# Oracle Banking Trade Finance Multi-Tenant Deployment



Release 14.7.6.0.0 G32572-01 April 2025

ORACLE

Oracle Banking Trade Finance Multi-Tenant Deployment, Release 14.7.6.0.0

G32572-01

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	v
Audience	V
Documentation Accessibility	v
Critical Patches	V
Diversity and Inclusion	V
Conventions	vi
Related Resources	vi
Screenshot Disclaimer	vi

### 1 Oracle Multi-Tenant Architecture

1.1 Overview of the Multitenant Architecture		1-1	
	1.1.1	Container Database	1-1
	1.1.2	Application Root	1-2
	1.1.3	Seed PDB	1-2
	1.1.4	Application PDB	1-2
1.2	Appli	cation Maintenance	1-2
	1.2.1	Application Installation	1-3
	1.2.2	Application Upgrade	1-3

## 2 Proposed Deployment Model

2.1	Shared Application	2-1
2.2	Shared Application and User Authentication	2-2
2.3	Shared Application with Shared Data - Default	2-3
2.4	Shared Application with Shared Data - Custom	2-4

# 3 Deployment and Installation Steps

3.1 Creation of Application Template	3-2
3.1.1 Purpose	3-2
3.1.2 Steps to be followed	3-2
3.1.2.1 Application Template PDB Creation	3-3



3.1.2.2	Property file creation with Application Template PDB	3-3
3.1.2.3	Loading objects into the Application Template PDB	3-3
3.2 Creation of	of Application Root and Application Seed	3-3
3.2.1 Purp	oose	3-4
3.2.1.1	Application Root and Application Seed Creation	3-4
3.2.2 Step	os to be followed	3-4
3.3 Applicatio	n Maintenance and PDB creation	3-5
3.3.1 Purp	oose	3-5
3.3.2 Step	os for Manual Application Setup	3-5
3.3.2.1	Application Installation	3-6
3.3.2.2	Application Root Objects Conversion	3-6
3.3.2.3	Application Seed Sync with the Application Root	3-7
3.3.2.4	Creation of Application PDB	3-8
3.3.3 Step	os for Application Setup When Transaction Data Exists	3-8
3.3.3.1	Creation of Application PDB	3-9
3.3.3.2	Application Installation	3-14
3.3.3.3	Application Root Objects Conversion	3-14
3.3.3.4	Application PDB Sync with the Application Root	3-16
3.3.3.5	Application Seed Sync with the Application Root	3-16
3.4 Day Zero	Setup	3-17
3.5 EAR Crea	tion and Deployment	3-17

# 4 EAR Creation and Deployment

4.1	Approot Object Conversion: Shared Application	4-1
4.2	Approot Object Conversion: Shared Application and User Authentication	4-5
4.3	Approot Object Conversion: Shared Application and Shared Data – Default	4-9
4.4	Approot Object Conversion: Shared Application and Shared Data – Custom	4-13

# 5 Mandatory step before PDB/SEED Sync

## 6 Possible Issues / FAQ

## 7 Annexure

7.1	Default Approot Entities for Common Core	7-1
7.2	Default Approot Entities for OBTFM	7-2



# Preface

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

# Purpose

This manual is designed to help you quickly get acquainted with Multi-Tenant Deployment.

# 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 asso with an action, or terms defined in text.	
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

# **Related Resources**

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

- Oracle Banking Trade Finance Release Notes
- Oracle Banking Trade Finance Install & Upgrade

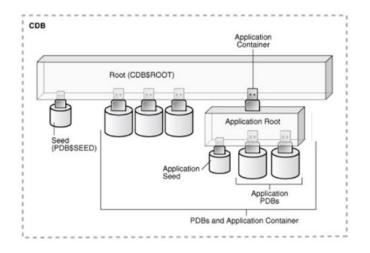
# Screenshot Disclaimer

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 Oracle Multi-Tenant Architecture

#### Figure 1-1 Multi-Tenant Architecture



- Overview of the Multitenant Architecture
- Application Maintenance Application maintenance refers to installing, uninstalling, upgrading, or patching an application.

# 1.1 Overview of the Multitenant Architecture

Container Database

The CDB is a collection of schemas, schema objects, and non-schema objects to which all PDBs belong.

- Application Root
- Seed PDB
- Application PDB

### 1.1.1 Container Database

The CDB is a collection of schemas, schema objects, and non-schema objects to which all PDBs belong.

Every CDB has one and only one root container named CDB\$ROOT. The root stores the system metadata required to manage PDBs. All PDBs belong to the root. The system container is the CDB root and all PDBs that belong to this root.

The CDB root does not store user data. Oracle recommends that you do not add common objects to the root or modify Oracle-supplied schemas in the root. However, you can create common users and roles for database administration. A common user with the necessary privileges can switch between containers.



Section Title

(Optional) Enter conceptual text here.

Example 1-1 Example Title

(Optional) Enter an example here.

## 1.1.2 Application Root

Consider an application root as an application-specific root container. It serves as a repository for a master definition of an application back end, including common data and metadata. To create an application root, connect to the CDB root and specify the AS APPLICATION CONTAINER clause in a CREATE PLUGGABLE DATABASE statement.

## 1.1.3 Seed PDB

Unlike a standard PDB, a seed PDB is not intended to support an application. Rather, the seed is a template for the creation of PDBs that support applications. To accelerate creation of application PDBs within an application container, you can create an application seed. An application container contains either zero or one application seed.

## 1.1.4 Application PDB

An application PDB belongs to exactly one application container. Unlike PDBs plugged in to the CDB root, application PDBs can share a master application definition within an application container. For example, a user\_details table in an application root might be a data-linked common object, which means it contains data accessible by all application PDBs plugged in to this root. PDBs that do not reside within the application container cannot access its application common objects.

# **1.2 Application Maintenance**

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

The basic steps for application maintenance are as follows:

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

• 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.

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

- Name of the application
- Application version number

# 1.2.2 Application Upgrade

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

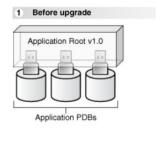
Typically, an upgrade changes the physical architecture of the application. For example, an upgrade might add new user accounts, tables, and packages, or alter the definitions of existing objects.

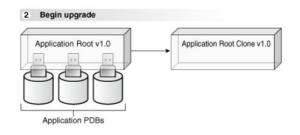
To upgrade the application, you must specify the following in the ALTER PLUGGABLE DATABASE APPLICATION statement:

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

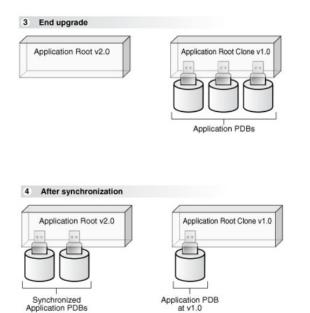
During an application upgrade, the application remains available. To make this availability possible, Oracle Database clones the application root.

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











#### Shared Application

In this model application would be deployed in an application container in 19C, Multiple front-end applications with URL is created per PDB.

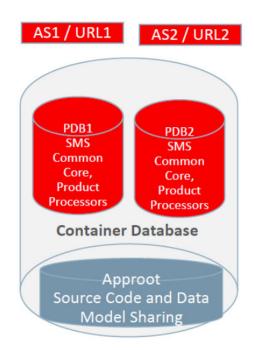
- Shared Application and User Authentication In this model application would be deployed in an application container in 19C, Single front end application and an URL.
- Shared Application with Shared Data Default This would be using Application Container in 19C, Single front end application and an URL. Sharing of Entities from Approot to individual PDBs.
- Shared Application with Shared Data Custom This would be using Application Container in 19C, Single front end application and an URL. Sharing of Entities from Approot to individual PDBs.

# 2.1 Shared Application

In this model application would be deployed in an application container in 19C, Multiple frontend applications with URL is created per PDB.

- Application would be deployed in an Application Container.
- Source code at Approot level shared with PDBs.
- Data Model at Approot level shared with PDBs.
- No sharing of data
- Multiple frontend application with URL per PDB (with common EAR file).



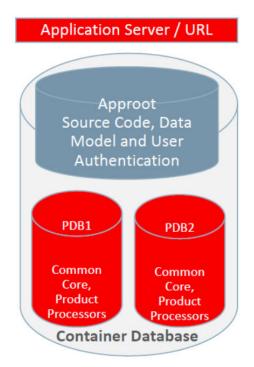


# 2.2 Shared Application and User Authentication

In this model application would be deployed in an application container in 19C, Single front end application and an URL.

- Application would be deployed in an Application Container.
- Source code at Approot level shared with PDBs.
- Data Model at Approot level shared with PDBs.
- Sharing of data related to User Authentication.
- Single Frontend Application and Single URL.





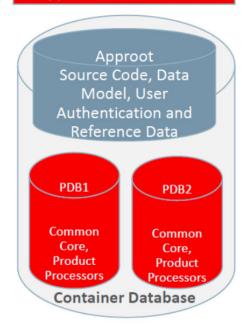
# 2.3 Shared Application with Shared Data - Default

This would be using Application Container in 19C, Single front end application and an URL. Sharing of Entities from Approot to individual PDBs.

- Application would be deployed in an Application Container.
- Source code at Approot level shared with PDBs.
- Data Model at Approot level shared with PDBs.
- Single Frontend Application and Single URL.
- Sharing of Entities/data like.
  - User Authentication, SMS Roles
  - Core Entities like Country, Currency, MIS Classes, UDFs
  - Chart of Account, Product, Account Class



#### Application Server / URL



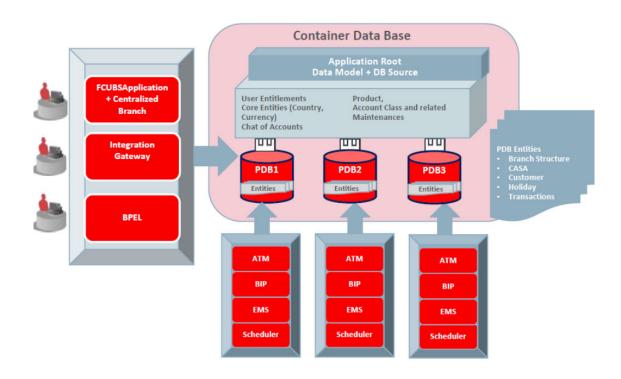
# 2.4 Shared Application with Shared Data - Custom

This would be using Application Container in 19C, Single front end application and an URL. Sharing of Entities from Approot to individual PDBs.

- Application would be deployed in an Application Container.
- Source code at Approot level shared with PDBs.
- Data Model at Approot level shared with PDBs.
- Single Frontend Application and Single URL.
- Sharing of Entities/data like.
  - User Authentication, SMS Roles
  - Core Entities like Country, Currency, MIS Classes, UDFs
  - Chart of Account, Product, Account Class
- User can opt-out the entities which are not required to be the candidates of approot and move to PDB.

Sample of components deployed in Shared Application and Shared Data model is given below:





#### Figure 2-1 Component Deployment Architecture

Application and Gateway will be common and single URL will be available for the application. ATM, BIP, EMS, Scheduler has to be configured separately for each PDBs.

# 3 Deployment and Installation Steps

As a pre-requisite, DB server has to be created with 18c database installed along with CDB setup.

Multi entity application root/PDB based application setup can be done by following below steps in sequential order, and detail of each steps explained as separate section subsequently.

- 1. Application Template PDB configuration
  - Application Template PDB is a normal PDB created under CDB to install the required DB objects for a product processor. This PDB will have a common schema and is used as a template for creating Application root through cloning.
- 2. Application root and Application Seed configuration
  - a. Application root
    - i. Application root is an application-specific root container and repositories for an application back end DB objects.
    - ii. Application root will be created through cloning from Application Template PDB.
  - b. Application Seed
    - i. Application seed is created to accelerate the creation of application PDBs within an application container.
    - ii. Application seed will be created from Application root through cloning and used as template to create one or more Application PDBs.
- 3. Application Installation
  - a. Application installation has to be done in the approot as version 1.0 with being user made explicit.
- 4. Application Root objects conversion
  - All the DB objects loaded in Application root will be converted as DATA LINK or METADATA LINK.
- 5. Application Seed Sync with the Application Root
  - a. Any changes deployed in Application Root will be available at Application PDB, if Application PDB sync with Application Root.

#### Note:

Application root should be synced with application seed always. It will help during the creation of new application PDBs. By this way the new application PDB created will have the all the patches applied.

- 6. Application PDB (entity) configuration from Application Seed
  - a. Application PDB is an associated PDB under Application Root. Application PDB will be created by clone from Application Seed.
- 7. Day Zero Setup



#### EAR Creation & Deployment

- Co-Deployment In case of Co-deployment all the product processor objects has to be loaded in the Application Template PDB, which will be cloned into Application Root and then subsequently cloned into Application Seed from Application Root inside an application container. Application Seed is used to accelerate the creation of application PDBs within an application container.
- Stand-alone Deployment– In case of stand-alone deployment, application set up steps has to be followed separately. Installation of multiple product processors can be done inside the same CDB with separate Application containers which has the template PDB, Application Seed and Application PDBs of its own. Same set of installation can be done inside a different CDB.

#### Note:

If .ear deployed in WebLogic server and SOA domain. RCU (Repository Configuration Utility) schemas should be created in a separate PDB. It should not be created within the application root container or in application PDBs.

- Creation of Application Template
- Creation of Application Root and Application Seed
- Application Maintenance and PDB creation
- Day Zero Setup
- EAR Creation and Deployment

# 3.1 Creation of Application Template

- Purpose
- Steps to be followed

## 3.1.1 Purpose

Application Template PDB is a normal PDB created under CDB to install the required DB objects for a product processor. This PDB will have a common schema and is used as a template for creating Application root through cloning.

## 3.1.2 Steps to be followed

Below steps to be followed to configure Application Template PDB:

- Application Template PDB Creation
- Property File Creation pointing to Application Template PDB
- Objects loading into the Application Template PDB.
- Application Template PDB Creation
- Property file creation with Application Template PDB
- Loading objects into the Application Template PDB



### 3.1.2.1 Application Template PDB Creation

- User has to login into CDB as a sys user.
- Application Template PDB has to be created under the CDB.
- This Application Template PDB will be kept as a gold copy and recommended to not to use for any other purpose.
- Application Template PDB can have one common schema which will be cloned to create further databases.

Below script will create the Application Template PDB with required grants under the CDB. DBA rights are required to perform this step.

Application Template PDB Creation

#### (Refer the Attachment panel of this document to view the script)

Input sample for the script:

CDB Schema User Name	Sys
CDB Schema Password	Sys
CDB Host	1.1.1.1
CDB Port	1524
CDB Name	FC142CDB
DB Mounted Path	/scratch/db1800dat
Template PDB Name	Templatepdb
Common User Name	CMNUSER
Common User Password	CMNUSER

### 3.1.2.2 Property file creation with Application Template PDB

- Existing installer will be used for the property file creation.
- Property file has to be created with Application Template PDB schema details. (Refer OBTF\_Installer\_Property\_File\_Creation document)

### 3.1.2.3 Loading objects into the Application Template PDB

Objects have to be loaded in the Application Template PDB using bat file [E.g.: SMSDBCompileRun.bat, TFDBCompileRun.bat] by silent installer for respective product processer.

Application Template PDB schema should be checked for sanity with zero invalids.

# 3.2 Creation of Application Root and Application Seed

Purpose



• Steps to be followed Below script will create the Application root and Application seed. DBA rights are required to perform this step.

# 3.2.1 Purpose

- Application Root
  - An application root shares some characteristics with the CDB root, because it can contain common objects, and some characteristics with a PDB, because it is created with the CREATE PLUGGABLE DATABASE statement.
- Application Seed
  - After Application Root creation, Application Seed to be created by clone from Application Root. Application seed to be synched with Application Root, whenever there is DB objects deployed in Application Root. i.e., Application seed will have latest DB references of Application Root. Application seed will be used as template to create (entity) Application PDBs.
  - An optional application PDB that serves as a template for creating other PDBs within an application container
- Application Root and Application Seed Creation

### 3.2.1.1 Application Root and Application Seed Creation

- Application Root
  - Application Root will be created from Application Template PDB through clone.
     Application Root will hold all the DB objects as single source repository. Initially, the database sources will be copied Application Template PDB. On subsequent patch set upgrade, the database sources will be deployed in Application Root using upgrade mode.
- Application Seed
  - After Application Root creation, Application Seed to be created by clone from Application Root. Application seed to be synched with Application Root, whenever there is DB objects deployed in Application Root. i.e., Application seed will have latest DB references of Application Root. Application seed will be used as template to create (entity) Application PDBs.

## 3.2.2 Steps to be followed

Below script will create the Application root and Application seed. DBA rights are required to perform this step.

#### Approot\_AppSeed\_Creation.sql

#### (Refer the Attachment panel of this document to view the script)

Input sample for the script:

CDB Schema User Name	Sys
CDB Schema Password	Sys
CDB Host	1.1.1.1



CDB Port	1524
CDB Name	FC142CDB
DB Mounted Path	/scratch/db1800dat
Template PDB Name	Templatepdb
Common User Name	CMNUSER
Common User Password	CMNUSER

# 3.3 Application Maintenance and PDB creation

- Purpose
- Steps for Manual Application Setup
- Steps for Application Setup When Transaction Data Exists

### 3.3.1 Purpose

- Application Maintenance:
  - An application installation is the initial creation of a master application definition. A typical installation creates user accounts, tables, and PL/SQL packages.
- Creation of Application PDB:
  - Application PDB (entity) to be created by clone from Application seed available under Application root. This is associated PDB under Application Root. Any DB sources changes deployed in Application Root, those changes to be synched with Application PDB, if required.
  - Later if new Application PDB to be created, new Application PDB will be created by clone from Application seed. Since Application seed will hold latest DB sources by syncing with Application Root.

# 3.3.2 Steps for Manual Application Setup

Below steps to be followed to configure Application Root PDB:

- Application Installation
- Application Root objects conversion
- Application Seed Sync with the Application Root
- Creation of Application PDB
- Application Installation Application installation has to be done in the approot as version 1.0 with being user made explicit.
- Application Root Objects Conversion
- Application Seed Sync with the Application Root
- Creation of Application PDB A PDB that is plugged in to an application container can be created from application seed through cloning.



### 3.3.2.1 Application Installation

Application installation has to be done in the approot as version 1.0 with being user made explicit.

This application name has to be used for further upgrade in case of object conversion and applying other patch set objects.

Below script will install the application in Application root. DBA rights are required to perform this step.

#### **Application Installation**

#### (Refer the Attachment Panel of this document to view the script)

Input sample for the script:

CDB Schema User Name	Sys
CDB Schema Password	Sys
CDB Host	1.1.1.1
CDB Port	1524
Application Root Name	Approot1
Common User Name	CMNUSER

### 3.3.2.2 Application Root Objects Conversion

- By default sharing type of all DB objects loaded in the Application Root will be none.
- A static table will hold the information of selected tables for which the sharing type is DATA LINK. Other tables will be treated as METADATA LINK.
- Sharing of object types such as INDEX, LOB, TABLE PARTITION, SEQUENCE, JOB, MATERIALIZED VIEW and DYNAMIC PACKAGES will remain as NONE.
- All other object types such as SYNONYM, VIEW, TRIGGER FUNCTION, PROCEDURE, and PACKAGE would be converted as METADATA LINK.

#### **Object Conversion**

- With the above sharing type considerations, DB object types will be converted as DATA LINK and METADATA LINK as part of this application root object conversion step.
- User has to connect to Application Root as common user and then apply changes in upgrade mode with the same application name used in step 3.
- This step will be done from the installer and user will have 4 options to do the conversion as,
- Shared Application
  - Shared Application and User Authentication
  - Shared Application and Shared Data Default



Shared Application and Shared Data – Custom

#### Note:

Application root will be created through cloning from Application Template PDB which will have only the static data. Transaction or maintenance related data will not be available in the Application root.

#### **Shared Application**

Here all the function Ids will be available as PDB function ids.

#### **Shared Application and User Authentication**

SMS function ids will be available in Approot and the remaining all function ids will be available as PDB function ids.

#### Shared Application and Shared Data – Default

Identified list of entities will be available in approot and sharing of entities from Approot to individual PDBs is applicable in this model.

#### Shared Application and Shared Data - Custom

Identified list of entities will be available in approot and sharing of entities from Approot to individual PDBs is applicable in this model.

Additionally, User can opt-out the entities which are not required to be the candidates of approot and those function ids will be moved to PDB.

The application name and type of deployment will be stored in CSTB\_PARAM table in approot.

PARAM_NAME	PARAM_VAL
MULTI_TENANT_APP_ NAME	OBTFM
MULTI_TENANT_DEPL OYMENT_MODEL	SA (or) SAUA (or) SASDD (or) SASDC

Object conversion is a one-time activity and if it is tried again, system will validate based on the availability of cstb\_param values.

### 3.3.2.3 Application Seed Sync with the Application Root

- In Application Root, post conversion of object type as DATA LINK and METADATA LINK, user need to sync Application Root with Application Seed.
- Post sync, characteristic of objects available in Application seed and Application PDBs will be same.
- Every patch set upgradation in Application Root,
  - User need to sync, Application Root with Application seed, to keep Application seed to hold the latest DB sources since Application seed will be used to create new PDBs further along.
- Below Scripts can also be used to execute this step. This step can be performed from common user.

Approot\_AppSeed\_Sync.sql



#### (Refer the Attachment panel of this document to view the script)

Input sample for the script:

Approot Schema Username	CMNUSER
Approot Schema Password	CMNUSER
Approot Host	1.1.1.1
Approot Port	1524
Application Root Name	Approot1
Application Name	FCUBS

### 3.3.2.4 Creation of Application PDB

A PDB that is plugged in to an application container can be created from application seed through cloning.

Below script will be used to create Application PDB from Application Seed. DBA rights are required to perform this step.

#### Application PDB Creation

#### (Refer the Attachment panel of this document to view the script)

Input sample for the script:

FC142CDB
Sys
Sys
1.1.1.1
1522
/scratch/db1800dat/FC142CDB/templatePDB/ users01.dbf
FCAPPROOT
FCAPPPDB1

# 3.3.3 Steps for Application Setup When Transaction Data Exists

If the maintenance/ transaction data import has to be carried out in the Application root and Application PDBs, below steps has to be followed in the sequential order:

- Creation of Application PDB
- Application Installation
- Application Root objects conversion
- Application PDB Sync with the Application Root
- Application Seed Sync with the Application Root



- Creation of Application PDB A PDB that is plugged in to an application container can be created from application seed through cloning.
- Application Installation Application installation has to be done in the approot as version 1.0 with being user made explicit.
- Application Root Objects Conversion
- Application PDB Sync with the Application Root This topic describes about application PDB sync with the application root.
- Application Seed Sync with the Application Root This topic describes about application seed sync with the application root.

### 3.3.3.1 Creation of Application PDB

A PDB that is plugged in to an application container can be created from application seed through cloning.

Below script will be used to create Application PDB from Application Seed. DBA rights are required to perform this step.

#### Application\_PDB\_Creation

#### (Refer the Attachment panel of this document to view the script)

Input sample for the script:

CDB Name	FC142CDB
CDB Schema User Name	Sys
CDB Schema Password	Sys
CDB HOST	1.1.1.1
CDB PORT	1522
CDB Mounted Path	/scratch/db1800dat
Application Root Name	FCAPPROOT
Application PDB Name	FCAPPPDB1

Note for Shared Application and User Authentication deployment model before object conversion:

SMS function ids will be available in Approot and the remaining all function ids will be available as PDB function ids.

- 1. Application root before object conversion will only have the static data.
- 2. If the data import has to be done to the application root schema, following steps 3 to 8 has to be carried out.
- 3. Triggers have to be disabled in the respective schemas before initiating the import.
- 4. Tables which are going to be available in the Application root as part of this model can be identified with the below query. (Total of around 21 tables)



```
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 b.object_type = 'TABLE')
AND EXISTS (SELECT 1
FROM cstm_approot_functions_menu c
WHERE c.function_id = a.function_id
AND c.modifiable = 'S');
```

- 5. The export data dump taken from the entities has to be imported into the application root schema only for these above set of tables.
- 6. For the PDB's, data from the entities can be directly imported into the respective application PDBs.
- 7. Once the import is completed, triggers have to be enabled again in the schemas.
- 8. After the data import, object conversion will be done from the installer.

#### Example

If there are two entity schemas available for India and Japan region and we have two export dump taken for these schemas.

#### Step 1: Importing data into the Application root schema

Import the dump taken from India entity schema for the given list of tables followed by the import of dump from Japan entity schema for the same list of tables.

If the table is already present in the application root schema, action should be allowed to just append the table data.

mpdp Approot_user/Approot_pwd@Approot_Schema			
bles= < Tables from the above script>			
ontent=DATA_ONLY			
IRECTORY=DUMP_FC144ENTITY1			
UMPFILE=FC144DEVPDB1 FULDUMP 210519.DMP			
LOGFILE=FC144DEVPDB1 FULDUMP APPROOT 260919 LOG.LOG			
EMAP SCHEMA=FC143ITR:FC14419CM1			
EMAP TABLESPACE=FC143ITR:FC14419CM1			
DATA_OPTIONS=skip_constraint_errors			
table exists action=append transform=OID:n			
Note:			
Remap Tablespace recheck in target schema before providing.			



#### Step 2: Importing data into the Application PDB schema

Once the first Application PDB is created from the application seed which will have only the data for static INCs, import the full dump taken from India entity schema.

Similarly, for the second application PDB import the full dump taken from Japan entity schema.

If the table is already present in the application PDB, action should be allowed to just append the table data.

impdp Approot\_user/Approot\_pwd@Approot\_Schema DIRECTORY=DUMP\_FC144ENTITY1 DUMPFILE=FC144DEVPDB1\_FULDUMP\_210519.DMP LOGFILE=FC144DEVPDB1\_FULDUMP\_PDB\_260919\_LOG.LOG REMAP\_SCHEMA=FC143ITR:FC14419CM1 REMAP\_TABLESPACE=FC143ITR:FC14419CM1 DATA\_OPTIONS=skip\_constraint\_errors table\_exists\_action=**append** transform=OID:n

# Note for Shared Application and Shared Data – Default deployment model before object conversion:

Identified list of entities will be available in approot and sharing of entities from Approot to individual PDBs is applicable in this model.

- 1. Application root before object conversion will only have the static data.
- 2. If the data import has to be done to the application root/ schema, following steps 3 to 8 has to be carried out.
- 3. Triggers have to be disabled in the respective schemas before initiating the import.
- 4. Tables which are going to be available in the Application root as part of this model can be identified with the below query. (Total of around 464 tables)

```
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 b.object_type = 'TABLE')
AND EXISTS (SELECT 1
FROM cstm_approot_functions_menu c
WHERE (c.function_id = a.function_id OR
a.function id IN ('STATIC', 'DYNAMIC')));
```



- 5. The export data dump taken from the entities has to be imported into the application root schema only for these above set of tables.
- 6. For the PDB's, data from the entities can be directly imported into the respective application PDBs.
- 7. Once the import is completed, triggers have to be enabled again in the schemas.
- 8. After the data import, object conversion will be done from the installer.

#### Example

If there are two entity schemas available for India and Japan region and we have two export dump taken for these schemas.

#### Step 1: Importing data into the Application root schema

Import the dump taken from India entity schema for the given list of tables followed by the import of dump from Japan entity schema for the same list of tables.

If the table is already present in the application root schema, action should be allowed to just append the table data.

impdp Approot\_user/Approot\_pwd@Approot\_Schema DIRECTORY=DUMP\_FC144ENTITY1 DUMPFILE=FC144DEVPDB1\_FULDUMP\_210519.DMP LOGFILE=FC144DEVPDB1\_FULDUMP\_PDB\_260919\_LOG.LOG REMAP\_SCHEMA=FC143ITR:FC14419CM1 REMAP\_TABLESPACE=FC143ITR:FC14419CM1/USERS DATA\_OPTIONS=skip\_constraint\_errors table\_exists\_action=**append** transform=OID:n

Remap Tablespace recheck in target schema before providing.

# Note for Shared Application and Shared Data – Default deployment model before object conversion:

Identified list of entities will be available in approot and sharing of entities from Approot to individual PDBs is applicable in this model.

- 1. Application root before object conversion will only have the static data.
- 2. If the data import has to be done to the application root/ schema, following steps 3 to 8 has to be carried out.
- 3. Triggers have to be disabled in the respective schemas before initiating the import.
- 4. Tables which are going to be available in the Application root as part of this model can be identified with the below query. (Total of around 464 tables)



```
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 b.object_type = 'TABLE')
AND EXISTS (SELECT 1
FROM cstm_approot_functions_menu c
WHERE (c.function_id = a.function_id OR
a.function_id IN ('STATIC', 'DYNAMIC')));
```

- 5. The export data dump taken from the entities has to be imported into the application root schema only for these above set of tables.
- 6. For the PDB's, data from the entities can be directly imported into the respective application PDBs.
- 7. Once the import is completed, triggers have to be enabled again in the schemas.
- 8. After the data import, object conversion will be done from the installer.

#### Example

If there are two entity schemas available for India and Japan region and we have two export dump taken for these schemas.

#### Step 1: Importing data into the Application root schema

Import the dump taken from India entity schema for the given list of tables followed by the import of dump from Japan entity schema for the same list of tables.

If the table is already present in the application root schema, action should be allowed to just append the table data.

tables= < <b>Tables</b> content= <b>DATA_O</b> DIRECTORY=DU DUMPFILE=FC14 LOGFILE=FC144 REMAP_SCHEM REMAP_TABLES DATA_OPTIONS=	impdp Approot_user/Approot_pwd@Approot_Schema tables= < <b>Tables from the above script</b> > content= <b>DATA_ONLY</b> DIRECTORY=DUMP_FC144ENTITY1 DUMPFILE=FC144DEVPDB1_FULDUMP_210519.DMP LOGFILE=FC144DEVPDB1_FULDUMP_APPROOT_260919_LOG.LOG REMAP_SCHEMA=FC143ITR:FC14419CM1 REMAP_TABLESPACE=FC143ITR:FC14419CM1 DATA_OPTIONS=skip_constraint_errors table_exists_action= <b>append</b> transform=OID:n		
	Note: Remap Tablespace recheck in target schema before providing.		

#### Step 2: Importing data into the Application PDB schema

Once the first Application PDB is created from the application seed which will have only the data for static INCs, import the full dump taken from India entity schema

Similarly, for the second application PDB import the full dump taken from Japan entity schema.

If the table is already present in the application PDB, action should be allowed to just append the table data.

impdp Approot\_user/Approot\_pwd@Approot\_Schema DIRECTORY=DUMP\_FC144ENTITY1 DUMPFILE=FC144DEVPDB1\_FULDUMP\_210519.DMP LOGFILE=FC144DEVPDB1\_FULDUMP\_PDB\_260919\_LOG.LOG REMAP\_SCHEMA=FC143ITR:FC14419CM1 REMAP\_TABLESPACE=FC143ITR:FC14419CM1 DATA\_OPTIONS=skip\_constraint\_errors table\_exists\_action=**append** transform=OID:n



### 3.3.3.2 Application Installation

Application installation has to be done in the approot as version 1.0 with being user made explicit.

This application name has to be used for further upgrade in case of object conversion and applying other patch set objects.

Below script will install the application in Application root. DBA rights are required to perform this step.

#### Application\_Installation

#### (Refer the Attachment Panel of this document to view the script)

Input sample for the script:

CDB Schema User Name	Sys
CDB Schema Password	Sys
CDB HOST	1.1.1.1
CDB PORT	1524
Application Root Name	Approot1
Application PDB Name	FCAPPPDB1
Common User Name	CMNUSER

### 3.3.3.3 Application Root Objects Conversion

• By default sharing type of all DB objects loaded in the Application Root will be none.



- A static table will hold the information of selected tables for which the sharing type is DATA LINK. Other tables will be treated as METADATA LINK
- Sharing of object types such as INDEX, LOB, TABLE PARTITION, SEQUENCE, JOB, MATERIALIZED VIEW and DYNAMIC PACKAGES will remain as NONE.
- All other object types such as SYNONYM, VIEW, TRIGGER FUNCTION, PROCEDURE, and PACKAGE would be converted as METADATA LINK.

#### **Object Conversion**

- With the above sharing type considerations, DB object types will be converted as DATA LINK and METADATA LINK as part of this application root object conversion step.
- User has to connect to Application Root as common user and then apply changes in upgrade mode with the same application name used in step 3.
- This step will be done from the installer and user will have 4 options to do the conversion as,
  - Shared Application
  - Shared Application and User Authentication
  - Shared Application and Shared Data Default
  - Shared Application and Shared Data Custom

#### Note:

Application root will be created through cloning from Application Template PDB which will have only the static data. Transaction or maintenance related data will not be available in the Application root.

#### Shared Application

Here all the function Ids will be available as PDB function ids.

#### Shared Application and User Authentication

#### Shared Application and Shared Data - Default

SMS function ids will be available in Approot and the remaining all function ids will be available as PDB function ids.

#### Shared Application and Shared Data - Default

Identified list of entities will be available in approot and sharing of entities from Approot to individual PDBs is applicable in this model.

#### Shared Application and Shared Data – Custom

Identified list of entities will be available in approot and sharing of entities from Approot to individual PDBs is applicable in this model.

Additionally, User can opt-out the entities which are not required to be the candidates of approot and those function ids will be moved to PDB.

The application name and type of deployment will be stored in CSTB\_PARAM table in approot.



PARAM_NAME	PARAM_VAL
MULTI_TENANT_APP_NAM E	OBTFM
MULTI_TENANT_DEPLOYM ENT_MODEL	SA (or) SAUA (or) SASDD (or) SASDC

Object conversion is a one-time activity and if it is tried again, system will validate based on the availability of cstb\_param values.

### 3.3.3.4 Application PDB Sync with the Application Root

This topic describes about application PDB sync with the application root.

- In Application Root, post conversion of object type as DATA LINK and METADATA LINK, user need to sync Application PDB with Application Root.
- Post sync, characteristic of objects available in Application root and Application PDBs will be the same.

Below Script can be used to execute this step. This step can be performed from common user.

#### Approot AppSeed\_Sync

#### (Refer the Attachment panel of this document to view the script)

Input sample for the script:

PDB Schema User Name	CMNUSER
PDB Schema Password	CMNUSER
PDB HOST	1.1.1.1
PDB PORT	1524
PDB Name	Approot1
Application Name	FCUBS

### 3.3.3.5 Application Seed Sync with the Application Root

This topic describes about application seed sync with the application root.

- In Application Root, post conversion of object type as DATA LINK and METADATA LINK, user need to sync Application Root with Application Seed.
- Post sync, characteristic of objects available in Application seed and Application root will be same.
- On every patch set upgrade in Application Root, user need to sync the application root with application seed, to keep Application seed hold the latest DB sources since Application seed will be used to create new PDBs further along.
   Below Scripts can also be used to execute this step. This step can be performed from common user.

#### Approot\_AppSeed\_Sync

#### (Refer the Attachment panel of this document to view the script)

Input sample for the script:



Approot Schema User Name	CMNUSER
Approot Schema Password	CMNUSER
Approot HOST	1.1.1.1
Approot PORT	1524
Approot Name	Approot1
Application Name	FCUBS

# 3.4 Day Zero Setup

Existing Installer can be used to do day zero setup with configuration mode as 'Application Root' and by selecting the radio button 'Utilities'. This step has to be executed for every entity PDB separately.

(Refer: OBTF\_DB\_Setup document)

# 3.5 EAR Creation and Deployment

- Existing installer file TFEarRun.bat can be used to create EAR.
- EAR deployment has to be deployed manually from console. During EAR deployment, JNDI connectivity details to be maintained for every Application PDB. JNDI details of Application PDB will be captured during Day Zero Setup.



# 4 EAR Creation and Deployment

- Existing installer file TFEarRun.bat can be used to create EAR.
- EAR deployment has to be deployed manually from console. During EAR deployment, JNDI connectivity details to be maintained for every Application PDB. JNDI details of Application PDB will be captured during Day Zero Setup.
- Approot Object Conversion: Shared Application This topic provides systematic instructions about approot object conversion - shared Application.
- Approot Object Conversion: Shared Application and User Authentication This topic provides systematic instructions about approot object conversion - shared application and user authentication.
- Approot Object Conversion: Shared Application and Shared Data Default This topic provides systematic instructions about approot object conversion - shared application and shared data – default.
- Approot Object Conversion: Shared Application and Shared Data Custom This topic provides systematic instructions about approot object conversion - shared application and shared data – default.

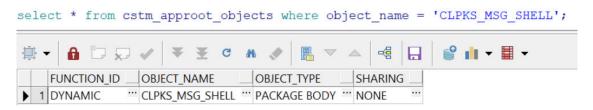
# 4.1 Approot Object Conversion: Shared Application

This topic provides systematic instructions about approot object conversion - shared Application.

**Shared Application** 

Kindly make sure all dynamic package exceptions should have an entry in "CSTM\_APPROOT\_OBJECTS" table.

Example: Only package body will be considered as exception and package will be converted to METADATA link.



For multi-tenant deployment setup using the installer with deployment model as 'Shared Application', follow the steps given below.

1. Double-click 'FCUBSInstaller.bat' batch file to launch Oracle FLEXCUBE Universal Installer. The following screen is displayed. Select Utilities option, configuration mode as "Application Root" and click 'Next' button.



Solution of the second	sal Installer 12.5.0.0.0	– 🗆 X
Oracle Banki	ng Installer	
Welcome To Oracle Univ	ersal Banking Installer	
Prerequisites		
<ul> <li>Oracle Databa</li> <li>JDK should be</li> </ul>	se should be Installed. e Installed.	
Please specify the JDK ar	nd Oracle Home path.	
JDK Path	C:/Program Files/Java/jdk1.8.0_181	frowse
Oracle Home Path	C:/app/client/pribalac/product/18.0.0/client_1	trowse
Configuration Mode	Application Root 💌	
Please select any one of	the below options:	
Utilities		
Exit	g	Back

2. Select 'Approot Object Conversion" in Utility Screen and click Next as shown below:

Soracle FLEXCUBE Universal Installer 12.5.0.0.0	– 🗆 X
Oracle Banking Installer	
Select an Utility:	
Approot Object Conversion	
🔾 Day Zero Setup	
O User Creation	
Reports DSN Entries	
O Entity Details	
SMS DSN Entries	
Switch Monitor Installation	
O ENV Property file operations	
O Block Chain	
Exit	Back Next

- 3. In the Approot Object Conversion screen, Enter Application Name and the Application root schema details where the conversion has to be applied and click on 'Test Connection'.
- 4. Once the connection is successful, '**Finish**' button will be enabled.
- 5. User has to select the option 'Shared Application' and click on the 'Finish' button to complete object conversion.

Oracle FLEXCUBE Universal Installer 12			
Oracle Banking Insta	aller		INSTALLER
Object conversion for Application ro	ot		
Enter Application Name FCUBS			
Provide Application root Schema deta	ils		
Name	Value		
Username	installer		
Password	•••••		
Service Name	testdb		
IP Address	10.10.10.10		
Port	1521		
TNS Connect Descriptor	testdb	Test Connection	
Select Option for conversion  Shared Application			
O Shared Application and User A	uthentication		
Shared Application and Shared	i Data - Default		
Shared Application and Shared	d Data - Custom		
		Des	
Exit Log		Bac	k Finish

6. Execution will take few minutes and post completion, a dialog box displays '**Compilation Success**' message in the front end.

Oracle Banking Inst		INSTALLER
Object conversion for Application	root	
Enter Application Name FCUBS		
Dravide Application root Cohema de	taile	
Provide Application root Schema de	Value	
Username	HUBUSER	
Password		
Service Name	F Message	×
IP Address		
Port	Execution is in progress.Please do not	t close the session
TNS Connect Descriptor	F	
Select Option for conversion		
Shared Application		
Shared Application and User	Authentication	
Shared Application and Shared Application	ed Data - Default	
Shared Application and Shar	eu Data - Custom	

7. This completes the setup and user can click on **Exit** to close the session.

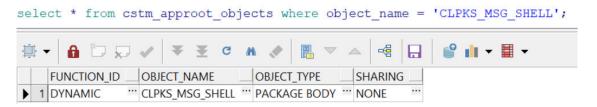
# 4.2 Approot Object Conversion: Shared Application and User Authentication

This topic provides systematic instructions about approot object conversion - shared application and user authentication.

#### Shared Application and User Authentication

Kindly make sure all dynamic package exceptions should have an entry in "CSTM APPROOT OBJECTS" table.

Example: Only package body will be considered as exception and package will be converted to METADATA link



For multi-tenant deployment setup using the installer with deployment model as 'Shared Application and User Authentication', follow the steps given below.



1. Double-click 'FCUBSInstaller.bat' batch file to launch Oracle FLEXCUBE Universal Installer. The following screen is displayed. Select Utilities option, configuration mode as "Application Root" and click 'Next' button.

Gracle FLEXCUBE Univer	sal Installer 12.5.0.0.0	– 🗆 X
Oracle Banki	ng Installer	
Welcome To Oracle Univ	ersal Banking Installer	
Prerequisites		
<ul> <li>Oracle Databa</li> <li>JDK should be</li> </ul>	se should be installed. e installed.	
Please specify the JDK a	nd Oracle Home path.	
JDK Path	C:/Program Files/Java/jdk1.8.0_181 Browse	
Oracle Home Path	C:/app/client/pribalac/product/18.0.0/client_1 Browse	
Configuration Mode	Application Root	
Please select any one of	the below options:	
Property File creation		
Utilities		
Exit	gBa	Next

2. Select 'Approot object Conversion" in Utility Screen and click Next as shown below:

Oracle FLEXCUBE Universal Installer 12.5.0.0.0	– 🗆 X
Oracle Banking Installer	
Select an Utility:	
Approot Object Conversion	
Day Zero Setup	
○ User Creation	
Reports DSN Entries	
O Entity Details	
SMS DSN Entries	
Switch Monitor Installation	
ENV Property file operations	
O Block Chain	
Exit Log	Back Next

- 3. In the Approot object conversion screen, **Enter Application Name** and the Application Root schema details where the conversion has to be applied and click on '**Test Connection**'.
- 4. Once the Connection is successful, 'Finish' button will be enabled.
- 5. User has to select the option 'Shared Application and User Authentication' and click on the 'Finish' button to complete object conversion.

Oracle FLEXCUBE Universal Installer 12 Oracle Banking Instal Object conversion for Application ro Enter Application Name FCUBS Provide Application root Schema deta	aller not		
Name	Value		
Username	installer		
Password	•••••		
Service Name	testdb		
IP Address	10.10.10		
Port	1521		
TNS Connect Descriptor	testdb	Test Connection	
Select Option for conversion           Shared Application           Shared Application and User A           Shared Application and Shared           Shared Application and Shared           Shared Application and Shared	d Data - Default		
Exit Log		Ba	ck Finish

6. Execution will take few minutes and post completion, a dialog box displays '**Compilation Success**' message in the front end.

Enter Application Name FCUBS		
Provide Application root Schema deta	ills	
Name	Value	
Username	HUBUSER	
Password	******	
Service Name	F Message	×
IP Address	Execution is in progress.Please do not close th	a seasion
Port	1 Execution is in progress. Please do not close un	e session
TNS Connect Descriptor	F	
Select Option for conversion		
C at an a total to a total tot		
Shared Application		
<ul> <li>Shared Application</li> <li>Shared Application and User A</li> </ul>	uthentication	
<ul> <li>Shared Application and User A</li> <li>Shared Application and Shared</li> </ul>	d Data - Default	
Shared Application and User A	d Data - Default	

7. This completes the setup and user can click on **Exit** to close the session.

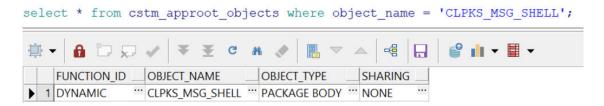
# 4.3 Approot Object Conversion: Shared Application and Shared Data – Default

This topic provides systematic instructions about approot object conversion - shared application and shared data – default.

#### Shared Application and User Authentication - Default

Kindly make sure all dynamic package exceptions should have an entry in "CSTM\_APPROOT\_OBJECTS" table.

Example: Only package body will be considered as exception and package will be converted to METADATA link



For multi-tenant deployment setup using the installer with deployment model as 'Shared Application and Shared Data - Default', follow the steps given below.



1. Double-click 'FCUBSInstaller.bat' batch file to launch Oracle FLEXCUBE Universal Installer. The following screen is displayed. Select Utilities option, configuration mode as "Application Root" and click 'Next' button.

Gracle FLEXCUBE Unive	ersal Installer 12.5.0.0.0	– 🗆 X
Oracle Banki	ing Installer	
Welcome To Oracle Uni	versal Banking Installer	
Prerequisites		
<ul> <li>Oracle Databa</li> <li>JDK should b</li> </ul>	ase should be Installed. e Installed.	
Please specify the JDK a	and Oracle Home path.	
JDK Path	C:/Program Files/Java/jdk1.8.0_181 Browse	
Oracle Home Path	C:/app/client/pribalac/product/18.0.0/client_1 Browse	
Configuration Mode	Application Root	
Please select any one o	f the below options:	
<ul> <li>Property File creation</li> <li>Utilities</li> </ul>		
Cumes		
Exit		Next

2. Select 'Approot object Conversion" in Utility Screen and click Next as shown below:

S Oracle FLEXCUBE Universal Installer 12.5.0.0.0	– 🗆 X
Oracle Banking Installer	
Select an Utility:	
Approot Object Conversion	
Day Zero Setup	
User Creation	
Reports DSN Entries	
<ul> <li>Entity Details</li> </ul>	
SMS DSN Entries	
Switch Monitor Installation	
O ENV Property file operations	
O Block Chain	
Exit Log	Back Next

- 3. In the Approot object conversion screen, **Enter Application Name** and the Application Root schema details where the conversion has to be applied and click on '**Test Connection**'.
- 4. Once the Connection is successful, 'Finish' button will be enabled.
- 5. User has to select the option 'Shared Application and Shared Data Default' and click on the 'Finish' button to complete object conversion.

Oracle FLEXCUBE Universal Installer 12.5	.0.0.0		- 🗆 🗙
Oracle Banking Instal	ler		
Object conversion for Application root			
Enter Application Name FCUBS			
Provide Application root Schema details	1		
Name	Value		
Username	installer		
Password	•••••		
Service Name	testdb		
IP Address	10.10.10.10		
Port	1521		
TNS Connect Descriptor	testdb	Test Connection	
Select Option for conversion			
Shared Application			
Shared Application and User Aut	hentication		
Shared Application and Shared I	)ata - Default		
Shared Application and Shared I	ata - Custom		
Exit Log		Bac	k Finish

6. Execution will take few minutes and post completion, a dialog box displays '**Compilation Success**' message in the front end.

Enter Application Name FCUBS		
Provide Application root Schema de Name	Value	
Username	HUBUSER	
Password		
Service Name	F Message	×
IP Address	V (i) Execution is in progress.Please do not clo	as the exercise
Port	Execution is in progress.Please do not clo	se the session
TNS Connect Descriptor	F	
Select Option for conversion  Shared Application  Shared Application and User  Shared Application and Share Shared Application and Share	ed Data - Default	

7. This completes the setup and user can click on Exit to close the session.

# 4.4 Approot Object Conversion: Shared Application and Shared Data – Custom

This topic provides systematic instructions about approot object conversion - shared application and shared data – default.

#### Shared Application and Shared Data – Custom

Kindly make sure all dynamic package exceptions should have an entry in "CSTM\_APPROOT\_OBJECTS" table.

Example: Only package body will be considered as exception and package will be converted to METADATA link

se	le	ct *	fro	n cs	tm_aj	opro	oot_o	obje	cts (	where	ob	ject_	nam	e =	'CLPH	KS_M	SG_S	HELL'	;
Ţ	<b>∳</b> →	6	Ð	$\overline{\mathbf{v}}$	~	₹	¥	c #			$\bigtriangledown$		-		ſ	•		•	
		FUNC	TION_	D	OBJEC	T_N/	AME		OBJEC	T_TYPE		SHAF	RING						
•	1	DYNA	MIC		CLPKS	_MS	G_SHE	ELL	PACK	AGE BC	DY .	" NON	E						

For multi-tenant deployment setup using the installer with deployment model as 'Shared Application and Shared Data -Custom', follow the steps given below.

1. Double-click 'FCUBSInstaller.bat' batch file to launch Oracle FLEXCUBE Universal Installer. The following screen is displayed. Select Utilities option, configuration mode as "Application Root" and click 'Next' button.



Gracle FLEXCUBE Univer	sal Installer 12.5.0.0.0		– 🗆 X
Oracle Banki	ng Installer		
Welcome To Oracle Univ	ersal Banking Installer		
Prerequisites			
<ul> <li>Oracle Databas</li> <li>JDK should be</li> </ul>	se should be installed. Installed.		
Please specify the JDK ar	nd Oracle Home path.		
JDK Path	C:/Program Files/Java/jdk1.8.0_181	Browse	
Oracle Home Path	C:/app/client/pribalac/product/18.0.0/client_1	Browse	
Configuration Mode	Application Root 💌		
Please select any one of Property File creation Utilities	the below options:		
Exit	9	Back	Next

2. Select 'Approot object Conversion" in Utility Screen and click Next as shown below:

Oracle FLEXCUBE Universal Installer 12.5.0.0.0	– 🗆 X
Oracle Banking Installer	
Select an Utility:	
Approot Object Conversion	
Day Zero Setup	
○ User Creation	
O Reports DSN Entries	
O Entity Details	
SMS DSN Entries	
Switch Monitor Installation	
ENV Property file operations	
O Block Chain	
Exit	Back Next

- 3. In the Approot object conversion screen, **Enter Application Name** and the Application Root schema details where the conversion has to be applied and click on '**Test Connection**'.
- 4. Once the Connection is successful, 'Next' button will be enabled.
- 5. User has to select the option 'Shared Application and Shared Data Custom' and click on the 'Next' button to take through the steps of movement of function ids to PDB.

Name     Value       Username     installer       Password     •••••••       Service Name     testdb       IP Address     10.10.10.10       Port     1521       TNS Connect Descriptor     testdb       Test Connection       Select Option for conversion       Shared Application     Shared Application and User Authentication       Shared Application and Shared Data - Default     Image: Shared Application and Shared Data - Custom	Oracle Banking Insta Object conversion for Application ro Enter Application Name FCUBS Provide Application root Schema deta	bot	
Password       •••••••         Service Name       testdb         IP Address       10.10.10.10         Port       1521         TNS Connect Descriptor       testdb         Test Connection	Name	Value	
Service Name testdb PAddress 10.10.10.10 Port 1521 TNS Connect Descriptor testdb Test Connection Select Option for conversion Shared Application Shared Application and User Authentication Shared Application and Shared Data - Default	Username	installer	
IP Address       10.10.10.10         Port       1521         TNS Connect Descriptor       testdb         Test Connection	Password	•••••	
Port 1521 TNS Connect Descriptor testdb Select Option for conversion Shared Application Shared Application and User Authentication Shared Application and Shared Data - Default	Service Name	testdb	
TNS Connect Descriptor testdb Test Connection Select Option for conversion Shared Application Shared Application and User Authentication Shared Application and Shared Data - Default	IP Address	10.10.10.10	
Select Option for conversion     Shared Application     Shared Application and User Authentication     Shared Application and Shared Data - Default	Port	1521	
Select Option for conversion  Shared Application  Shared Application and User Authentication  Shared Application and Shared Data - Default	TNS Connect Descriptor	testdb	
	<ul> <li>Shared Application</li> <li>Shared Application and User A</li> <li>Shared Application and Shared</li> </ul>	d Data - Default	

- 6. In the Next Screen, user can opt-out the entities which are not required to be the candidates of approot and those function ids will be moved to PDB.
- 7. There will be two multi blocks available.
  - a. First multi block will list the details of function groups which are the Approot candidates.
  - **b.** Second multi block will list the function ids corresponding to each of the function group in the first block.
- 8. User can select more than one function group and the respective function ids will also be appended to the second multi block against the function group on click of 'View Details' button.



🛃 Oracle f	LEXCUBE Universal Instal	ler 12.5.0.0.0		– 🗆 X
Ora	cle Banking In	staller		
Moveme	ent of Application root fu	nction Ids to PDB		
	Fun	ction Group Description		
	Accounting and MIS			
	Bank Parameters			
	Customers			
	EMS			
	Common Entity			
			View Details	
	Function Group	Function Id	Function Description	Move to PDB
E	kit Log			Back Next

9. Second multi block will have the check box 'Move to PDB' against each function ID.

	cle Banking	ot function lds to PDB		INSTALLE	R
novem		Function Group Description			
	Accounting and MIS	6			
	Bank Parameters	n			
	Customers				
~	EMS				
~	Common Entity				
			View Details		
			View Details		
	Function Group	Function Id	Function Description	Move to PDB	
Comm	on Entity	CYDCDEFE	Currencies	<b>₽</b>	1-
Comm	on Entity	CYDCRATY	Currency Rate Types	~	
Comm	on Entity	ISDBICDE	BIC Codes		
Comm	on Entity	ISDBICPU	Bank Identifier Code Upload		
Comm	on Entity	ISDBKDPL	Bank Directory Plus		
Comm	on Entity	ISDCTMEX	Clearing Codes		
Comm	on Entity	ISDEBANP	BICPIUSIBAN		
	on Entity	ISDESBAN	IBAN Information		
Comm	on Entity	ISDIBEXC	IBAN Exclusion List		
	on Entity	ISDIBNPL	IBAN Plus		
Comm		ISDNTMEX	Clearing Networks		
Comm Comm	on Entity				

- **10.** Once the selection is completed, click on the '**Next**' button to move to the next screen where the complete list of function ids.
- **11.** The dependent function ids of the selected functions opted to move to PDB will be listed in the below section.

Pracle Banking Installer		STALLER
Function Ids applicable for movement to PDB		
Function Id	Function Description	
CYDCDEFE	Currencies	
CYDCRATY	Currency Rate Types	
ISDBICDE	BIC Codes	
Dependent Function Ids	Constitut Description	
Dependent Function Ids Function Id	Function Description	
	Function Description	
Function Id		
Function Id	Clearing Networks	
Function Id ISDNTMEX MIDGRPMT	Clearing Networks MIS Groups	
Function Id ISDNTMEX MIDGRPMT ISDEBANP	Clearing Networks MIS Groups BICPlusIBAN	
Function Id ISDNTMEX MIDGRPMT ISDEBANP ISDBICPU	Clearing Networks MIS Groups BICPIusIBAN Bank Identifier Code Upload	
Function Id ISDNTMEX MIDGRPMT ISDEBANP ISDBICPU MIDXCODE	Clearing Networks MIS Groups BICPIusIBAN Bank Identifier Code Upload Cost Codes	
Function Id ISDNTMEX MIDGRPMT ISDEBANP ISDBICPU MIDXCODE CYDCDEFE	Clearing Networks MIS Groups BICPIusIBAN Bank Identifier Code Upload Cost Codes Currencies	
Function Id ISDNTMEX MIDGRPMT ISDEBANP ISDBICPU MIDXCODE CYDCDEFE STDCNMNT	Clearing Networks MIS Groups BICPIusIBAN Bank Identifier Code Upload Cost Codes Currencies Country Codes	

- **12.** Object conversion can be completed by clicking on the **Finish** button.
- **13.** Execution will take few minutes and post completion, a dialog box displays '**Compilation Success**' message in the front end.

	Function Id	Function Descripti	ion
CYDCDEFE		Currencies	
CYDCRATY		Currency Rate Types	
ISDBICDE		BIC Codes	
Dependent Function Ids			
	(i) Execution is in progr	ess.Please do not close the session Description	n
ISDNTMEX	Execution is in prog		n
MIDGRPMT	Execution is in progr	OK	n
MIDGRPMT ISDEBANP	Execution is in progr		n
MIDGRPMT ISDEBANP ISDBICPU	Execution is in progr	Bank Identifier Code Upload	n
MIDGRPMT ISDEBANP ISDBICPU MIDXCODE	Execution is in progr	Bank Identifier Code Upload Cost Codes	n 
MIDGRPMT ISDEBANP ISDBICPU	Execution is in progr	Bank Identifier Code Upload	
MIDGRPMT ISDEBANP ISDBICPU MIDXCODE	Execution is in progr	Bank Identifier Code Upload Cost Codes	
MIDGRPMT ISDEBANP ISDBICPU MIDXCODE CYDCDEFE	Execution is in program	Bank Identifier Code Upload Cost Codes Currencies	n 
MIDGRPMT ISDEBANP ISDBICPU MIDXCODE CYDCDEFE STDCNMNT	Execution is in progr	OK Bank Identifier Code Upload Cost Codes Currencies Country Codes	n 



**14.** This completes the setup and user can click on **Exit** to close the session.



# Mandatory step before PDB/SEED Sync

This topic provides systematic instructions to mandatory step before PDB/SEED sync.

- 1. Login into the Application Entity PDB/SEED as sys user.
- Create log\_error table using Log\_Error\_Table.DDL followed by create function fn\_error\_handler.fnc

Log\_Error\_Table

fn\_error\_handler

3. Alter the DB Syncing error handling parameters.

Alter the DB Syncing error handling parameters.

ALTER DATABASE PROPERTY SET SYNC\_ERROR\_HANDLER = 'sys.fn\_error\_handler'; Below are the errors handled during sync in Application PDB / Entity PDB.

Oracle Docs		
Oracle Error	Cause	Action
ORA-24344	A sql/plsql compilation error occurred.	Return OCI_SUCCESS_WITH_INFO along with the error code.
ORA-06512	Backtrace message as the stack is unwound by unhandled exceptions.	Fix the problem causing the exception or write an exception handler for this condition. Or you may need to contact your application administrator or DBA.
ORA-65297	An operation was attempted that can only be performed outside an application action (install, uninstall, upgrade, or patch)	Perform the operation outside an application action.
ORA-65274	An operation was attempted that can only be performed in an application action (install, uninstall, upgrade, or patch).	Begin an application action.
ORA-00001	An UPDATE or INSERT statement attempted to insert a duplicate key. For Trusted Oracle configured in DBMS MAC mode, you may see this message if a duplicate entry exists at a different level.	Either remove the unique restriction or do not insert the key.
ORA-01430	An ALTER TABLE ADD statement specified the name of a column that is already in the table. All column names must be unique within a table.	Specify a unique name for the new column, then re-execute the statement.



Oracle Docs		
Oracle Error	Cause	Action
ORA-02264	The specified constraint name has to be unique.	Specify a unique constraint name for the constraint.
ORA-01434	A DROP SYNONYM statement specified a synonym that does not exist. Existing synonym names may be listed by querying the data dictionary.	Specify the name of an existing synonym in the DROP SYNONYM statement.
ORA-00955	An attempt was made to create a database object (such as a table, view, cluster, index, or synonym) that already exists. A user's database objects must have distinct names.	Enter a unique name for the database object or modify or drop the existing object so it can be reused.
ORA-06550	Usually a PL/SQL compilation error.	None
ORA-04063	Cause: Attempt to execute a stored procedure or use a view that has errors. For stored procedures, the problem could be syntax errors or references to other, non-existent procedures. For views, the problem could be a reference in the view's defining query to a non-existent table. Can also be a table which has references to non-existent or inaccessible types.	Fix the errors and/or create referenced objects as necessary.

# 6 Possible Issues / FAQ

This topic explains about possible issues / FAQ

#### Significance of the Application Name

The Application name provided at step 3 of the deployment will be used for any object modification like object conversion or patch-set application. Suggested name – FCUBS.

#### **Roles for the Common User**

The common user should have DBA role while application install or upgrade. It can be revoked once the application maintenance is completed.

#### Can there be multiple Applications available in case of Co- deployment?

- It is recommended to have a single application as the Common core units can be released as part of any product processor and if the object can be linked to only one application.
- Modification of the object belonging to one application cannot be modified in another application.

#### Day zero -set up in multi- tenant

- Day zero set up has be done for each of the PDBs created under the approot. The record insertion will be based on the sharing type of the object.
- If the sharing is METADATA LINK, then the record for the table will be inserted into PDB schema and if the sharing is DATA LINK, record insertion happens in the approot schema for that table.

#### PDB creation possible errors

Encountered the below error when the template PDB has read only schemas also available additionally.

```
ORA-65005: missing or invalid file name pattern for file - /scratch/db1800dat/BRVCDB19C/
SEEDFC142APPROOT/temp012018-01-08 16-05-42-077-PM.dbf
```

In such case, the FILE\_NAME\_CONVERT has to be provided with the full path till the temp file instead of the Approx and PDB path. Below link is referred to resolve this issue:

https://mosemp.us.oracle.com/epmos/faces/DocumentDisplay? \_afrLoop=188548547043444&id=1910646.1&displayIndex=1&\_afrWindowMode=0&\_adf.ctrlstate=2mboo8is2\_4

#### Sync failure with the PDB

 When synch with PDB fails, there is no definite solution available. Back up of the PDB can be taken before an upgrade and in case of synch failure; new PDB can be created and applied with the backup data.



 Generally, for multi-tenant the recommendation is that objects will be compiled in a normal schema to check the sanity and to make sure the Invalids are zero. Once that is successful, the compilation will be done in Multi-tenant database.

#### Sync with PDB at different time

- Once the application upgrade is completed in approot, it can be synched up to the PDB. If the PDBs are not synched at the same time, there will be a mismatch between the front end and backend objects.
- In such case when a single PDB is parked for synching afterwards, a separate front URL with backup EAR has to be created to point to the PDB schema.

#### During patch set deployment encountered below issues during sync into entity pdbs

#### ORA-21700: object does not exist or is marked for delete

#### ORA-44201: cursor needs to be reparsed

- Root cause can be traced in DBA\_APP\_ERRORS / DBA\_ERRORS oracle table.
- Execute below command in Approot and Pdb. Consolidate list and create a sql file.

SELECT INVALIDOBJECT1 FROM (SELECT 'alter '    REFERENCED_TYPE    ' '    REFERENCED_NAME    ' compile;' INVALIDOBJECT1, 1 INDX
FROM USER_DEPENDENCIES
WHERE NAME IN (SELECT object_name FROM user_objects WHERE status = 'INVALID')
AND TYPE = 'PACKAGE'
AND REFERENCED_TYPE IN ('PACKAGE', 'PACKAGE BODY') AND REFERENCED_NAME NOT IN ('STANDARD')
UNION SELECT 'alter '    OBJECT_TYPE    ' '    OBJECT_NAME    ' compile;' INVALIDOBJECT1,
2 INDX
FROM USER_OBJECTS WHERE OBJECT NAME IN
(SELECT object_name FROM user_objects WHERE status = 'INVALID') AND OBJECT_TYPE IN ('PACKAGE')
UNION
SELECT 'alter package '    OBJECT_NAME    ' compile body;' INVALIDOBJECT1, 3 INDX
FROM USER_OBJECTS
WHERE status = 'INVALID' AND OBJECT_TYPE IN ('PACKAGE BODY'))
ORDER BY INDX;

- Start the upgrade in approot.
- Drop the root cause objects.
- Create the root cause objects.
- Execute the sql file placed in a path.
- End upgrade



- Sync to Entity pdb.
- Verify the result using DBA\_APP\_ERRORS/ DBA\_ERRORS/USER\_OBJECTS status = 'INVALID'.



# 7 Annexure

- Default Approot Entities for Common Core
- Default Approot Entities for OBTFM

### 7.1 Default Approot Entities for Common Core

- 1. Core Entities/Maintenances
  - a. Country Code
  - b. Host Code & Timezone
  - c. Currency
  - d. Currency Rate types
  - e. Language Code
  - f. Rate Code Definition\*\*\*\*
- 2. SMS Entities/Maintenances
  - a. Entity Maintenance
  - b. User Master (SSD)
  - c. Role Master (SSD)
  - d. Function Maintenance
  - e. PII & Mask Maintenance
  - f. SSO Parameters
  - g. Hot Keys
  - h. Customer Access group
  - i. Department Maintenance
- 3. External Entities
  - a. External Chart of Accounts
  - b. External Transaction Codes
  - c. External Credit Approval
- 4. MIS and UDF
  - a. MIS Class & Codes
  - b. MIS Group
  - c. MIS Cost Codes
  - d. MIS Pool

<sup>&</sup>lt;sup>\*</sup> Islamic Entities wherever applicable

- e. UDF Definition
- f. UDF Function ID Mapping
- 5. Other Entities
  - a. BIC Codes and related maintenances
  - b. Process Definition
  - c. Amount Text
  - d. Media
  - e. Gateway Multi-Entity Function Ids\*\*
    - i. Upload Source
    - ii. External System
    - iii. Amendment Maintenance

### 7.2 Default Approot Entities for OBTFM

- 1. Core Entities/Maintenances
  - a. Chart of Accounts
  - b. Revaluation Setup
  - c. Transaction Codes
  - d. Currency Denominations
  - e. Customer Categories, Prefixes, Groups, Ownership, Relation
  - f. Issuer Codes
  - g. Overrides
- 2. Subsystem and Classes
  - a. Commission, Interest, Charge, Tax Scheme Class
  - b. Status Codes
  - c. ICCF Rule master\*\*
  - d. Tax Rule Master\*\*
  - e. Tax Categories, Tax Scheme, Tax Rate codes
  - f. Product UDF Mapping\*\*
  - g. Message Types, Media, Locations, SWIFT Tags
- 3. CASA and TD (Conventional and Islamic\*\*\*\*)
  - a. Account Class and Account Class Group
  - b. Interest and Charge Rule and Product
  - c. Interest and Charge SDEs

<sup>\*</sup> New function IDs

<sup>\*</sup> New function IDs

<sup>\*</sup> New function IDs

<sup>\*</sup> New function IDs

<sup>\*</sup> Islamic Entities wherever applicable

- d. Standing Instruction Product
- e. Structured Deposit Product
- f. PDC Product
- 4. Retail Lending (Conventional and Islamic\*\*\*\*)
  - a. Retail Lending Product and related Maintenances
  - b. Leasing Product
  - c. Mortgage Product
  - d. Microfinance Product
  - e. Collection Product
- 5. Teller
  - a. Retail Teller Product
  - b. Corporate Teller Product
  - c. Utility Payment Product
- 6. Trade (Conventional and Islamic\*\*\*\*)
  - a. Letter of Credit Product and Related Maintenances
  - b. Bills and Collection Product and Related Maintenances
- 7. Treasury (Conventional and Islamic\*\*\*\*)
  - a. Foreign Exchange Product and Related Maintenances
  - b. Money Market Product and Related Maintenances
  - c. Securities Repo Product and Related Maintenances
  - d. Corporate Deposit Product and Related Maintenances
  - e. Securities Product and Related Maintenances
  - f. Derivatives Product and Related Maintenances
- 8. Other Modules (Conventional and Islamic\*\*\*\*)
  - a. Asset Management Fund Product
  - b. Fixed Assets Product
  - c. Expense Processing Product
  - d. Intermediary Product
  - e. Retail Bills Product

<sup>&</sup>lt;sup>\*</sup> Islamic Entities wherever applicable

Islamic Entities wherever applicable

<sup>&</sup>lt;sup>1</sup> Islamic Entities wherever applicable

<sup>&</sup>lt;sup>\*</sup> Islamic Entities wherever applicable

### **Application Installation**

#### Purpose

It is used for application installation.

#### 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 Configuration
SPOOL "&SPOOL PATH"
/* Inputs are recieved */
/* Connect CDB as sys user */
accept P CDB USER Prompt 'Enter CDB Schema Username: '
accept P CDB PWD Prompt 'Enter CDB Schema Password: '
accept P CDB HOST Prompt 'Enter CDB Schema Host: '
accept P CDB PORT Prompt 'Enter CDB Schema Port: '
accept P APPROOT NAME Prompt 'Enter Application Root Name: '
accept P APPLICATION NAME Prompt 'Enter application name to be installed: '
accept P COMMON USER Prompt 'Enter Common User Name: '
/* Connecting to Application Root As SYSDBA*/
conn &P CDB USER/&P CDB PWD@(DESCRIPTION=(ADDRESS LIST=(ADDRESS=(PROTOCOL=TCP)
(HOST=&P CDB HOST) (PORT=&P CDB PORT))) (CONNECT DATA=(SERVER=DEDICATED)
(SERVICE NAME=&P APPROOT NAME))) as sysdba;
alter pluggable database application &P APPLICATION_NAME begin install '1.0';
    exec dbms pdb.set user explicit('&P COMMON USER');
alter pluggable database application &P APPLICATION NAME end install;
SET ERRORLOGGING OFF
SPOOL OFF
```



# Application\_PDB\_Creation

#### Purpose

It is used in the application PDB creation.

#### Syntax

/\* Pre-requisites: Step 2 on application root and application seed has to be 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 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 Configuration SPOOL "& SPOOL PATH" /\* Inputs are recieved \*/ /\* Connect Approot as sys user \*/ accept P\_CDB\_USER Prompt 'Enter CDB Username: ' accept P\_CDB\_PWD Prompt 'Enter CDB Password: ' accept P\_CDB\_HOST Prompt 'Enter CDB Host: ' accept P\_CDB\_PORT Prompt 'Enter CDB Port: ' accept P\_CDB\_NAME Prompt 'Enter CDB Schema Name: ' accept P DB MOUNTED PATH Prompt 'Enter Approot mounted path for approot application seed creation: [Eq: /scratch/db1800dat]' accept P APPROOT NAME Prompt 'Enter Application Root Name: ' accept P\_APPPDB\_NAME Prompt 'Please provide name for Application PDB Name -- Application Root associated PDB: ' /\* Connecting to Application Root As SYSDBA\*/ conn &P\_CDB\_USER/ &P CDB PWD@(DESCRIPTION=(ADDRESS LIST=(ADDRESS=(PROTOCOL=TCP) (HOST=&P\_CDB\_HOST)(PORT=&P\_CDB\_PORT))) (CONNECT DATA=(SERVER=DEDICATED)(SERVICE NAME=&P APPROOT NAME))) as sysdba; /\* Creating Application Associated PDB\*/ CREATE pluggable database &P APPPDB NAME FROM &P APPROOT NAME\$SEED file\_name\_convert=('&P\_DB\_MOUNTED\_PATH/&P\_CDB\_NAME/ SEED&P APPROOT NAME/,'&P DB MOUNTED PATH/&P APPROOT NAME/ &P APPPDB NAME/'); ALTER pluggable database &P APPPDB NAME OPEN; SET ERRORLOGGING OFF SPOOL OFF

## Application\_Template\_PDB\_Creation

#### Purpose

This script is used for application template PDB creation.

#### 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 Template PDB Configuration SPOOL "&SPOOL PATH" /\* CDB sys user name and password to be given \*/ accept P CDB USER Prompt 'Enter CDB Schema Username: [Eg: sys]' accept P CDB PWD Prompt 'Enter CDB Schema Password: [Eg: PASSWORD]' accept P CDB HOST Prompt 'Enter CDB Schema Host: [Eq: fcubs.in.oracle.com]' accept P CDB PORT Prompt 'Enter CDB Schema Port: [Eg: 1521]' accept P CDB NAME Prompt 'Enter CDB Service Name: [Eg: FCUBSCDB]' accept P DB MOUNTED PATH Prompt 'Enter CDB mounted path: [Eg: /scratch/ db1800dat]' accept P APP TEMPLATE PDB Prompt 'Enter Name for Application Template PDB to be created: [Eq: pdbfcubs]' accept P COMMON USER Prompt 'Enter Common Username to be created: [Eg: fcubs]' accept P COMMON USER PWD Prompt 'Enter Pwd for Common User : [Eg: fcubs]' accept P COMMON TSPACE Prompt 'Enter TableSpace Name : [Eg: fcubs]' /\* Connecting to CDB as sysdba \*/ CONN &P CDB USER/&P CDB PWD@&P CDB NAME AS sysdba; create pluggable database &P APP TEMPLATE PDB ADMIN USER sourceadmin IDENTIFIED BY sourceadmin file name convert=('pdbseed','&P APP TEMPLATE PDB'); alter pluggable database &P APP TEMPLATE PDB open; alter pluggable database &P APP TEMPLATE PDB save state; /\*connecting to template pdb as sysdba \*/ conn &P CDB USER/&P CDB PWD@(DESCRIPTION=(ADDRESS LIST=(ADDRESS=(PROTOCOL=TCP)

(HOST=&P CDB HOST) (PORT=&P CDB PORT))) (CONNECT DATA=(SERVER=DEDICATED)



```
(SERVICE NAME=&P APP TEMPLATE PDB))) as sysdba;
create tablespace &P COMMON USER datafile '&P DB MOUNTED PATH/&P CDB NAME/
&P APP TEMPLATE PDB/&P COMMON TSPACE..dbf' size 100M autoextend on next 10M
maxsize 30000M;
CREATE USER &P COMMON USER IDENTIFIED BY &P COMMON USER PWD default
tablespace &P COMMON USER quota unlimited on &P COMMON USER;
grant execute on dbms sql to &P COMMON USER;
grant execute on dbms lock to &P COMMON USER;
grant execute on dbms job to &P COMMON USER;
grant execute on dbms alert to &P COMMON USER;
grant execute on dbms refresh to &P COMMON USER;
grant execute on dbms pipe to &P COMMON USER;
grant execute on dbms_shared_pool to &P_COMMON USER;
grant execute on dbms application info to &P COMMON USER;
grant execute on utl file to &P COMMON USER;
grant select on v $process to &P COMMON USER;
grant select on v $session to &P COMMON USER;
grant select on v_$instance to &P_COMMON_USER;
grant select on v $timer to &P COMMON USER;
grant select on v $database to &P COMMON USER;
grant select on v $parameter to &P COMMON USER;
grant select on v_$nls_parameters to &P_COMMON_USER;
grant select on dba jobs running to &P COMMON USER;
grant create session to &P_COMMON_USER;
grant create synonym to &P COMMON USER;
grant create view to &P COMMON USER;
grant create sequence to &P COMMON USER;
grant create table to &P COMMON USER;
grant create procedure to &P COMMON USER;
grant create trigger to &P COMMON USER;
grant create type to &P COMMON USER;
grant create library to &P COMMON USER;
grant create database link to &P COMMON USER;
grant create any synonym to &P COMMON USER;
grant select on dba_jobs to &P_COMMON_USER;
grant create database link to &P COMMON USER;
grant create materialized view to &P COMMON USER;
grant execute on dbms ag to &P COMMON USER;
grant execute on dbms aqadm to &P COMMON USER;
grant execute on dbms_job to &P_COMMON_USER;
grant execute on dbms lock to &P COMMON USER;
grant execute on dbms pipe to &P COMMON USER;
grant execute on dbms refresh to &P COMMON USER;
grant execute on dbms rls to &P COMMON USER;
create public synonym dbms shared pool for sys.dbms shared pool;
grant execute on dbms shared pool to &P COMMON USER;
grant execute on dbms_sql to &P_COMMON_USER;
grant execute on utl file to &P COMMON USER;
grant select on SYS.TRANSPORT SET VIOLATIONS to &P COMMON USER;
grant create evaluation context to &P COMMON USER;
grant create rule to &P COMMON USER;
grant create job to &P COMMON USER;
grant create rule set to &P COMMON USER;
grant exp full database to &P COMMON USER;
```



```
grant alter tablespace to &P COMMON USER;
grant manage tablespace to &P COMMON USER;
grant execute on DBMS FILE TRANSFER to &P COMMON USER;
grant execute on SYS.DBMS TTS to &P COMMON USER;
grant execute on SYS.DBMS DATAPUMP to &P COMMON USER;
grant JAVAUSERPRIV to &P COMMON USER;
grant execute on dbms scheduler to &P COMMON USER;
create public synonym UTL RECOMP for sys.UTL RECOMP;
grant execute on UTL RECOMP to &P COMMON USER;
grant execute on DBMS MONITOR to &P COMMON USER;
grant select on dba directories to &P COMMON USER;
grant execute on DBMS CRYPTO to &P COMMON USER;
grant select on gv_$session to &P_COMMON_USER;
grant create any directory to &P COMMON USER;
grant select on SYS.DBA_SCHEDULER_RUNNING_JOBS to &P_COMMON_USER;
grant execute on sys.dbms redact to &P COMMON USER;
grant SELECT on sys.redaction_policies to &P COMMON USER;
grant SELECT on sys.redaction columns to &P COMMON USER;
grant SELECT on sys.redaction values for type full to &P COMMON USER;
grant create session, connect, resource to &P COMMON USER;
grant SELECT ON dba_applications to &P_COMMON_USER;
grant SELECT ON dba app versions to &P COMMON USER;
grant dba to &P COMMON USER;
SET ECHO OFF
clear screen
```

```
spool off
```

## Approot\_AppSeed\_Sync.sql

#### Purpose

Application Root - PDB Model Configuration

#### 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 Configuration
SPOOL "&SPOOL PATH"
/* Inputs are received */
accept P_APPROOT_USER Prompt 'Enter Approot Schema Username: '
accept P APPROOT PWD Prompt 'Enter Approot Schema Password: '
accept P APPROOT HOST Prompt 'Enter Approot Schema Host: '
accept P APPROOT PORT Prompt 'Enter Approot Schema Port: '
accept P APPROOT NAME Prompt 'Enter Application Root Name: '
accept P APPLICATION NAME Prompt
                                  'Enter application name to be upgraded for
object conversion: '
/*Connecting to Application seed*/
conn &P APPROOT USER/
&P_APPROOT_PWD@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
(HOST=&P APPROOT HOST) (PORT=&P APPROOT PORT))) (CONNECT DATA=(SERVER=DEDICATED)
(SERVICE NAME=&P APPROOT NAME$SEED)));
/*Synching object conversion to application seed */
alter pluggable database application &P APPLICATION NAME sync;
```

SET ERRORLOGGING OFF SPOOL OFF



### Approot\_PDB\_Sync

#### Purpose

Check the approot PDB Sync.

#### 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"
/* Inputs are received */
accept P_PDB_USER Prompt 'Enter PDB Schema Username: '
accept P PDB PWD Prompt 'Enter PDB Schema Password: '
accept P PDB HOST Prompt 'Enter PDB Schema Host: '
accept P_PDB_PORT Prompt 'Enter PDB Schema Port: '
accept P PDB NAME Prompt 'Enter the PDB name to be synched: '
accept P_APPLICATION_NAME Prompt 'Enter the application name: '
/*Connecting to pdb */
conn &P PDB USER/&P PDB PWD@ (DESCRIPTION=(ADDRESS LIST=(ADDRESS=(PROTOCOL=TCP)
(HOST=&P PDB HOST) (PORT=&P PDB PORT))) (CONNECT DATA=(SERVER=DEDICATED)
(SERVICE_NAME=&P_PDB_NAME)));
/*Synching the application with pdbs */
alter pluggable database application &P APPLICATION NAME sync;
SET ERRORLOGGING OFF
SPOOL OFF
```



## fn\_error\_handler

#### Purpose

This script is used as error handler.

#### **Syntax**



# Log\_error\_Table

#### Purpose

This script is used for log error table.

#### Syntax

CREATE TABLE log\_error (ERROR\_CODE VARCHAR2(100),LOG\_TIME DATE);



### Approot\_AppSeed\_Creation.sql

Approot appseed sync sql command.

Prerequisites

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 SOLBLANKLINES ON
SET SERVEROUTPUT ON
SET ERRORLOGGING ON
SET ECHO ON
prompt Welcome to Approot and ApprootSeed Configuration
SPOOL "&SPOOL PATH"
/* Inputs are recieved */
accept P CDB USER Prompt 'Enter CDB Schema Username: [Eg: sys]'
accept P CDB PWD Prompt 'Enter CDB Schema Password: [Eg: PASSWORD]'
accept P CDB HOST Prompt 'Enter CDB Schema Host: [Eg: fcubs.in.oracle.com]'
accept P CDB PORT Prompt 'Enter CDB Schema Port: [Eg: 1521]'
accept P CDB NAME Prompt 'Enter CDB Schema Name: [Eg: FCUBSCDB]'
accept P DB MOUNTED PATH Prompt 'Enter CDB mounted path for approot
application seed creation[Eg: /scratch/db1800dat] :'
accept P TEMPLATE PDB Prompt 'Enter Template PDB Name: [Eg: pdbfcubs]'
accept P APPROOT NAME Prompt 'Enter Approot Name: [Eq: appfcubs]'
accept P PDB TO APPPDB Prompt 'Please provide path for pdb to apppdb.sql:
[E.q:
C:\app\client\user\product\19.0.0\client 1\rdbms\admin\pdb to apppdb.sql]'
accept P COMMON USER Prompt 'Enter Common Username created in Template PDB:
[E.g: fcubs]'
/* Connecting to cdb
conn sys/FC142SYS18C@fc142cbd as sysdba */
CONN &P_CDB_USER/&P_CDB_PWD@&P_CDB_NAME AS sysdba;
/* Creating the Approot */
CREATE pluggable database &P APPROOT NAME AS application container FROM
&P TEMPLATE PDB file name convert=('&P TEMPLATE PDB','&P APPROOT NAME');
```

ALTER pluggable database &P APPROOT NAME open;

/\* Connecting to Approot as sysdba\*/
conn &P\_CDB\_USER/&P\_CDB\_PWD@(DESCRIPTION=(ADDRESS\_LIST=(ADDRESS=(PROTOCOL=TCP)
(HOST=&P\_CDB\_HOST)(PORT=&P\_CDB\_PORT)))(CONNECT\_DATA=(SERVER=DEDICATED)
(SERVICE NAME=&P APPROOT NAME))) as sysdba;

grant select on v\_\$session to &P\_COMMON\_USER container=all; grant create session to &P\_COMMON\_USER container=all; grant select on gv\_\$session to &P\_COMMON\_USER container=all; grant select on gv\_\$session to &P\_COMMON\_USER container=all; grant select on v\_\$database to &P\_COMMON\_USER container=all;

/\*Creating Application Seed Manually\*/
create pluggable database as seed from &P\_APPROOT\_NAME
file\_name\_convert=('&P\_DB\_MOUNTED\_PATH/&P\_CDB\_NAME/
&P\_APPROOT\_NAME/','&P\_DB\_MOUNTED\_PATH/&P\_CDB\_NAME/SEED&P\_APPROOT\_NAME/');

alter pluggable database &P\_APPROOT\_NAME\$SEED open; alter session set container = &P APPROOT NAME\$SEED;

@&P\_PDB\_TO\_APPPDB; select cause, type, message, status, action from pdb\_plug\_in\_violations;

SET ERRORLOGGING OFF SPOOL OFF

