Oracle® Banking Microservices Architecture

Common Core Services Installation Guide

Release 14.6.1.0.0

F61110-01

August 2022

ORACLE®

Oracle Banking Microservices Architecture Common Core Services Installation Guide, Release 14.6.1.0.0

F61110-01

Contents

Preface

Introduction

This guide helps to install the common core services on designated environment. It is assumed that all the prior setup is already done related to WebLogic installation, WebLogic managed server creation, and Oracle DB installation.

> ✏️ **Note:**
>
> For the exact version to be installed, refer to the **Tech Stack** section of Release Notes.

It is recommended to use the dedicated managed server for each of the common core services.

Audience

This guide is intended for the WebLogic admin or ops-web team who are responsible for installing the OFSS banking products.

Acronyms and Abbreviations

The list of the acronyms and abbreviations used in this guide are as follows:

Table 1   List of Acronyms and Abbreviations

Abbreviation	Description
SMS	Security Management System
CMC	Common Core

List of Topics

This guide is organized as follows:

Table 2   List of Topics

Topics	Description
Database Setup	The topic provides the information about the database setup.
Domain and Cluster Configuration	The topic provides the information about the domain and cluster configuration.
Data Sources Creation	The topic provides the information about the data sources creation.
Deployments	The topic provides the information about the deployments.

Table 2   (Cont.) List of Topics

Topics	Description
Document Verification Framework	This topic provides the information about the document verification framework.
Restarts and Refresh	The topic provides the information about the restarts and refresh.
Logging Area	This topic provides the information about the logging area.
Known Issues - Resolutions	This topic provides the information about the known issues and its resolutions.

Organization

This guide allows the user to install the following services:

1. CMC-ACCOUNT-SERVICES

2. CMC ADDITIONAL-ATTRIBUTES-SERVICES

3. CMC-ADVICE-SERVICES

4. CMC-BASE-SERVICES

5. CMC-BATCH-SERVICES

6. CMC-BRANCH-SERVICES

7. CMC-BUSINESSOVERRIDES-SERVICES

8. CMC-CHARGES-CALCULATION-SERVICES

9. CMC-COREBANKING-ADAPTER-SERVICE

10. CMC-CURRENCY-SERVICES

11. CMC-CUSTOMER-SERVICES

12. CMC-DATASEGMENT-SERVICES

13. CMC-EXTERNAL-CHART-ACCOUNT

14. CMC-EXTERNAL-SYSTEM-SERVICES

15. CMC-EXTERNAL-VIRTUAL-ACCOUNT-SERVICES

16. CMC-FACILITIES-SERVICE

17. CMC-FC-AI-ML-SERVICES

18. CMC-ML-INDB-SERVICES

19. CMC-NLP-DASHBOARD-WIDGET-SERVICES

20. CMC-NLP- MAINTENANCE-SERVICES

21. CMC-NLP-OPENNLP-SERVICES

22. CMC-NLP-PIPELINE-SERVICES

23. CMC-NLP-TEXT-EXTRACTION-SERVICES

24. CMC-OBCBS-SERVICES

25. CMC-OBRH-SERVICE

26. CMC-REPORT-SERVICE

Related Documents

For more information, refer to the following documents:

- *ANNEXURE-1*

1
Database Setup

This topic describes the database setup related configuration for Common Core Installation.

Prerequisites

The below setup is dependent on Common Core Schema and SMS schema.

Before proceeding with the below setup, make sure that the Common Core Schema and SMS Schema is provided.

Database Setup

To set up Database, Common Core schema to be created.

Table 1-1   List of Services

Service Name	Schema Required
CMC-ACCOUNT-SERVICES	Yes (Common Core Schema)
CMC-ADDITIONAL-ATTRIBUTES-SERVICE	Yes (Common Core Schema)
CMC-ADVICE-SERVICES	Yes (Common Core Schema)
CMC-BASE-SERVICES	Yes (Common Core Schema)
CMC-BATCH-SERVICES	Yes (Common Core Schema)
CMC-BRANCH-SERVICES	Yes (Common Core Schema)
CMC-BUSINESSOVERRIDES-SERVICES	Yes (Common Core Schema)
CMC-CHARGES-CALCULATION-SERVICES	Yes (Common Core Schema)
CMC-COREBANKING-ADAPTER-SERVICE	Yes (Common Core Schema)
CMC-CURRENCY-SERVICES	Yes (Common Core Schema)
CMC-CUSTOMER-SERVICES	Yes (Common Core Schema)
CMC-DATASEGMENT-SERVICES	Yes (Common Core Schema)
CMC-EXTERNAL-CHART-ACCOUNT	Yes (Common Core Schema)
CMC-EXTERNAL-SYSTEM-SERVICES	Yes (Common Core Schema)
CMC-EXTERNAL-VIRTUAL-ACCOUNT-SERVICES	Yes (Common Core Schema)
CMC-FACILITIES-SERVICE	Yes (Common Core Schema)
CMC-FC-AI-ML-SERVICES	Yes (Common Core Schema)
CMC-ML-INDB-SERVICES	Yes (Common Core Schema)
CMC-NLP-DASHBOARD-WIDGET-SERVICES	Yes (Common Core Schema)
CMC-NLP-MAINTENANCE-SERVICES	Yes (Common Core Schema)
CMC-NLP-OPENNLP-SERVICES	Yes (Common Core Schema)
CMC-NLP-PIPELINE-SERVICES	Yes (Common Core Schema)
CMC-NLP-TEXT-EXTRACTION-SERVICES	Yes (Common Core Schema)
CMC-OBCBS-SERVICES	Yes (Common Core Schema)
CMC-OBRH-SERVICE	Yes (Common Core Schema)
CMC-REPORT-SERVICE	Yes (Common Core Schema)

Chapter 1

Table 1-1   (Cont.) List of Services

Service Name	Schema Required
CMC-RESOURCE-SEGMENT-ORCHESTRATOR-SERVICE	Yes (Common Core Schema)
CMC-SCREENCLASS-SERVICES	Yes (Common Core Schema)
CMC-SETTLEMENTS-SERVICES	Yes (Common Core Schema)
CMC-TRANSACTIONCONTROLLER-SERVICES	Yes (Common Core Schema)
CMC-TXN-CODE-SERVICES	Yes (Common Core Schema)
CMC-OPDS-SERVICES	Yes (Common Core Schema)
CMC-APPLICATIONCATEGORY-SERVICES	Yes (Common Core Schema)
CMC-CHECKLIST-SERVICES	Yes (Common Core Schema)
CMC-CHECKLISTMANAGEMENT-SERVICES	Yes (Common Core Schema)
CMC-COMMENTS-SERVICES	Yes (Common Core Schema)
CMC-DOCUMENT-SERVICES	Yes (Common Core Schema)
CMC-DOCUMENTMANAGEMENT-SERVICES	Yes (Common Core Schema)
CMC-KYCCHECK-SERVICES	Yes (Common Core Schema)
CMC-MAILNOTIFICATION-SERVICES	Yes (Common Core Schema)
CMC-PRIORITY-SERVICES	Yes (Common Core Schema)
CMC-PROCESSCODE-SERVICE	Yes (Common Core Schema)
CMC-SEQUENCEGENERATOR-SERVICES	Yes (Common Core Schema)
CMC-SLA-SERVICES	Yes (Common Core Schema)

2
Domain and Cluster Configuration

This topic describes the domain configuration for Common Core services.

Prerequisites

Before proceeding with the below setup, make sure that the below steps are done.

- The machine should have Java JDK installed.
- Oracle Fusion Middleware is installed on the machine.

> **✎ Note:**
>
> Before proceeding with the below steps, complete Oracle Banking
> Microservices Architecture installation as guided.

- Steps for creating all Common core domains, properties like port numbers, names will be changing based on the domain. Screenshots provided for such deviations. The domain creation process remains the same.

> **✎ Note:**
>
> For the exact version to be installed, refer to **Tech Stack** section of **Release Notes**.

Steps to Create Domain

It is recommended to have different managed server in one domain for each application. For Creating Domain and Configuration, refer to **Create Domain and Cluster Configuration** section in *ANNEXURE-1*.

3

Data Sources Creation

This topic describes the data sources creation for Common Core services.

Prerequisites

Database and application setup for Oracle Banking Microservices Architecture must be performed before the deployment setup.

The data sources for the respective micro-services must be created before the application deployment. Each data source targets the corresponding server on which the application will be deployed.

The following sections describes the list of data sources required to be created for Common Core services and the steps to configure them in the server.

Figure 3-1   Data Source Creation
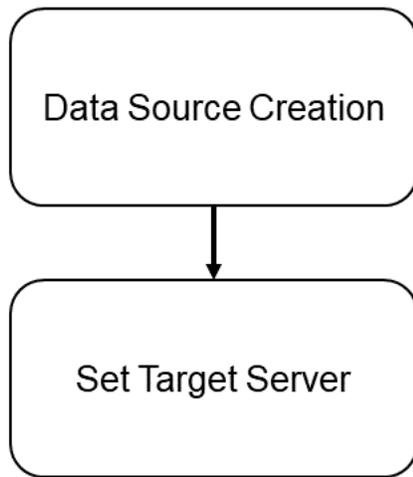
Data Sources List

The below list of the data sources must be created for each domain before the deployment of applications onto the managed servers.

Table 3-1   Data Sources List

Data Source Name	Data Source JNDI	Targets
PLATO	jdbc/PLATO	Common Core Managed Server
CMNCORE	jdbc/CMNCORE	Common Core Managed Server, API Gateway Server
PLATOBATCH	jdbc/PLATOBATCH	Common Core Managed Server

Table 3-1 (Cont.) Data Sources List

Data Source Name	Data Source JNDI	Targets
SMS	jdbc/sms	Common Core Managed Server
PLATOUI	jdbc/PLATO_UI_CONFIG	Common Core Managed Server
PLATOFEED	jdbc/PLATOFEED	Common Core Managed Server
PLATOSECURITY	jdbc/PLATO_SECURITY	Common Core Managed Server
<PRODUCT PROJECTION SCHEMA> (Required only for DB based ML use cases)	jdbc/<PRODUCT PROJECTION SCHEMA>	Common Core Managed Server

> **Note:**
>
> Steps to create data source in WebLogic is same for all the above except for the connection details and names.

Common Core Services using Plato-Feed

The below listed common core services requires the data source to be created for plato-feeds on each domain prior to deployment of applications onto managed servers.

Table 3-2 List of Services using Plato-Feed

Service Name	Data Source Name	Data Source JNDI	Targets
cmc-account-services	PLATOFEED	jdbc/PLATOFEED	Common Core Managed Server
cmc-base-services	PLATOFEED	jdbc/PLATOFEED	Common Core Managed Server
cmc-branch-services	PLATOFEED	jdbc/PLATOFEED	Common Core Managed Server
cmc-settlement-services	PLATOFEED	jdbc/PLATOFEED	Common Core Managed Server
cmc-currency-services	PLATOFEED	jdbc/PLATOFEED	Common Core Managed Server
cmc-customer-services	PLATOFEED	jdbc/PLATOFEED	Common Core Managed Server

Steps to create Data Source

For creating data sources in managed server, refer to **Create Data Sources** section in *ANNEXURE-1*.

4

Deployments

This topic describes the list of war files for the deployment and depreciation of the Common Core services, and the steps to deploy and undeploy them in the server.

Prerequisites

The database setup and data sources creation need to be performed before the application deployment stage. Each of the services corresponds to a specific war file that needs to be deployed into the server.

Figure 4-1   Deployments

```
┌──────────────────────────┐
│                          │
│   Collect the war files  │
│                          │
└────────────┬─────────────┘
             │
             ▼
┌──────────────────────────┐
│                          │
│    Deploy into Server    │
│                          │
└──────────────────────────┘
```
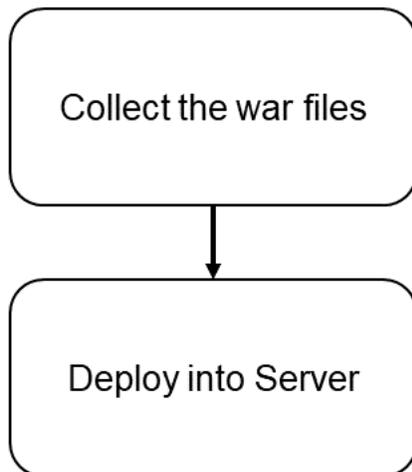
Deployments List

The below table provides the details of the deployments required on each domain for the Common Core application to run. Deploy one after the other in the same order as given below.

Table 4-1   Deployments List

Application	Archive name	OSDC path	Targets
CMC-ACCOUNT-SERVICES	cmc-account-services-{version}.war	{Unzip OSDC Package} \cmc-account-services	Common Core Managed Server
CMC-ADDITIONAL-ATTRIBUTES-SERVICE	cmc-additional-attributes-services-{version}.war	{Unzip OSDC Package} \cmc-additional-attributes-services	Common Core Managed Server
CMC-ADVICE-SERVICES	cmc-advice-services-{version}.war	{Unzip OSDC Package} \cmc-advice-services	Common Core Managed Server

Table 4-1   (Cont.) Deployments List

Application	Archive name	OSDC path	Targets
CMC-BASE-SERVICES	cmc-base-services-{version}.war	{Unzip OSDC Package}\cmc-base-services	Common Core Managed Server
CMC-BATCH-SERVICES	cmc-batch-services-{version}.war	{Unzip OSDC Package}\cmc-batch-services	Common Core Managed Server
CMC-BRANCH-SERVICES	cmc-branch-services-{version}.war	{Unzip OSDC Package}\cmc-branch-services	Common Core Managed Server
CMC-BUSINESSOVERRIDES-SERVICES	cmc-businessoverrides-services-{version}.war	{Unzip OSDC Package}\cmc-businessoverrides-services	Common Core Managed Server
CMC-CHARGES-CALCULATION-SERVICES	cmc-charges-calculation-services-{version}.war	{Unzip OSDC Package}\cmc- charges-calculation -services	Common Core Managed Server
CMC-COREBANKING-ADAPTER-SERVICE	cmc-corebanking-adapter-service-{version}.war	{Unzip OSDC Package}\cmc-corebanking-adapter-service	Common Core Managed Server
CMC-CURRENCY-SERVICES	cmc-currency-services-{version}.war	{Unzip OSDC Package}\cmc-currency-services	Common Core Managed Server
CMC-CUSTOMER-SERVICES	cmc-customer-services-{version}.war	{Unzip OSDC Package}\cmc-customer-services	Common Core Managed Server
CMC-DATASEGMENT-SERVICES	cmc-datasegment-services-{version}.war	{Unzip OSDC Package}\cmc-datasegment-services	Common Core Managed Server
CMC-EXTERNAL-CHART-ACCOUNT	cmc-external-chart-account-{version}.war	{Unzip OSDC Package}\cmc-external-chart-account	Common Core Managed Server
CMC-EXTERNAL-SYSTEM-SERVICES	cmc-external-system-services-{version}.war	{Unzip OSDC Package}\cmc-external-system-services	Common Core Managed Server
CMC-EXTERNAL-VIRTUAL-ACCOUNT-SERVICES	cmc-external-virtual-account-services-{version}.war	{Unzip OSDC Package}\cmc-external-virtual-account-services	Common Core Managed Server
CMC-FACILITIES-SERVICE	cmc-facilities-service-{version}.war	{Unzip OSDC Package}\cmc-facilities-service	Common Core Managed Server
CMC-FC-AI-ML-SERVICES	cmc-fc-ai-ml-services-{version}.war	{Unzip OSDC Package}\cmc-fc-ai-ml-services	Common Core Managed Server
CMC-ML-INDB-SERVICES	cmc-ml-indb-services-{version}.war	{Unzip OSDC Package}\cmc-ml-indb-services	Common Core Managed Server
CMC-NLP-DASHBOARD-WIDGET-SERVICES	cmc-nlp-dashboard-widget-services-{version}.war	{Unzip OSDC Package}\cmc-nlp-dashboard-widget-services	Common Core Managed Server
CMC-NLP-MAINTENANCE-SERVICES	cmc-nlp-maintenance-services-{version}.war	{Unzip OSDC Package}\cmc-nlp-maintenance-services	Common Core Managed Server
CMC-NLP-OPENNLP-SERVICES	cmc-nlp-opennlp-services-{version}.war	{Unzip OSDC Package}\cmc-nlp-opennlp-services	Common Core Managed Server
CMC-NLP-PIPELINE-SERVICES	cmc-nlp-pipeline-services-{version}.war	{Unzip OSDC Package}\cmc-nlp-pipeline-services	Common Core Managed Server

Table 4-1   (Cont.) Deployments List

Application	Archive name	OSDC path	Targets
CMC-NLP-TEXT-EXTRACTION-SERVICES	cmc-nlp-text-extraction-services-{version}.war	{Unzip OSDC Package}\cmc-nlp-text-extraction-services	Common Core Managed Server
CMC-OBCBS-SERVICES	cmc-obcbs-services-{version}.war	{Unzip OSDC Package}\cmc-obcbs-service	Common Core Managed Server
CMC-OBRH-SERVICES	cmc-obrh-services-{version}.war	{Unzip OSDC Package}\cmc-obrh-services	Common Core Managed Server
CMC-REPORT-SERVICE	cmc-report-service-{version}.war	{Unzip OSDC Package}\cmc-report-service	Common Core Managed Server
CMC-RESOURCE-SEGMENT-ORCHESTRATOR-SERVICE	cmc-resource-segment-orchestrator-service-{version}.war	{Unzip OSDC Package}\cmc-resource-segment-orchestrator-service	Common Core Managed Server
CMC-SCREENCLASS-SERVICES	cmc-screenclass-services-{version}.war	{Unzip OSDC Package}\cmc-screenclass-services	Common Core Managed Server
CMC-SETTLEMENTS-SERVICES	cmc-settlements-services-{version}.war	{Unzip OSDC Package}\cmc-settlements-services	Common Core Managed Server
CMC-TRANSACTIONCONTROLLER-SERVICES	cmc-transactioncontroller-services-{version}.war	{Unzip OSDC Package}\cmc-transactioncontroller-services	Common Core Managed Server
CMC-TXN-CODE-SERVICES	cmc-txn-code-services-{version}.war	{Unzip OSDC Package}\cmc-txn-code-services	Common Core Managed Server
CMC-APPLICATIONCATEGORY-SERVICES	cmc-applicationcategory-services-{version}.war	{Unzip OSDC Package}\cmc-applicationcategory-services	Mid-Office Managed Server
CMC-CHECKLIST-SERVICES	cmc-checklist-services-{version}.war	{Unzip OSDC Package}\cmc-checklist-services	Mid-Office Managed Server
CMC-CHECKLISTMANAGEMENT-SERVICES	cmc-checklistmanagement-services-{version}.war	{Unzip OSDC Package}\cmc-checklistmanagement-services	Mid-Office Managed Server
CMC-COMMENTS-SERVICES	cmc-comments-services-{version}.war	{Unzip OSDC Package}\cmc-comments-services	Mid-Office Managed Server
CMC-DOCUMENT-SERVICES	cmc-document-services-{version}.war	{Unzip OSDC Package}\cmc-document-services	Mid-Office Managed Server
CMC-DOCUMENTMANAGEMENT-SERVICES	cmc-documentmanagement-services-{version}.war	{Unzip OSDC Package}\cmc-documentmanagement-services	Mid-Office Managed Server
CMC-KYCCHECK-SERVICES	cmc-kyccheck-services-{version}.war	{Unzip OSDC Package}\cmc-kyccheck-services	Mid-Office Managed Server
CMC-MAILNOTIFICATION-SERVICES	cmc-mailnotification-services-{version}.war	{Unzip OSDC Package}\cmc-mailnotification-services	Mid-Office Managed Server

Table 4-1 (Cont.) Deployments List

Application	Archive name	OSDC path	Targets
CMC-PRIORITY-SERVICE	cmc-priority-service-{version}.war	{Unzip OSDC Package}\cmc-priority-service	Mid-Office Managed Server
CMC-PROCESSCODE-SERVICE	cmc-processcode-service-{version}.war	{Unzip OSDC Package}\cmc-processcode-service	Mid-Office Managed Server
CMC-SEQUENCEGENERATOR-SERVICES	cmc-sequencegenerator-services-{version}.war	{Unzip OSDC Package}\cmc-sequencegenerator-services	-
CMC-SLA-SERVICES	cmc-sla-services-{version}.war	{Unzip OSDC Package}\cmc-sla-services	-
CMC-OPDS-SERVICES	cmc-opds-services-{version}.war	{Unzip OSDC Package}\cmc-opds-services	Common Core Managed Server

> **Note:**
>
> Refer to OSDC file for the exact version number for each service.
>
> For the instructions to deploy the application, refer to the **Deploy Application** section in *ANNEXURE-1* .

Depreciated Services

Deploy the common core service to clean up the depreciated service-related Database objects from common core / SMS / PLATO schemas.

Table 4-2 Depreciation File

Application	Archive name	OSDC path	Targets
CMC-SERVICE-DEPRECATION-SVC	cmc-service-deprecation-svc-{version}.war	{Unzip OSDC Package}\cmc-service-deprecation-svc	Common Core Managed Server

The below table provides the details of the depreciated services from the Common Core application and undeploy the services one after other in the same order as given below.

Table 4-3 List of Depreciated services

Application	Archive name	Targets
CMC-EARMARK-SERVICES	cmc-earmark-services-{version}.war	Common Core Managed Server
CMC-QUEUE-SERVICES	cmc-queue-services-{version}.war	Common Core Managed Server

Table 4-3   (Cont.) List of Depreciated services

Application	Archive name	Targets
CMC-SERVICE-DEPRECATION-SVC	cmc-service-deprecation-svc-{version}.war	Common Core Managed Server

> **Note:**
>
> Refer to OSDC file for the exact version number for each service.
>
> For the instructions to undeploy the deprecated applications, refer to the **Undeploy Application** section in *ANNEXURE-1*.

5

Document Verification Framework

This topic describes the document verification framework where after deployment of Common Core applications in the WebLogic server.

Prerequisites

Python, Tesseract and other required libraries must be installed for running document verification APIs.

Python Installation:
Oracle Linux yum server hosts software for Oracle Linux and compatible distributions. If the latest approved distribution of python (python 3.9.5) is not available on the yum server, then follow the steps mentioned in the below location for installing python manually. Please make sure you specify the python version properly.

1. `sudo yum install wget`

2. `sudo yum install gcc readline readline-devel`

3. `sudo yum install zlib zlib-devel`

4. `sudo yum install libffi-devel openssl-devel`

5. `wget https://www.python.org/ftp/python/3.9.5/Python-3.9.5.tgz`

6. `tar xzf Python-3.9.5.tgz`

7. `cd Python-3.9.5`

8. `sudo ./configure --enable-optimizations --prefix=/scratch/software/python39`

9. `sudo make altinstall (make altinstall is used to prevent replacing the default python binary file /usr/bin/python.)`

10. `sudo update-alternatives --install /usr/bin/python3 python3 /scratch/ software/python39/bin/python3.9 (Adding symlink to python3)`

11. `sudo update-alternatives --config python3 (This will show the list of symlinks available. Select the required/particular version, in our case python3.9.5)`

> **✎ Note:**
>
> The user may encounter some problems while executing the 10th and 11th steps. If the user gets an error as shown below, then the user should be a root user to perform these two additional steps.

Error: "`sudo: update-alternatives: command not found`"

To be a root user, you will have to execute the command mentioned below and then continue to execute 10th and 11th step,

```
sudo su – root
```

Tesseract Installation

Tesseract is an optical character recognition (OCR) engine for various operating systems. The latest version must be installed on the machine to extract the text from the documents.

Refer to **9.4 Tesseract Installation** section in **Oracle Banking Microservices Platform Foundation Installation Guide** to manually install the latest version of the tesseract.

Document Verification Application Installation

The app will be shipped as a byte-coded whl file. This wheel file will install all the implementation files without the dependencies. All the required dependencies are bundled together in a python.zip file which are to be extracted and installed separately (refer to Step 4 below). It's recommended to install the whl file and the dependencies in a virtual environment using "pip" so that it doesn't affect any other operations or applications running in the system.

In addition to this, a Config.ini file is required for the Eureka server configuration. Please create a config file with the name Config.ini and paste the text below:

```
[DEFAULT]
eureka_server=http://<Host Name>:<Port Number>/plato-discovery-service/
eureka
```

You can edit the eureka server address if needed. The app name should not be changed. This is important for Role-Based Access Control. Please note that registering the app on Eureka is optional and you can skip this if not needed. But in any case, Config.ini is required. In eureka_server variable you can simply give localhost url.

The folder structure to be followed is:

```
├── root_dir
├── python
├── Config.ini
└── venv
```

> **Note:**
>
> Please make sure that the user are using linux operating system and the installed Python version is 3.9.5 and that of pip is above 20.0.0. Run the following command to upgrade pip to the latest version.

```
pip install --upgrade pip
```
Once the pip is upgraded, follow the steps below to install the app and the dependencies:

1. Create python virtual environment using the following command in python3.9.5,

   ```
   python –m venv <venv_name>
   ```

2. Activate the newly created virtual environment by using the following command.

```
source <venv_name>/bin/activate
```

3. Use the below command to install the application wheel package provided,

```
pip install <wheel_package_name>.whl
```

4. Now all the dependencies need to be installed. In order to do this, extract the python.zip file provided, go into the python folder (cd python/) and run the following commands:

```
pip install configparser --no-index --find-links.
pip install connexion --no-index --find-links.
pip install datefinder --no-index --find-links.
pip install dateparser --no-index --find-links.
pip install Flask --no-index --find-links.
pip install importlib-metadata --no-index --find-links.
pip install numpy --no-index --find-links.
pip install opencv-python --no-index --find-links.
pip install pdf2image --no-index --find-links.
pip install Pillow --no-index --find-links.
pip install pyap --no-index --find-links.
pip install pybase64 --no-index --find-links.
pip install pytesseract --no-index --find-links.
pip install python-dateutil --no-index --find-links.
pip install scipy --no-index --find-links.
pip install six --no-index --find-links.
pip install python-magic --no-index --find-links.
(optional) pip install py-eureka-client --find-links.
```

Installing py-eureka-client is optional as it is needed only if you want to register the app on Eureka.

5. After installing the wheel package and the dependencies, we can run the document verification server using the below-mentioned command,

```
python -m ofss_ml_document_verification_server
```

6. Please note: This will by default run the app on port 8090 and not register the app to Eureka. To do that please use the below-mentioned command:

```
python -m ofss_ml_document_verification_server -p 5000 -r true
```
The above commands make the app run on port 5000 and registers to the Eureka server as well. These arguments may or may not be used together and the port number can be any. By default, the system has been configured to -r false.
Please note that once the service is registered on Eureka, it will need role-based access to send and receive requests.
For example: if the app is registered on `http://<Host Name>:<Port Number>/plato-discovery-service`, then we need a bearer token from `http://<Host Name>:<Port Number>/api-gateway/platojwtauth` and then call `http://<Host Name>:<Port Number>/api-gateway/ofss_ml_document_verification_server/extractInformation` with the following headers:
```
1. Authorization – bearer <token>
2. appid- (ex- CMNCORE)
3. branchCode
4. content-Type – application/json
5. userId
```
Please note that the userId and branchCode will be based on the flyway script entries.

SMS Scripts:

```
Insert into SMS_TM_SERVICE_ACTIVITY
SERVICE_ACTIVITY_CODE,DESCRIPTION,CLASS_NAME,METHOD_NAME,APPLICATION
_ID,SERVICE_TYPE,UI_ACTIVITY_CODE)
values
('CMC_SA_OFSS_ML_DOCUMENT_VERIFICATION_EXTRACT_INFORMATION','Extract
s meaningful details from an
image','OFSS_ML_DOCUMENT_VERIFICATION_SERVER','extractInformation','
CMC','Service API',null);
commit;


Insert into SMS_TM_FUNCTIONAL_ACTIVITY
(FUNCTIONAL_ACTIVITY_CODE, APPLICATION_ID, TYPE) values
('CMC_FA_OFSS_ML_DOCUMENT_VERIFICATION_EXTRACT_INFORMATION','CMC','O
');
commit;


Insert into SMS_TM_FUNC_ACTIVITY_DETAIL
(ID,FUNCTIONAL_ACTIVITY_CODE,SERVICE_ACTIVITY_CODE) values
('CMC_FD_OFSS_ML_DOCUMENT_VERIFICATION_EXTRACT_INFORMATION',
'CMC_FA_OFSS_ML_DOCUMENT_VERIFICATION_EXTRACT_INFORMATION',
'CMC_SA_OFSS_ML_DOCUMENT_VERIFICATION_EXTRACT_INFORMATION');
commit;
```

Plato Scripts:

```
Insert into PROPERTIES
(ID,APPLICATION,PROFILE,LABEL,KEY,VALUE) values
(PROPERTIES_ID_SEQ.NEXTVAL,'plato-api-
gateway','jdbc','jdbc','zuul.routes.ofssmldoc.path','/
ofss_ml_document_verification_server/**');
Insert into PROPERTIES
(ID,APPLICATION,PROFILE,LABEL,KEY,VALUE) values
(PROPERTIES_ID_SEQ.NEXTVAL,'plato-api-
gateway','jdbc','jdbc','zuul.routes.ofssmldoc.serviceId','ofss_ml_do
cument_verification_server');
Insert into PROPERTIES
(ID,APPLICATION,PROFILE,LABEL,KEY,VALUE) values
(PROPERTIES_ID_SEQ.NEXTVAL,'plato-api-
gateway','jdbc','jdbc','zuul.routes.ofssmldoc.stripPrefix','false');
commit;
```

This procedure makes sure that only authenticated users can use the API. However, the developers running the app have an option to disable registry on Eureka and test the API normally.

7. To run the document verification server in the background, use the command below.

```
nohup python -m ofss_ml_document_verification_server & tail -f
nohup.out
```

> **Note:**
>
> After the execution of the above command, all the execution logs will be added to nohup.out (text file). Now the user may close the terminal and the app will still be running on port.

8. To terminate/kill the app, we can use the netstat command to find the process_id using the port on which the app is running and then use the kill command with the process_id of the app as shown below to terminate the application.

```
netstat -nlp | grep 8090
kill -9 <process_id>
```

6
Restarts and Refresh

This topic provides the information to restart and refresh the application.

Once the application is deployed, restart all the managed servers. For each application, call path `/refresh` for refreshing the configuration properties.

> **Note:**
>
> Refer to the **Restart Servers** section in *ANNEXURE-1* for the instruction to restart the server.

7
Logging Area

This topic describes the logging area where the logs are found after deploying the Common Core applications in the WebLogic server.

Common Core Application writes logs in the below area of the server:

`<WEBLOGIC_DOMAIN_CONFIG_AREA>/servers/CMNCOREAPP/logs/CMNCOREAPP.out`

Let's assume a domain has been created **CommonCore_domain** with **managed_server** name called **CMNCOREAPP** in the following server area.

`/scratch/oracle/middleware/user_projects/domains/CommonCore_domain.`

Logging area for Common core is `/scratch/oracle/middleware/user_projects/domains/ CommonCore_domain/servers/CMNCOREAPP/logs/CMNCOREAPP.out`

8

Known Issues - Resolutions

This topic describes the known issues encountered while using the application and its resolutions.

cmc-ml-indb-services

Troubleshooting deployment failure in cmc-ml-indb-services.war

Deployment Error –

1. org.flywaydb.core.api.FlywayException: Validate failed: Migration checksum mismatch for migration version 101.32.5.0.0.4.101032004.1.1 -> Applied to database : 589007451 -> Resolved locally : -692687536 :org.flywaydb.core.api.FlywayException:Validate failed: Migration checksum mismatch for migration version 101.32.5.0.0.4.101032004.1.1 -> Applied to database : 589007451 -> Resolved locally : -692687536

2. org.flywaydb.core.api.FlywayException: Validate failed: Migration checksum mismatch for migration version 101.32.5.0.0.4.101032004.1.1 -> Applied to database : -692687536-> Resolved locally : 589007451:org.flywaydb.core.api.FlywayException:Validate failed: Migration checksum mismatch for migration version 101.32.5.0.0.4.101032004.1.1 -> Applied to database : -> -692687536 Resolved locally : 589007451

Resolution –

1. Error 1 should get resolved after applying the war from latest patchset.

2. Error 2 results in case there are manual modifications made in table flyway_schema_history. That is, checksum of script/file V101_32_5.0.0_4_101032004_1_1__CMC_TM_ML_ALGO_REPO.sql has been manually changed/updated to -692687536 in flyway table. If so please delete all the entries in flyway_schema_history table for script V101_32_5.0.0_4_101032004_1_1__CMC_TM_ML_ALGO_REPO.sql then redeploy/re-run the latest cmc-ml-indb-services.war.

cmc-branch-service

Troubleshooting steps for failed cmc-branch-service-6.0.0

For branch service, the below steps needs to be executed for deploying the service over the March patchset.

1. Run the following scripts on the SMS schema:
   delete from "flyway_schema_history" where "version" in ('101.4.5.11.0.1.00051001006.10.1');

   delete from SMS_TM_FUNC_ACTIVITY_DETAIL where ID in ('CMC_FA_BRANCH_WORKHR_GETALL','CMC_BRANCH_WORKHR_GETID','CMC_BRANCH_WORKHR_SAVE','CMC_FA_BRANCH_TIME_GETALL','CMC_BRANCH_WORKHR_AUTHQRY','CMC_BRANCH_WORKHR_AUTH','CMC_BRANCH_WORKHR_CLOSERECRD','CMC_BRANCH_WORKHR_DELETE','CMC_BRANCH_WORKHR_GETHISTRY', 'CMC_BRANCH_WORKHR_MODRECRD','CMC_BRANCH_WORKHR_ROPNRECRD');

commit;

2. Start the cmc-branch-service-6.0.0 application in the Weblogic.

Index