Conductor Process Installation Guide
Oracle Banking Corporate Lending Process Management

Release 14.4.0.0.0

Part No. F30281-01

May 2020

Oracle Banking Corporate Lending Process Management Conductor Process Installation 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. Preface

1.1 Introduction

This guide would help you to install the OBCLPM Conductor based process on designated environment. It is assumed that all the prior setup is already done related to Netflix Conductor.

1.2 Audience

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

1.3 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.

1.4 Organization

This installation user guide would allow you to install following -

1. OBCLPM Conductor Process.

1.5 Related Documents

- Oracle Banking Corporate Lending Process Management Installation Guide
- Plato Infrastructure Services Installation Guide

2. Deployments

2.1 OBCLPM Processes

Deploy the following list of conductor processes for OBCLPM:

In conductor workflows are defined using a JSON based DSL and includes a set of tasks that are executed as part of the workflows. The tasks are either system tasks or simple tasks (aka worker tasks).
These simple/worker tasks are implemented by application(s) and run in a separate environment from Conductor. These tasks talk to Conductor server via REST client.

In OBCLPM, for every workflow a workflow definition json (e.g. LoanAmmendmend.JSON) file and if required its own worker tasks definition json (TaskDefinitions.JSON) file will be maintained. Also, there is a SharedSimpleTaskDefinitions.JSON file having task definitions that are used across different workflows.

S.No	OBCLPM Processes	Process Specific Tasks (if any)
1	LOAN AMENDMENT PROCESS	CLMO_FA_LNAMND_HANDOFF
2	LOAN DISBURSEMENT PROCESS	CLMO_FA_LNDSBR_HANDOFF
3	LOAN DRAFT PROPOSAL PROCESS	
4	LOAN DRAWDOWN PROCESS	
5	LOAN FINANCIAL AMENDMENT PROCESS	CLMO_FA_FNAMND_HANDOFF
6	LOAN ORIGINATION PROCESS	CLMO_FA_LNPMNT_HANDOFF
7	LOAN PAYMENT PROCESS	CLMO_FA_LNPMNT_HANDOFF
8	NLP PROCESS	CLMO_FA_NLPPRC_ENRICH_NLPDETAILS, CLMO_FA_NLPPRC_UPDATE_TXNLOGS, CLMO_FA_NLPPRC_INITIATE_CLMO_APPLICATION
9	RATE QUOTATION PROCESS	CLMO_FA_LNRQTN_CONTRACT_REVERSAL, CLMO_FA_LNRQTN_INIT_LNORG
10	ROLLOVER PROCESS	CLMO_FA_ROLVAU_HANDOFF
11	SYNDICATION AGENCY DRAWDOWN PROCESS	CLMO_FA_SNAGDD_TRYREQ
12	SYNDICATION OPSENRICHMENT PROCESS	CLMO_FA_SNOPEN_HANDOFF
13	SYNDICATION PARTICIPANT PROCESS	
14	SYNDICATION PARTICIPANT DRAWDOWN PROCESS	CLMO_FA_SNPTDD_TRYREQ
15	SYNDICATION POST MANDATE AGENT PROCESS	
16	SYNDICATION POST MANDATE ARRANGER PROCESS	CLMO_FA_DO_NOTHING
17	SYNDICATION PREMANDATE COARRANGER PROCESS	CLMO_FA_CREDIT_CHECK
18	SYNDICATION PRE MANDATE LEAD ARRANGER PROCESS	

S.No	Common Task Definitions (Used across different OBCLPM processes
1	FA_KYC_CHECKER
2	FA_CAMS_HANDOFF
3	FA_LNORGN_HANDOFF
4	FA_STATUS_UPDATE
5	FA_RISK_APPROVAL
6	FA_INITIATE_OPS_ENRICH
7	FA_BORROWER_ONBOARDING
8	FA_MILESTONE_CHECK
9	FA_INITAITE_QE
10	FA_SNDRWD_HANDOFF

2.2 Steps to Deploy Conductor Process

Note: Steps to Deploy a process remains the same for all the process files.

Pre-Requisite: **PLATO-O** and **PLATO-ORCH-SERVICE** services should be up and registered in the Eureka registry. Refer: ‘Plato_Infrastructure_Services_Installation_Guide.docx’ for the installation of PLATO-O and PLATO-ORCH-SERVICE.

Post-Requisite: CLMO-ORCHESTRATION-SERVICE should be deployed after registering common task definitions and all workflow specific task definitions (total of 26 task definitions listed above) as this service would have worker implementation for all the simple tasks. Presently this service has 26 worker implementations with 36 threads configured for worker coordinator to manage these workers.

In OBCLPM, for every workflow a workflow definition json (e.g. LoanDraftProposal.JSON) file and if required its own worker tasks definition json (TaskDefinitions.JSON) file will be maintained. Also, there is a SharedSimpleTaskDefinitions.JSON file having simple task definitions that are used across all workflows. To deploy all or any workflow definition,

1. Firstly shared simple tasks definitions should be done. for that below endpoint can be invoked from postman tool with header and payload as described below.
   (Note: The host name and port number of the PLATO-ORCH-SERVICE from the eureka registry should be used in this endpoint)

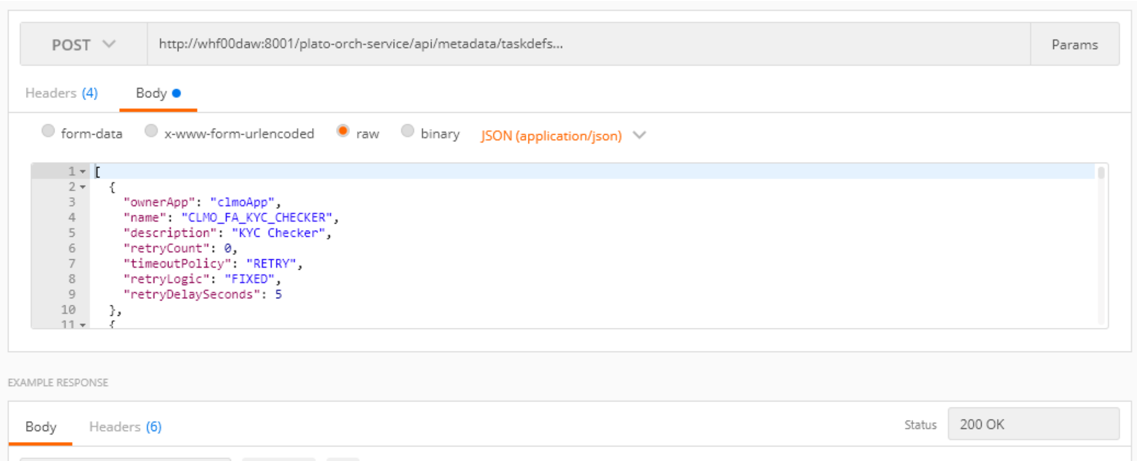
Task definitions end point: http://whf00mdj:9009/plato-orch-service/api/metadata/taskdefs

HTTP Method: POST

Header parameters:

Content-Type:application/json
appId:platoorch
branchCode:004
userId:USER1

Payload: SharedSimpleTaskDefinitions.JSON content

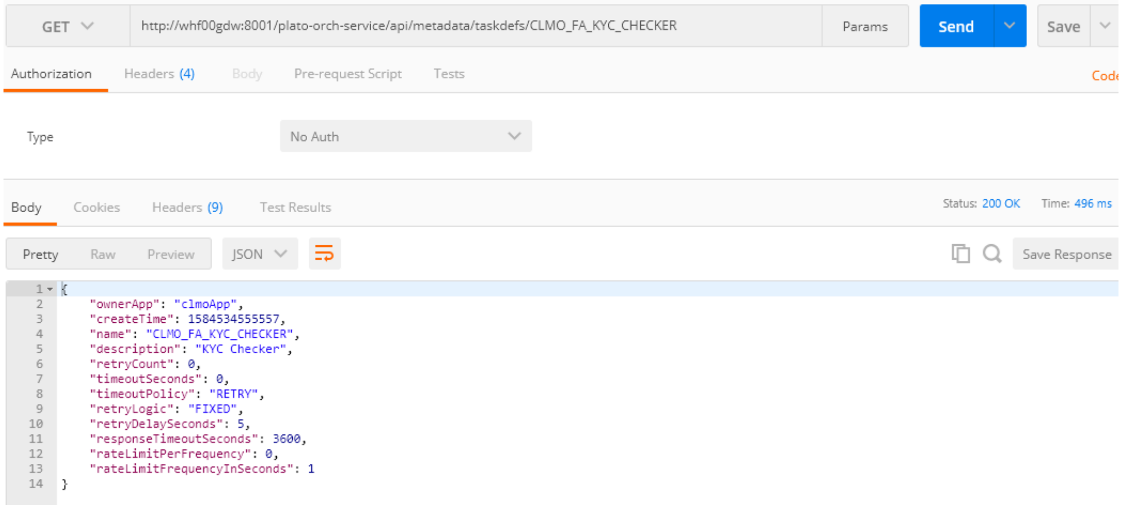


http://whf00daw:8001/plato-orch-service/api/metadata/taskdefs...

```
[
  {
    "ownerApp": "clmoApp",
    "name": "CLMO_FA_KYC_CHECKER",
    "description": "KYC Checker",
    "retryCount": 0,
    "timeoutPolicy": "RETRY",
    "retryLogic": "FIXED",
    "retryDelaySeconds": 5
  },
  {
```

Status 200 OK

To verify if any of the task is registered , http://whf00mdj:9009/plato-orch-service/api/metadata/taskdefs/{taskname} endpoint can be invoked with GET HTTP Method as shown below.



http://whf00gdw:8001/plato-orch-service/api/metadata/taskdefs/CLMO_FA_KYC_CHECKER

Type: No Auth

Status: 200 OK Time: 496 ms

```
{
    "ownerApp": "clmoApp",
    "createTime": 1584534555557,
    "name": "CLMO_FA_KYC_CHECKER",
    "description": "KYC Checker",
    "retryCount": 0,
    "timeoutSeconds": 0,
    "timeoutPolicy": "RETRY",
    "retryLogic": "FIXED",
    "retryDelaySeconds": 5,
    "responseTimeoutSeconds": 3600,
    "rateLimitPerFrequency": 0,
    "rateLimitFrequencyInSeconds": 1
}
```

2. Next workflow specific task definitions should be done. for that use the task definition endpoint mentioned at step 1 and provide workflows specific TaskDefinitions.JSON file content as payload.

3. Next workflow definitions should be done. for that use the workflow definition endpoint mentioned below and provide workflows definition json (e.g. LoanDraftProposal.JSON) file content as payload.

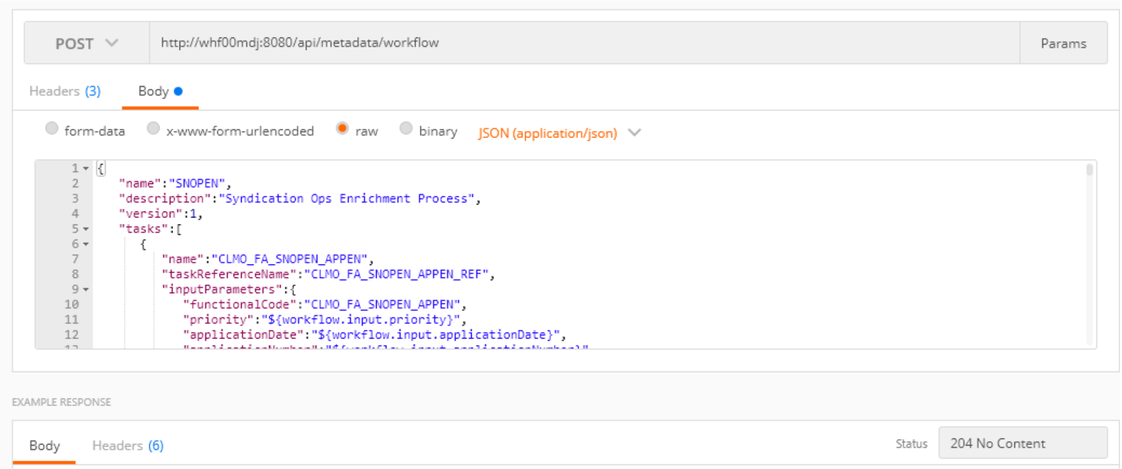
k definitions end point: http://whf00mdj:9009/plato-orch-service/api/metadata/workflow

HTTP Method: POST

Header parameters:

Content-Type:application/json
appId:platoorch
branchCode:004
userId:USER1

Payload: LoanDraftProposal.JSON content



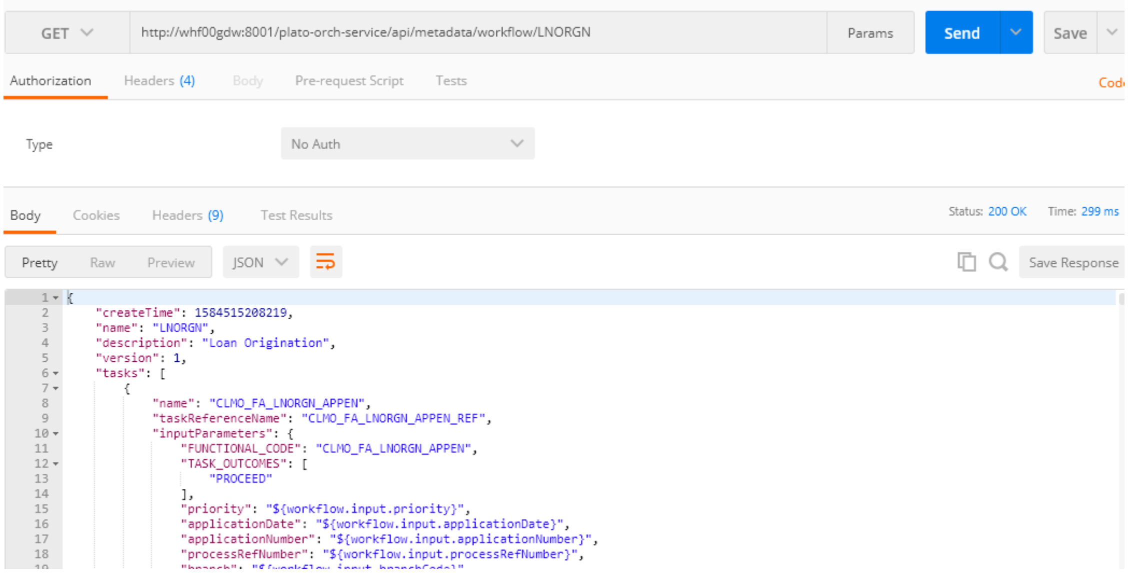
http://whf00mdj:8080/api/metadata/workflow

```
{
    "name":"SNOPEN",
    "description":"Syndication Ops Enrichment Process",
    "version":1,
    "tasks":[
      {
        "name":"CLMO_FA_SNOPEN_APPEN",
        "taskReferenceName":"CLMO_FA_SNOPEN_APPEN_REF",
        "inputParameters":{
          "functionalCode":"CLMO_FA_SNOPEN_APPEN",
          "priority":"${workflow.input.priority}",
          "applicationDate":"${workflow.input.applicationDate}",
          "applicationNumber":"${workflow.input.applicationNumber}"
```

Status 204 No Content

To verify if the workflow is registered , http://whf00mdj:9009/plato-orch-service/api/metadata/workflow/DRFTPR?version=1 endpoint can be invoked with GET HTTP Method and check if workflow definitions is available. The workflow name to be used in this GET can be referred from name property workflow definition json.

Similarly, all the workflow definitions can be registered and verified one by one.



http://whf00gdw:8001/plato-orch-service/api/metadata/workflow/LNORGN

Type No Auth

Status: 200 OK Time: 299 ms

```
{
    "createTime": 1584515208219,
    "name": "LNORGN",
    "description": "Loan Origination",
    "version": 1,
    "tasks": [
        {
            "name": "CLMO_FA_LNORGN_APPEN",
            "taskReferenceName": "CLMO_FA_LNORGN_APPEN_REF",
            "inputParameters": {
                "FUNCTIONAL_CODE": "CLMO_FA_LNORGN_APPEN",
                "TASK_OUTCOMES": [
                    "PROCEED"
                ],
                "priority": "${workflow.input.priority}",
                "applicationDate": "${workflow.input.applicationDate}",
                "applicationNumber": "${workflow.input.applicationNumber}",
                "processRefNumber": "${workflow.input.processRefNumber}",
                "branch": "${workflow.input.branchCode}"
```