRM);
23 END;
24 /
l_sql: ALTER PLUGGABLE DATABASE APPLICATION FCUBS SYNC
PL/SQL procedure successfully completed.

SQL>
SQL> SET ERRORLOGGING OFF
SQL> SPOOL OFF
SQL>
```

01_Start_Upgrade

This script is used for initiating an application upgrade. This will initiate the application from current version to the next version (patch-set version).

Syntax

```
SET VERIFY ON
SET HEAD ON
SET FEEDBACK 1
SET ARRAY 1
SET LINESIZE 10000
SET PAGESIZE 50000
SET LONG 10000
SET ECHO ON
SET TRIMSPPOOL ON
SET COLSEP ';'
SET SERVEROUT OFF
clear screen
SPOOL ON
SET SQLBLANKLINES ON
SET SERVEROUTPUT ON
SET ERRORLOGGING ON
SET ECHO ON
prompt Welcome to Application Upgrade initiation
SPOOL "&SPOOL_PATH"

DECLARE
    l_app_name    VARCHAR2(128);
    l_app_currver VARCHAR2(30);
    l_sql         VARCHAR2(256);
BEGIN
    BEGIN
        SELECT app_name
            INTO l_app_name
            FROM dba_applications
            WHERE app_implicit <> 'Y'
            AND app_name = (SELECT param_val FROM cstb_param WHERE Param_name =
'MULTI_TENANT_APP_NAME');
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            dbms_output.put_line('Error1 Nodata--->' || SQLERRM);
        WHEN OTHERS THEN
            dbms_output.put_line('Error1 others--->' || SQLERRM);
    END;

    BEGIN
        SELECT MAX(app_version)
            INTO l_app_currver
            FROM dba_app_versions
            WHERE app_name = l_app_name;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
```

```
        dbms_output.put_line('Error2 Nodata--->'||SQLERRM);
    WHEN OTHERS THEN
        dbms_output.put_line('Error2 others--->'||SQLERRM);
END;

l_sql := 'ALTER PLUGGABLE DATABASE APPLICATION ' || l_app_name||' BEGIN
UPGRADE '''|| l_app_currver || ''' TO '''|| '&P_APPLICATION_NEXTVER' ||'''';
dbms_output.put_line('l_sql: ' || l_sql);
EXECUTE IMMEDIATE l_sql;

l_sql := 'ALTER SYSTEM SET DEFAULT_SHARING = NONE';
dbms_output.put_line('l_sql: ' || l_sql);
EXECUTE IMMEDIATE l_sql;

EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('Error --->'||SQLERRM);
END;
/

SET ERRORLOGGING OFF
SPOOL OFF
```

Recompilation of invalids

Recompilation of objects will be initiated inside the application upgrade for sanity with zero invalids with the below script:

Syntax

```
/* Script for Shared Application + Shared Data */
SET VERIFY ON
SET HEAD ON
SET FEEDBACK 1
SET ARRAY 1
SET LINESIZE 10000
SET PAGESIZE 50000
SET LONG 10000
SET ECHO ON
SET TRIMSPPOOL ON
SET COLSEP ';'
SET SERVEROUT OFF
clear screen
SPOOL ON
SET SQLBLANKLINES ON
SET SERVEROUTPUT ON
SET ERRORLOGGING ON
SET ECHO ON
prompt Welcome to Application Upgrade Invalids Recompilation
SPOOL "&SPOOL_PATH"

DECLARE
    inval_cnt      NUMBER := 0;
    l_object_name  VARCHAR2(240);
BEGIN
    WHILE inval_cnt < 3 LOOP
        --SCRIPT
        FOR J IN (Select 'alter ' || object_type || ' ' || object_name || '
compile' invalidobject1,
                    object_name
                  FROM user_objects
                  WHERE status = 'INVALID'
                  AND created_appid IS NOT NULL
                  AND object_type IN
('VIEW', 'SYNONYM', 'PROCEDURE', 'FUNCTION', 'PACKAGE', 'TRIGGER', 'MATERIALIZED
VIEW'))
        LOOP
            BEGIN
                l_object_name := j.object_name;
                dbms_output.put_line(chr(10));
                EXECUTE IMMEDIATE J.invalidobject1;
            EXCEPTION
                WHEN OTHERS THEN
                    dbms_output.put_line('failed for -->' || l_object_name);
            END;
        END LOOP;
        inval_cnt := inval_cnt + 1;
    END LOOP;
```

```

        END LOOP;
    EXCEPTION
        WHEN OTHERS THEN
            dbms_output.put_line('FAILED FOR -->' || l_object_name);
    END;
/
DECLARE
    inval_cnt1        NUMBER := 0;
    l_object_name     VARCHAR2(240);
BEGIN
    WHILE inval_cnt1 < 3 LOOP
        --SCRIPT
        FOR k IN (Select 'alter package ' || object_name || ' compile body'
invalidobject2,
                object_name
                FROM user_objects
                WHERE status = 'INVALID'
                AND created_appid IS NOT NULL
                AND object_type IN ('PACKAGE BODY'))
        LOOP
            BEGIN
                l_object_name := k.object_name;
                dbms_output.put_line(chr(10));
                EXECUTE IMMEDIATE k.invalidobject2;
            EXCEPTION
                WHEN OTHERS THEN
                    dbms_output.put_line('FAILED FOR -->' || l_object_name);
            END;
        END LOOP;
        inval_cnt1 := inval_cnt1 + 1;
    END LOOP;
EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('FAILED FOR -->' || l_object_name);
END;
/
select count(*) From user_objects Where status = 'INVALID';

SET ERRORLOGGING OFF
SPOOL OFF

```

End Application upgrade

This script is run for ending an application upgrade for patch-set.

Pre-requisites:

Step 3 on Application associated pdb creation is completed

Syntax

```
SET VERIFY ON
SET HEAD ON
SET FEEDBACK 1
SET ARRAY 1
SET LINESIZE 10000
SET PAGESIZE 50000
SET LONG 10000
SET ECHO ON
SET TRIMSPOOL ON
SET COLSEP ';'
SET SERVEROUT OFF
clear screen
SPOOL ON
SET SQLBLANKLINES ON
SET SERVEROUTPUT ON
SET ERRORLOGGING ON
SET ECHO ON
prompt Welcome to Application end Upgrade
SPOOL "&SPOOL_PATH"

DECLARE
    l_app_name    VARCHAR2(128);
    l_sql         VARCHAR2(256);
BEGIN
    BEGIN
        SELECT app_name
            INTO l_app_name
            FROM dba_applications
            WHERE app_implicit <> 'Y'
              AND app_name = (SELECT param_val FROM cstb_param WHERE param_name
= 'MULTI_TENANT_APP_NAME');
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                dbms_output.put_line('Error1 Nodata--->' || SQLERRM);
            WHEN OTHERS THEN
                dbms_output.put_line('Error1 others--->' || SQLERRM);
        END;
        l_sql := 'ALTER PLUGGABLE DATABASE APPLICATION ' || l_app_name || ' END
UPGRADE ' ;
        dbms_output.put_line('l_sql: ' || l_sql);

        EXECUTE IMMEDIATE l_sql;

    EXCEPTION
```

```
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error --->' || SQLERRM);
END;
/

SET ERRORLOGGING OFF
SPOOL OFF
```

End Application upgrade

This script runs the invalid script by connecting to the common user in approot outside the upgrade.

Syntax

```
/* Script for Shared Application + Shared Data */
SET VERIFY ON
SET HEAD ON
SET FEEDBACK 1
SET ARRAY 1
SET LINESIZE 10000
SET PAGESIZE 50000
SET LONG 10000
SET ECHO ON
SET TRIMSPPOOL ON
SET COLSEP ';'
SET SERVEROUT OFF
clear screen
SPOOL ON
SET SQLBLANKLINES ON
SET SERVEROUTPUT ON
SET ERRORLOGGING ON
SET ECHO ON
prompt Welcome to Application Upgrade Invalids Recompilation
SPOOL "&SPOOL_PATH"

DECLARE
    inval_cnt      NUMBER := 0;
    l_object_name   VARCHAR2(240);
BEGIN
    WHILE inval_cnt < 3 LOOP
        --SCRIPT
        FOR J IN (Select 'alter ' || object_type || ' ' || object_name ||'
compile' invalidobject1,
                    object_name
                    FROM user_objects
                    WHERE status = 'INVALID'
                    AND created_appid IS NULL
                    AND object_type IN
('VIEW','SYNONYM','PROCEDURE','FUNCTION','PACKAGE','TRIGGER','MATERIALIZED
VIEW'))
        LOOP
            BEGIN
                l_object_name := j.object_name;
                dbms_output.put_line(chr(10));
                EXECUTE IMMEDIATE J.invalidobject1;
            EXCEPTION
                WHEN OTHERS THEN
                    dbms_output.put_line('failed for -->' || l_object_name);
            END;
        END LOOP;
        inval_cnt := inval_cnt + 1;
    END LOOP;
```

```

        END LOOP;
    EXCEPTION
        WHEN OTHERS THEN
            dbms_output.put_line('FAILED FOR -->' || l_object_name);
    END;
/
DECLARE
    inval_cnt1          NUMBER := 0;
    l_object_name        VARCHAR2(240);
BEGIN
    WHILE inval_cnt1 < 3 LOOP
        --SCRIPT
        FOR k IN (Select 'alter package ' || object_name || ' compile body'
invalidobject2,
                object_name
                FROM user_objects
                WHERE status = 'INVALID'
                AND created_appid IS NULL
                AND object_type IN ('PACKAGE BODY'))
        LOOP
            BEGIN
                l_object_name := k.object_name;
                dbms_output.put_line(chr(10));
                EXECUTE IMMEDIATE k.invalidobject2;
            EXCEPTION
                WHEN OTHERS THEN
                    dbms_output.put_line('FAILED FOR -->' || l_object_name);
            END;
        END LOOP;
        inval_cnt1 := inval_cnt1 + 1;
    END LOOP;
EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('FAILED FOR -->' || l_object_name);
END;
/
select count(*) From user_objects Where status = 'INVALID';

SET ERRORLOGGING OFF
SPOOL OFF

```

Start Application upgrade

This script is run to initiate another application upgrade for object conversion.

Syntax

```
SET VERIFY ON
SET HEAD ON
SET FEEDBACK 1
SET ARRAY 1
SET LINESIZE 10000
SET PAGESIZE 50000
SET LONG 10000
SET ECHO ON
SET TRIMSPPOOL ON
SET COLSEP ';'
SET SERVEROUT OFF
clear screen
SPOOL ON
SET SQLBLANKLINES ON
SET SERVEROUTPUT ON
SET ERRORLOGGING ON
SET ECHO ON
prompt Welcome to Application Upgrade initiation
SPOOL "&SPOOL_PATH"

DECLARE
    l_app_name      VARCHAR2(128);
    l_app_currver   VARCHAR2(30);
    l_sql           VARCHAR2(256);
BEGIN

    BEGIN
        SELECT app_name
            INTO l_app_name
            FROM dba_applications
            WHERE app_implicit <> 'Y'
            AND app_name = (SELECT param_val FROM cstb_param WHERE Param_name =
'MULTI_TENANT_APP_NAME');
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            dbms_output.put_line('Error1 Nodata--->' || SQLERRM);
        WHEN OTHERS THEN
            dbms_output.put_line('Error1 others--->' || SQLERRM);
    END;

    BEGIN
        SELECT MAX(app_version)
            INTO l_app_currver
            FROM dba_app_versions
            WHERE app_name = l_app_name;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            dbms_output.put_line('Error2 Nodata--->' || SQLERRM);
```

```
        WHEN OTHERS THEN
            dbms_output.put_line('Error2 others--->'||SQLERRM);
    END;

    l_sql := 'ALTER PLUGGABLE DATABASE APPLICATION ' || l_app_name||' BEGIN
    UPGRADE '''|| l_app_currver || ''' TO '''|| '&P_APPLICATION_NEXTVER' ||'''';
    dbms_output.put_line('l_sql: ' || l_sql);
    EXECUTE IMMEDIATE l_sql;

    l_sql := 'ALTER SYSTEM SET DEFAULT_SHARING = NONE';
    dbms_output.put_line('l_sql: ' || l_sql);
    EXECUTE IMMEDIATE l_sql;

EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('Error --->'||SQLERRM);
END;
/

SET ERRORLOGGING OFF
SPOOL OFF
```

Application Root objects conversion for new objects

This script is used in converting the new DL objects during patch-set based on the deployment model of the application during installation.

Syntax

```
/*      Script      for      Shared      Object      Conversion      for      patch-set
*/
SET VERIFY ON
SET HEAD ON
SET FEEDBACK 1
SET ARRAY 1
SET LINESIZE 10000
SET PAGESIZE 50000
SET LONG 10000
SET ECHO ON
SET TRIMSPPOOL ON
SET COLSEP ';'
SET SERVEROUT OFF
clear screen
SPOOL ON
SET SQLBLANKLINES ON
SET SERVEROUTPUT ON
SET ERRORLOGGING ON
SET ECHO ON
prompt Welcome to Upgrade New object Conversion
SPOOL    "&SPOOL_PATH"

DECLARE
    l_count          NUMBER;
    l_app_deployment VARCHAR2(30);
BEGIN
    SELECT count(*)
        INTO l_count
        FROM user_objects
        WHERE sharing = 'NONE' --to get the new set of DL approot objects if any
        AND object_name IN
            (SELECT DISTINCT a.object_name
              FROM cstm_aproot_objects a
             WHERE sharing = 'DL'
               AND UPPER(object_type) = 'TABLE'
               AND EXISTS (SELECT 1
                           FROM user_objects b
                          WHERE b.object_name = a.object_name)
               AND EXISTS (SELECT 1
                           FROM cstm_aproot_functions_menu c
                          WHERE c.function_id = a.function_id
                             AND c.modifiable IN ('Y', 'S')));
    dbms_output.put_line('l_count:      ' || l_count);

    IF l_count > 0 THEN
```

```

dbms_output.put_line('New DL objects are available');
SELECT param_val
  INTO l_app_deployment
  FROM cstb_param
 WHERE param_name = 'MULTI_TENANT_DEPLOYMENT_MODEL';
dbms_output.put_line('l_app_deployment: '||l_app_deployment);

IF l_app_deployment IS NOT NULL AND l_app_deployment = 'SAUA' THEN
  UPDATE smtb_menu menu
    SET menu.approot_flg = 'Y'
  WHERE menu.function_id IN
    (SELECT function_id
      FROM cstm_approot_functions_menu
      WHERE modifiable = 'S'
     UNION
      SELECT summary_fn_id
      FROM cstm_approot_functions_menu
      WHERE modifiable = 'S'
      AND summary_fn_id IS NOT NULL) --SMS function id 'S'
    AND menu.approot_flg <> 'Y'; --excluding the already modified
approot function ids in menu.
ELSIF l_app_deployment IS NOT NULL AND l_app_deployment = 'SASDD' THEN
  UPDATE smtb_menu menu
    SET menu.approot_flg = 'Y'
  WHERE menu.function_id IN
    (SELECT function_id
      FROM cstm_approot_functions_menu
     UNION
      SELECT summary_fn_id
      FROM cstm_approot_functions_menu
      WHERE summary_fn_id IS NOT NULL)
    AND menu.approot_flg <> 'Y'; --excluding the already modified
approot function ids in menu.
ELSIF l_app_deployment IS NOT NULL AND l_app_deployment = 'SASDC' THEN
  /*Assumption new table cstm_approot_menu_custom_movedtopdb will be
available
      and is populated with the function ids which are moved to PDB as
part of custom deployment
      It has 2 columns FUNCTION_ID and SUMMARY_FN_ID*/

  UPDATE smtb_menu menu
    SET menu.approot_flg = 'Y'
  WHERE menu.function_id IN
    (SELECT function_id
      FROM cstm_approot_functions_menu
     UNION
      SELECT summary_fn_id
      FROM cstm_approot_functions_menu
      WHERE summary_fn_id IS NOT NULL)
    AND menu.function_id NOT IN --excluding the function ids moved
to PDB already.
    (SELECT function_id
      FROM cstm_approot_menu_movedtopdb
     UNION
      SELECT summary_fn_id
      FROM cstm_approot_menu_movedtopdb

```

```

        WHERE summary_fn_id IS NOT NULL)
        AND menu.approot_flg <> 'Y'; --excluding the already modified
approot function ids in menu.
    END IF;

    BEGIN
        FOR I IN (SELECT 'BEGIN ' || chr(10) ||
                        'DBMS_PDB.SET_DATA_LINKED('&P_APPROOT_USER''' ||
',''' ||
                        Object_Name || ''',' || Namespace || '); ' ||
chr(10) ||
                        'EXCEPTION ' || chr(10) ||
                        'WHEN OTHERS THEN ' || chr(10) ||
                        'DBMS_OUTPUT.PUT_LINE(''ERROR ->' || SQLERRM); ' ||
                        chr(10) || 'END;' sqlobject
        FROM user_objects
        WHERE sharing = 'NONE' --to get the new set of DL
approot objects if any
        AND object_name IN
        (SELECT DISTINCT a.object_name
        FROM cstm_approot_objects a
        WHERE sharing = 'DL'
        AND UPPER(object_type) = 'TABLE'
        AND EXISTS (SELECT 1
                     FROM user_objects b
                     WHERE b.object_name = a.object_name)
        AND EXISTS
        (SELECT 1
         FROM cstm_approot_functions_menu c
         WHERE c.function_id = a.function_id
         AND c.modifiable IN ('Y', 'S')))) LOOP
            DBMS_OUTPUT.PUT_LINE(chr(10));
            EXECUTE IMMEDIATE I.sqlobject;
            DBMS_OUTPUT.PUT_LINE(I.sqlobject);
        END LOOP;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Error --->' || SQLERRM);
    END;
    ELSE
        dbms_output.put_line('No new DL objects available');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('Error --->' || SQLERRM);
END;
/

SET ERRORLOGGING OFF
SPOOL OFF

```

Object_Conversion

Purpose

This script is used for converting the modified MDL objects when there is a re-creation [objects with Create or Replace command during creation] happens during patch-set

Syntax

```
SET VERIFY ON
SET HEAD ON
SET FEEDBACK 1
SET ARRAY 1
SET LINESIZE 10000
SET PAGESIZE 50000
SET LONG 10000
SET ECHO ON
SET TRIMSPPOOL ON
SET COLSEP ';'
SET SERVEROUT OFF
clear screen
SPOOL ON
SET SQLBLANKLINES ON
SET SERVEROUTPUT ON
SET ERRORLOGGING ON
SET ECHO ON
prompt Welcome to Upgrade object conversion
SPOOL "&SPOOL_PATH"
BEGIN
    FOR I IN (SELECT 'BEGIN ' || chr(10) ||
                    'DBMS_PDB.SET_METADATA_LINKED('&P_APPROOT_USER''' ||
                    ', ''' ||
                    Object_Name || ', ' || Namespace || '); ' || chr(10)
                    ||
                    'EXCEPTION ' || chr(10) || 'WHEN OTHERS then ' ||
                    chr(10) ||
                    'DBMS_OUTPUT.PUT_LINE(''ERROR ->' || SQLERRM); ' ||
                    chr(10) || 'END;' sqlobject
                FROM user_objects
                WHERE sharing = 'NONE'
                  AND object_type NOT IN ('INDEX', 'LOB', 'TABLE
PARTITION', 'SEQUENCE', 'JOB', 'MATERIALIZED VIEW', 'MATERIALIZED VIEW LOG')
                  AND application = 'Y'
                  AND (object_name, object_type) NOT IN (SELECT
object_name, object_type
                                                    FROM
cstm_approot_objects
                                                    WHERE function_id
= 'DYNAMIC'
                                                    AND sharing
= 'NONE'
                                                    )
                ) LOOP
        dbms_output.put_line(chr(10));
```

```
        EXECUTE IMMEDIATE I.sqlobject;  
        dbms_output.put_line(I.sqlobject);  
    END LOOP;  
EXCEPTION  
WHEN OTHERS THEN  
    dbms_output.put_line('Error --->'||SQLERRM);  
END;  
/  
SET ERRORLOGGING OFF  
SPOOL OFF
```

Invalids_Recompilation_Inside_Upgrade

Purpose

This script is used to initiate the recompilation of objects inside the application upgrade for sanity.

Syntax

```
/* Script for Shared Application + Shared Data */
SET VERIFY ON
SET HEAD ON
SET FEEDBACK 1
SET ARRAY 1
SET LINESIZE 10000
SET PAGESIZE 50000
SET LONG 10000
SET ECHO ON
SET TRIMSPPOOL ON
SET COLSEP ';'
SET SERVEROUT OFF
clear screen
SPPOOL ON
SET SQLBLANKLINES ON
SET SERVEROUTPUT ON
SET ERRORLOGGING ON
SET ECHO ON
prompt Welcome to Application Upgrade Invalids Recompilation
SPPOOL "&SPPOOL_PATH"

DECLARE
    inval_cnt      NUMBER := 0;
    l_object_name   VARCHAR2(240);
BEGIN
    WHILE inval_cnt < 3 LOOP
        --SCRIPT
        FOR J IN (Select 'alter ' || object_type || ' ' || object_name ||'
compile' invalidobject1,
                    object_name
                    FROM user_objects
                    WHERE status = 'INVALID'
                    AND created_appid IS NOT NULL
                    AND object_type IN
('VIEW','SYNONYM','PROCEDURE','FUNCTION','PACKAGE','TRIGGER','MATERIALIZED
VIEW'))
        LOOP
            BEGIN
                l_object_name := j.object_name;
                dbms_output.put_line(chr(10));
                EXECUTE IMMEDIATE J.invalidobject1;
            EXCEPTION
                WHEN OTHERS THEN
                    dbms_output.put_line('failed for -->' || l_object_name);
            END;
        END LOOP;
    END LOOP;
END;
```

```

        END LOOP;
        inval_cnt := inval_cnt + 1;
    END LOOP;
EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('FAILED FOR -->' || l_object_name);
END;
/
DECLARE
    inval_cnt1      NUMBER := 0;
    l_object_name    VARCHAR2(240);
BEGIN
    WHILE inval_cnt1 < 3 LOOP
        --SCRIPT
        FOR k IN (Select 'alter package ' || object_name || ' compile body'
invalidobject2,
                object_name
                FROM user_objects
                WHERE status = 'INVALID'
                AND created_appid IS NOT NULL
                AND object_type IN ('PACKAGE BODY'))
        LOOP
            BEGIN
                l_object_name := k.object_name;
                dbms_output.put_line(chr(10));
                EXECUTE IMMEDIATE k.invalidobject2;
            EXCEPTION
                WHEN OTHERS THEN
                    dbms_output.put_line('FAILED FOR -->' || l_object_name);
            END;
        END LOOP;
        inval_cnt1 := inval_cnt1 + 1;
    END LOOP;
EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('FAILED FOR -->' || l_object_name);
END;
/
select count(*) From user_objects Where status = 'INVALID';

SET ERRORLOGGING OFF
SPOOL OFF

```

End_Upgrade

Purpose

This script is run for ending an application upgrade for patch-set.

Syntax

```
/* Pre-requisites: Step 3 on Application associated pdb creation is completed
*/
SET VERIFY ON
SET HEAD ON
SET FEEDBACK 1
SET ARRAY 1
SET LINESIZE 10000
SET PAGESIZE 50000
SET LONG 10000
SET ECHO ON
SET TRIMSPPOOL ON
SET COLSEP ';'
SET SERVEROUT OFF
clear screen
SPOOL ON
SET SQLBLANKLINES ON
SET SERVEROUTPUT ON
SET ERRORLOGGING ON
SET ECHO ON
prompt Welcome to Application end Upgrade
SPOOL "&SPOOL_PATH"

DECLARE
    l_app_name VARCHAR2(128);
    l_sql VARCHAR2(256);
BEGIN
    BEGIN
        SELECT app_name
        INTO l_app_name
        FROM dba_applications
        WHERE app_implicit <> 'Y'
        AND app_name = (SELECT param_val FROM cstb_param WHERE param_name
= 'MULTI_TENANT_APP_NAME');
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            dbms_output.put_line('Error1 Nodata--->' || SQLERRM);
        WHEN OTHERS THEN
            dbms_output.put_line('Error1 others--->' || SQLERRM);
    END;
    l_sql := 'ALTER PLUGGABLE DATABASE APPLICATION ' || l_app_name || ' END
UPGRADE ';
    dbms_output.put_line('l_sql: ' || l_sql);

    EXECUTE IMMEDIATE l_sql;

EXCEPTION
```

```
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error --->' || SQLERRM);
END;
/

SET ERRORLOGGING OFF
SPOOL OFF
```

Invalids_Recompilation_Outside_Upgrade.sql

Purpose

This script is run for invalid script by connecting to the common user in approot outside the upgrade.

Syntax

```
/* Script for Shared Application + Shared Data */
SET VERIFY ON
SET HEAD ON
SET FEEDBACK 1
SET ARRAY 1
SET LINESIZE 10000
SET PAGESIZE 50000
SET LONG 10000
SET ECHO ON
SET TRIMSPPOOL ON
SET COLSEP ';'
SET SERVEROUT OFF
clear screen
SPOOL ON
SET SQLBLANKLINES ON
SET SERVEROUTPUT ON
SET ERRORLOGGING ON
SET ECHO ON
prompt Welcome to Application Upgrade Invalids Recompilation
SPOOL "&SPOOL_PATH"

DECLARE
    inval_cnt      NUMBER := 0;
    l_object_name   VARCHAR2(240);
BEGIN
    WHILE inval_cnt < 3 LOOP
        --SCRIPT
        FOR J IN (Select 'alter ' || object_type || ' ' || object_name ||'
compile' invalidobject1,
                    object_name
                    FROM user_objects
                    WHERE status = 'INVALID'
                    AND created_appid IS NULL
                    AND object_type IN
('VIEW','SYNONYM','PROCEDURE','FUNCTION','PACKAGE','TRIGGER','MATERIALIZED
VIEW'))
        LOOP
            BEGIN
                l_object_name := j.object_name;
                dbms_output.put_line(chr(10));
                EXECUTE IMMEDIATE J.invalidobject1;
            EXCEPTION
                WHEN OTHERS THEN
                    dbms_output.put_line('failed for -->' || l_object_name);
            END;
        END LOOP;
    END LOOP;
END;
```

```

        END LOOP;
        inval_cnt := inval_cnt + 1;
    END LOOP;
EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('FAILED FOR -->' || l_object_name);
END;
/
DECLARE
    inval_cnt1      NUMBER := 0;
    l_object_name    VARCHAR2(240);
BEGIN
    WHILE inval_cnt1 < 3 LOOP
        --SCRIPT
        FOR k IN (Select 'alter package ' || object_name || ' compile body'
invalidobject2,
                object_name
                FROM user_objects
                WHERE status = 'INVALID'
                AND created_appid IS NULL
                AND object_type IN ('PACKAGE BODY'))
        LOOP
            BEGIN
                l_object_name := k.object_name;
                dbms_output.put_line(chr(10));
                EXECUTE IMMEDIATE k.invalidobject2;
            EXCEPTION
                WHEN OTHERS THEN
                    dbms_output.put_line('FAILED FOR -->' || l_object_name);
            END;
        END LOOP;
        inval_cnt1 := inval_cnt1 + 1;
    END LOOP;
EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('FAILED FOR -->' || l_object_name);
END;
/
select count(*) From user_objects Where status = 'INVALID';

SET ERRORLOGGING OFF
SPOOL OFF

```

PDB_Sync

Purpose

This script is run to synchronize the PDBs with the latest application changes in the application root.

Syntax

```
SET VERIFY ON
SET HEAD ON
SET FEEDBACK 1
SET ARRAY 1
SET LINESIZE 10000
SET PAGESIZE 50000
SET LONG 10000
SET ECHO ON
SET TRIMSPOOL ON
SET COLSEP ';'
SET SERVEROUT OFF
clear screen
SPOOL ON
SET SQLBLANKLINES ON
SET SERVEROUTPUT ON
SET ERRORLOGGING ON
SET ECHO ON
prompt Welcome to Application PDB Sync
SPOOL "&SPOOL_PATH"

DECLARE
    l_app_name VARCHAR2(128);
    l_sql VARCHAR2(256);
BEGIN
    BEGIN
        SELECT app_name
            INTO l_app_name
            FROM dba_applications
            WHERE app_implicit <> 'Y';
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            dbms_output.put_line('Error1 Nodata--->' || SQLERRM);
        WHEN OTHERS THEN
            dbms_output.put_line('Error1 others--->' || SQLERRM);
    END;
    l_sql := 'ALTER PLUGGABLE DATABASE APPLICATION ' || l_app_name || ' SYNC ';
    dbms_output.put_line('l_sql: ' || l_sql);
    EXECUTE IMMEDIATE l_sql;

EXCEPTION
WHEN OTHERS THEN
    dbms_output.put_line('Error --->' || SQLERRM);
END;
/
```

```
SET ERRORLOGGING OFF  
SPOOL OFF
```

Index

A

Application Upgrade, [2](#)

P

Pre-Requisites, [1](#)