

Multi-Tenant Deployment  
Oracle Banking Payments

Release 14.4.0.0.0

[May] [2020]



---

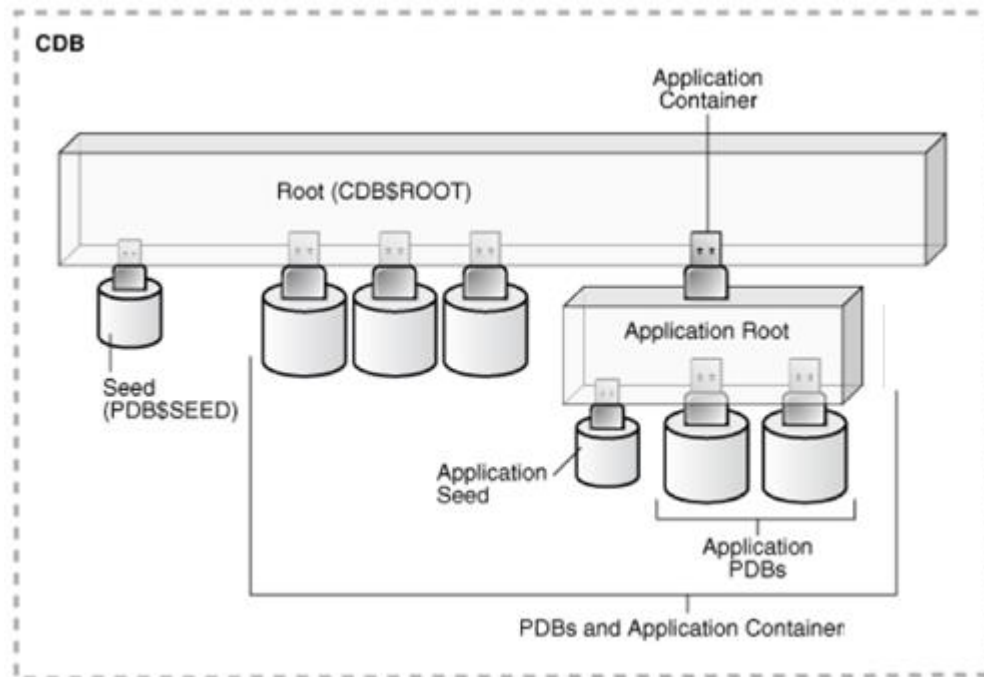
## Table of Contents

<b>1.</b>	<b>Oracle Multi-Tenant Architecture .....</b>	<b>1</b>
1.1	Overview of the Multitenant Architecture .....	1
1.1.1	Container Database .....	1
1.1.2	Application Root .....	1
1.1.3	Seed PDB .....	2
1.1.4	Application PDB .....	2
1.2	Application Maintenance .....	2
1.2.1	Application Installation .....	2
1.2.2	Application Upgrade .....	2
<b>2.</b>	<b>Proposed Deployment Model .....</b>	<b>4</b>
2.1	Shared Application .....	4
2.2	Shared Application and User Authentication .....	4
2.3	Shared Application with Shared Data - Default .....	5
2.4	Shared Application with Shared Data - Custom .....	5
<b>3.</b>	<b>Deployment and Installation Steps .....</b>	<b>7</b>
3.1	Creation of Application Template .....	8
3.1.1	Purpose .....	8
3.1.2	Steps to be followed .....	8
3.2	Creation of Application Root and Application Seed .....	9
3.2.1	Purpose .....	9
3.2.2	Steps to be followed .....	9
3.3	Creation of Application PDB .....	12
3.3.1	Purpose .....	12
3.3.2	Steps to be followed .....	12
3.4	Day Zero Setup .....	13
3.5	EAR Creation and Deployment .....	13
<b>4.</b>	<b>Annexure .....</b>	<b>14</b>
4.1	Default Approot Entities for Common Core .....	14
4.2	Default Approot Entities for Oracle Banking Payments .....	15

---

# 1. Oracle Multi-Tenant Architecture

## Multi-Tenant Architecture



---

## 1.1 Overview of the Multitenant Architecture

---

### 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.

---

### 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.

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

The basic steps for application maintenance are as follows:

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

### 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.

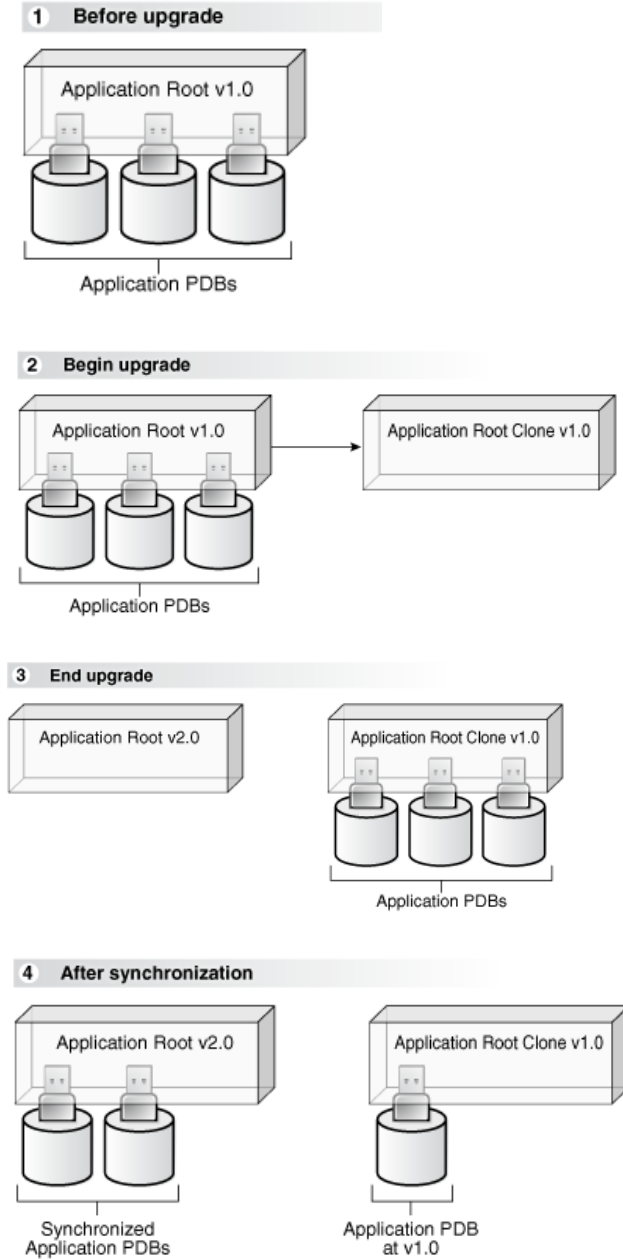
To upgrade the application, you must specify the following in the ALTER PLUGGABLE DATABASE APPLICATION statement:

- 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.

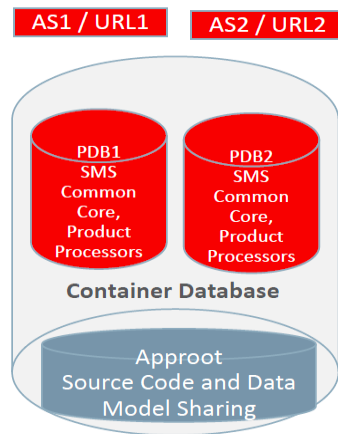
The following figure gives an overview of the application upgrade process.



## 2. Proposed Deployment Model

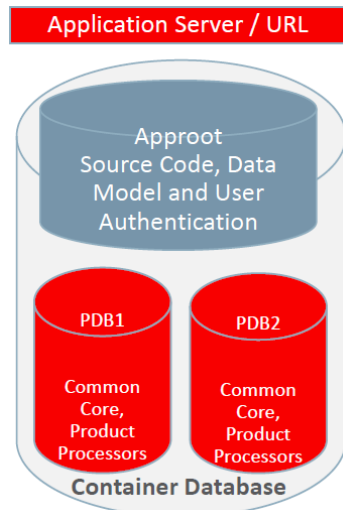
### 2.1 Shared Application

- ✓ In this model application would be deployed in an application container in 18C, Multiple front-end 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

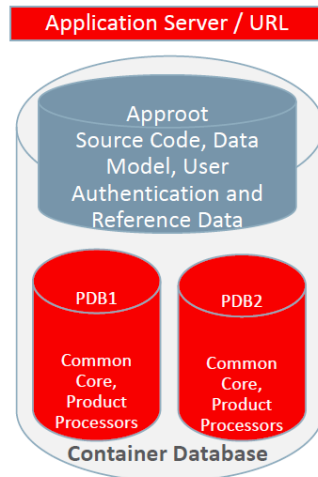
- ✓ In this model application would be deployed in an application container in 18C, Single front end application and an URL.
  - Application would be deployed in an Application Container
  - Source code at Approot level shared with PDBs
  - Data Model at Approot level shared with PDBs
  - Sharing of 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 18C, 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, Network, Pricing, Accounting & External systems preferences

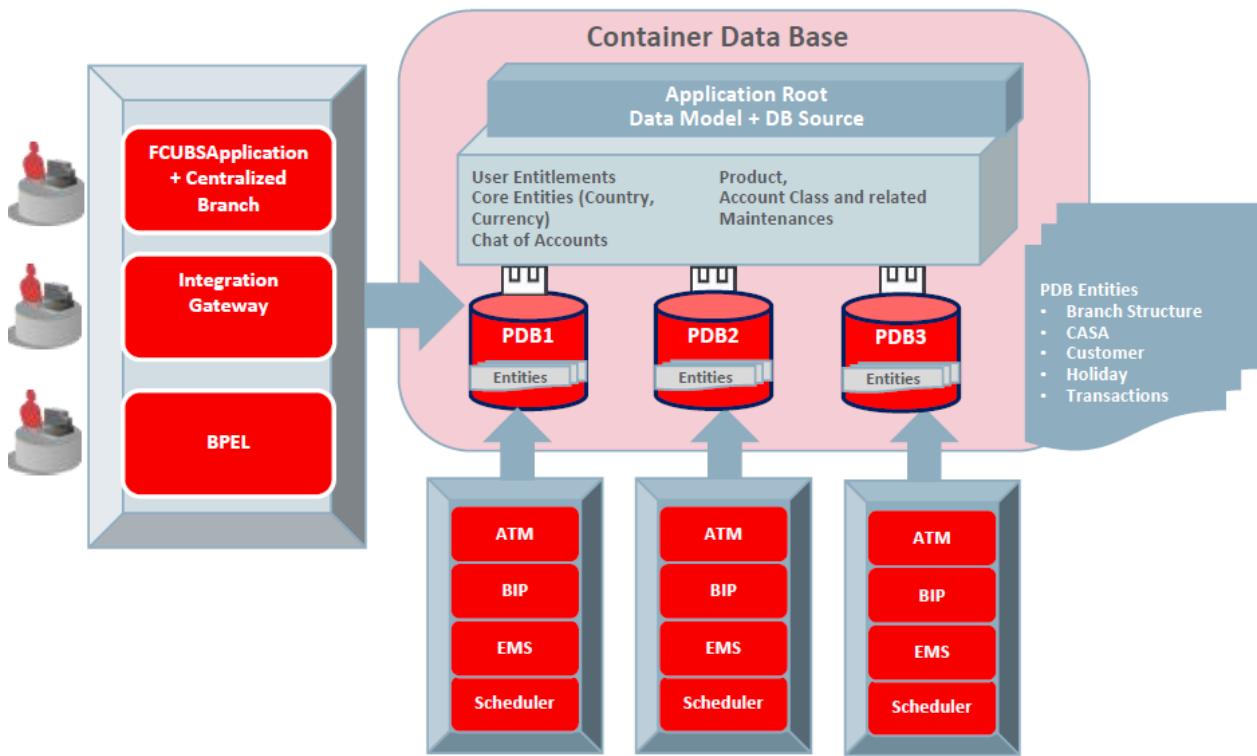


## 2.4 Shared Application with Shared Data - Custom

---

- ✓ This would be using Application Container in 18C, 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, Network, Pricing, Accounting & External systems preferences
  - 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:



**Component Deployment Architecture**

Application and Gateway will be common and single URL will be available for the application. 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
  - a. 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
  - i. Application installation has to be done in the aproot as version 1.0 with being user made explicit.
4. Application Root objects conversion
  - i. 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
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.

## 3.1 Creation of Application Template

---

### 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.

#### 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.sql

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

Input sample for the script:

<b>CDB Schema User Name</b>	Sys
<b>CDB Schema Password</b>	Sys
<b>CDB Host</b>	1.1.1.1
<b>CDB Port</b>	1524
<b>CDB Name</b>	FC142CDB
<b>DB Mounted Path</b>	/scratch/db1800dat
<b>Template PDB Name</b>	Templatepdb
<b>Common User Name</b>	CMNUSER
<b>Common User Password</b>	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 FCUBS\_Property\_File\_Creation.docx)

### 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, ROFCDBCompileRun.bat] by silent installer for respective product processor.
- Application Template PDB schema should be checked for sanity with zero invalids.

## 3.2 Creation of Application Root and Application Seed

---

### 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

### 3.2.2 Steps to be followed

---

- Below steps to be followed to configure Application Template PDB
  - ✓ Application Root and Application Seed Creation
  - ✓ Application Installation
  - ✓ Application Root objects conversion
  - ✓ Application Seed Sync with the Application Root

#### 3.2.2.1 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.

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:

<b>CDB Schema User Name</b>	Sys
<b>CDB Schema Password</b>	Sys
<b>CDB Host</b>	1.1.1.1
<b>CDB Port</b>	1524
<b>CDB Name</b>	FC142CBD
<b>DB Mounted Path</b>	/scratch/db1800dat
<b>Template PDB Name</b>	Templatepdb
<b>Approot Name</b>	Approot1
<b>Pdb to app pdb path</b>	C:\app_18c\client\user\product\18.0.0\client_1\rdbms\admin\pdb_to_apppdb.sql
<b>Common User Name</b>	CMNUSER

### 3.2.2.2 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.sql

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

Input sample for the script:

<b>CDB Schema User Name</b>	Sys
<b>CDB Schema Password</b>	Sys
<b>CDB Host</b>	1.1.1.1
<b>CDB Port</b>	1524
<b>Application Root Name</b>	Approot1
<b>Common User Name</b>	CMNUSER

### 3.2.2.3 Application Root objects conversion

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

### 3.2.2.3.1 Various Sharing type

- 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 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.

### 3.2.2.4 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

#### 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.

### 3.2.2.5 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 will be available in Application seed and Application PDBs.
- 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:

<b>Approot Schema Username</b>	CMNUSER
<b>Approot Schema Password</b>	CMNUSER
<b>Approot Host</b>	1.1.1.1
<b>Approot Port</b>	1524
<b>Application Root Name</b>	Approot1
<b>Application Name</b>	Oracle Banking Payments

### 3.3 Creation of Application PDB

---

#### 3.3.1 Purpose

---

- 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 to be followed

---

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.sql

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

Input sample for the script:

<b>CDB Name</b>	OBPM142CDB
<b>CDB Schema User Name</b>	Sys
<b>CDB Schema Password</b>	Sys
<b>CDB HOST</b>	1.1.1.1
<b>CDB PORT</b>	1522
<b>CDB Mounted Path</b>	/scratch/db1800dat/OBPM142CDB/templatePDB/users01.dbf
<b>Application Root Name</b>	OBPMAPPROOT
<b>Application PDB Name</b>	OBPMAPPPDB1

<b>PDB_TO_APPDB</b>	C:\app_18c\client\user\product\18.0.0\client_1\rdbms\admin\pdb_to_appdb.sql
---------------------	---

### 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: **Database\_Practices.docx**)

### 3.5 EAR Creation and Deployment

---

- Existing installer file ROFCEarRun.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.

### 4.1 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
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
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
  - 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

\* New function IDs



## 4.2 Default Approot Entities for Oracle Banking Payments

---

1. Access Rights
  - a. Role Queue Access Rights
  - User Queue Access Rights
2. Common Maintenances
  - a. Allowed Character Set
  - b. System Parameters
  - c. Pain.002 Reason Code Mapping
3. Cross Border/ High Value Payments
  - a. SWIFT gpi Static Preferences
  - b. SWIFT gpi Directory
  - c. Flat File gpi Directory Upload
  - d. RMA/RMA Plus
  - e. RTGS Directory
  - f. RTGS Directory Upload
  - g. PSD Preferences
4. External Systems
  - a. External Accounting System
  - b. External Credit Approval System
  - c. Exchange Rate System
  - d. External Pricing System
  - e. Sanctions Check System
5. Collections / Network
  - a. Creditor Scheme ID
  - b. Role Limit
  - c. Queue Connection Profile
6. US Payments
  - a. Fedwire, NACHA Directory
  - b. Fedwire, NACHA Directory Upload
  - c. RTP Routing Directory



Multi-Tenant Deployment  
Oracle Banking Payments  
Version 14.4.0.0.0  
[May] [2020]

Oracle Financial Services Software Limited  
Oracle Park  
Off Western Express Highway  
Goregaon (East)  
Mumbai, Maharashtra 400 063  
India

Worldwide Inquiries:  
Phone: +91 22 6718 3000  
Fax: +91 22 6718 3001

---

<https://www.oracle.com/industries/financial-services/index.html>

Copyright © 2017, 2020, Oracle and/or its affiliates. All rights reserved.

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

**U.S. GOVERNMENT END USERS:** Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 failsafe, 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.

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.

This software or hardware and documentation may provide access to or information on 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. 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.