Multi-Entity Deployment User Guide

Release 14.5.2.0.0

Part No. F45916-01

August 2021

ORACLE FINANCIAL SERVICES

ORACLE

Multi-Entity Deployment User Guide
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
www.oracle.com/financialservices/

Table of Contents

1. Overview

Banks may have multiple implementation across geographies that necessitates the need to support multiple entities.
“Multi Entity” feature, introduced in OBMA products, will enable a single instance of the product (and the underlying Oracle Banking Microservices Architecture platform) to onboard multiple entities of the bank onto the platform.
As part of “Multi Entity” feature, below are the functionalities that will be supported in all OBMA products

- Creation of "Multi-Entity Admin" user(s)
- Entity Definition and Maintenance
- Creation of "Entity Admin" User(s) & regular Users
- Mapping of users (entity admins or regular users) to one or more entities - The users of the application will be central in nature and users can have access to one or more entities.
- User Entitlement will be local to the entity

This guide details the approach that could be considered as a reference, while moving into multi-entity model.

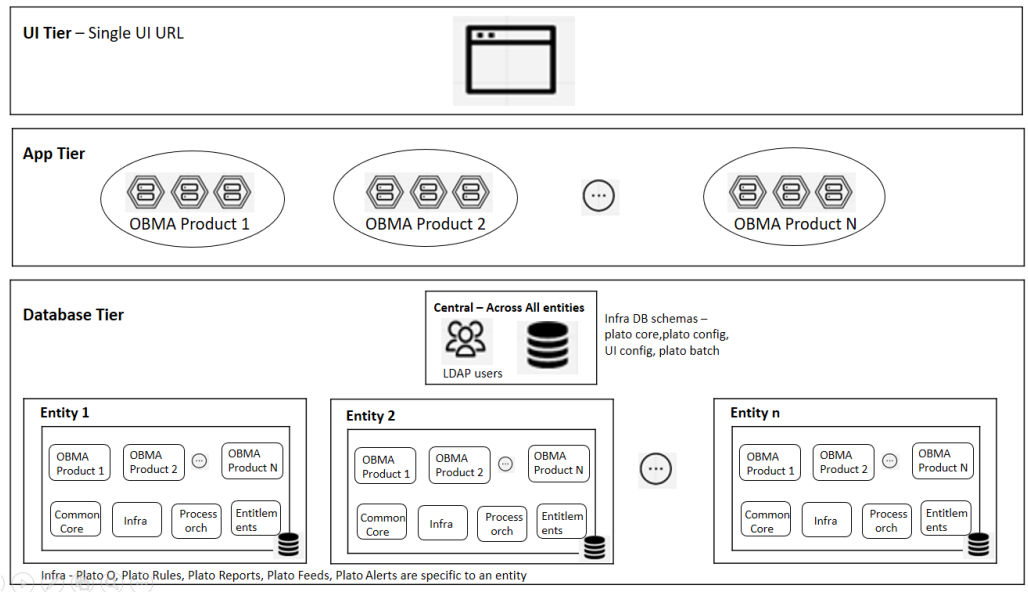
1.1 Deployment Diagram

Below diagram depicts multi-entity based deployment model.

UI Tier – UI Domain will be shared across multiple entities for a bank and so same UI URL will be used.

A user can be mapped to one or more entities and a single home entity. During login, user will be logged into to the home entity and an option would be provided to switch to any of the other associated entities.

Multi Entity admin user has the special access to create/modify new entities in the system.



App Tier – One or more managed servers that host all the microservices that are to be deployed for a product.

This includes

- Infrastructure services – Plato Infra services viz Plato Discovery, Plato Api Gateway, Plato Batch etc that are used across all products
- SMS service – for Role Based Authorisation
- Common Core and Mid-office Common core services - Common domain related services that are used across by one or more products
- Domain services – micro-services related to OBMA products (OBTFPM, OBCFPM, OBLM, OBVAM etc...)

Same as UI tier, App tier will also be shared across multiple entities. Based on the entity id provided in the request header, DB schema to the entity will be accessed for all CRUD operations.

Database Tier – Segregation of entities should be done in the DB layer. Separate DB schemas should be define and used for the entities.
Below are shared across multiple entities

- LDAP users
- few infrastructure related DB schemas - Plato Config, Plato UI config, Plato core, Plato Batch
- Below schemas will NOT be shared and should be specific to an entity
- Infra related schemas - Plato O, Plato Rules, Plato Reports, Plato Feeds, Plato Alerts
- User entitlements – SMS schema
- Common core schema
- Product specific DB schemas (each product will have multiple schemas; ideally 1 schema per microservice/sub-domain)

Banks that have a single entity should also follow the same architecture but with "DEFAULT_ENTITY" configured in the system.

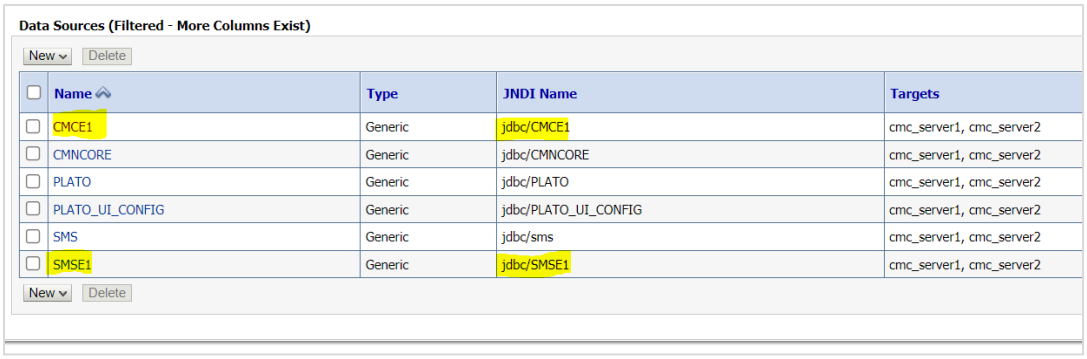
1.2 Default Entity Creation

During environment setup, when microservices are deployed, DMLs/DDLs related to "DEFAULT_ENTITY" will be executed through flyway scripts.

Multi entity Admin user should be created as mentioned in section 7.2 of "**Oracle_Banking_Microservices_Platform_Foundation_Installation_Guide**"

1.3 New Entity Creation

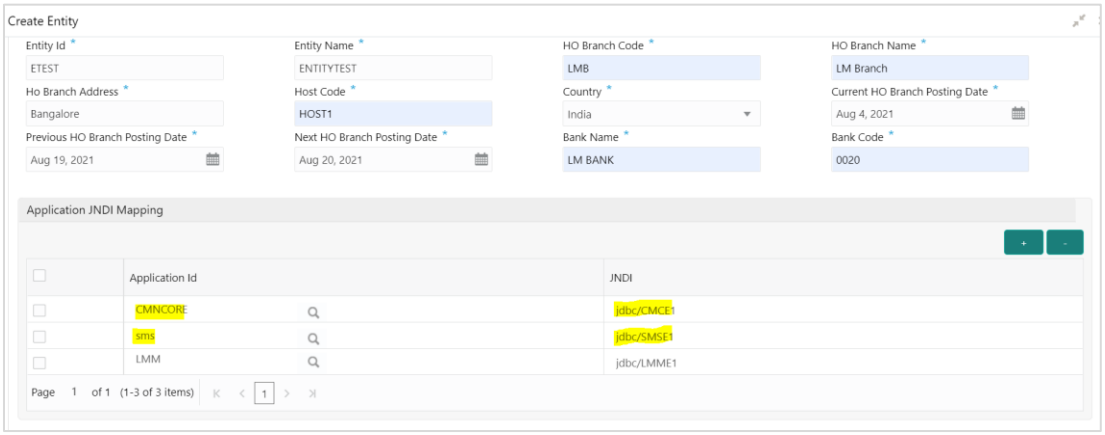
Multi entity admin users have the rights to create/modify entities that are to be created for the bank.
"Entity Maintenance" section of "Common_Core_User Guide" can be referred for creating new entities.
Before creating new entities through the application, DB schemas corresponding to various domains should be identified and corresponding "Data Sources" should be created in weblogic server.



Once the Data Sources are mapped with the corresponding DB schemas and servers, restart the PLATO, CMC, SMS and other required managed servers.

Check and verify in Eureka to see if all the services are up and running.

Login to the application as an entity admin user and proceed with entity creation. Select the required application id and map it with the new JNDI configured in weblogic.



As shown in the above snapshot, as part of entity creation through app-shell, JNDI names for each of the applications should be provided.

When the multi entity admin create an entity on click of the “Save” button in “Create Entity” screen, the following processes will execute in the background

- The entity details will be saved in the PLATO_TM_ENTITY table.
- The JNDIs will be saved in the APPLICATION_LEDGER table.
- The flyway scripts for all the micro services will get executed in their respective schemas.
- Once the flyway execution is completed a new role “ENTITY_ADMIN” will be created in the entity. This step will insert scripts into the following tables:
  - SMS_TM_ROLE
  - SMS_TW_ROLE
  - SMS_TM_ROLE_ACTIVITY
  - SMS_TW_ROLE_ACTIVITY

This role will be assigned to the entity admin user in the user creation step.

- The Head Office branch details will be inserted into the CMC_TM_CORE_BRANCH and CMC_TW_CORE_BRANCH tables.
- The Bank details will be inserted into the CMC_TM_CORE_BANK and CMC_TW_CORE_BANK tables.
- The System dates will be inserted into the CMC_TM_SYSTEM_DATES and CMC_TW_SYSTEM_DATES tables.

Once after confirming that the relevant DB entries are added as per above, Day-0 scripts should be run manually for each of the entities created through UI.