Oracle® Banking Collections Cloud Service

Batch Execution Guide

Release 14.8.0.0.0
G30162-01
April 2025

ORACLE

Oracle Banking Collections Cloud Service Batch Execution Guide, Release 14.8.0.0.0

G30162-01

Primary Authors: (primary author), (primary author)

Contributing Authors: (contributing author), (contributing author)

Contributors: (contributor), (contributor)

Contents

Preface

This document helps you to understand the sequence in which the batches should be executed.

This section consists of following topics:

Purpose

This guide is designed to help acquaint you with the Batch Execution Guide application. This guide provides answers to specific features and procedures that the user need to be aware of the module to function successfully.

Audience

This guide is intended for the users of Oracle Banking Collections Cloud Service.

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 customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

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.

Related Resources

For more information, see these related Oracle resources:

- Oracle Banking Collections License Guide
- Oracle Banking Collections Security Guide
- Oracle Banking Collections Maintenance User Guide
- Oracle Banking Collections Transactions User Guide

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
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.

1

Introduction

This topic describes the information about the Introduction of batch execution sequence.

Oracle Banking Collections Cloud Service is a new generation debt collections product. It is built grounds up on micro-services architecture with focus on UI, usability, innovation, ease of integration and high performance. The application supports complete collections lifecycle - from pre-delinquency to early, mid, and late collections with outcome-focused capabilities.

This document describes the details of the batch processes required as a part of Oracle Banking Collections Cloud Service processing along with the batch execution sequence.

Business Process flow

Following steps describe the activities flow:

1. Oracle Banking Collections receives delinquent accounts from different host systems.
2. Cases are created for their primary party and accounts are linked to the case.
3. Accounts are then classified under different segments and various strategies are applied to recover the overdue amount.
4. When payments are received on these accounts, account is moved out of collections based on pre-defined criteria.
5. Once all accounts of the party are moved out of collections, Oracle Banking Collections updates status of the case as closed.

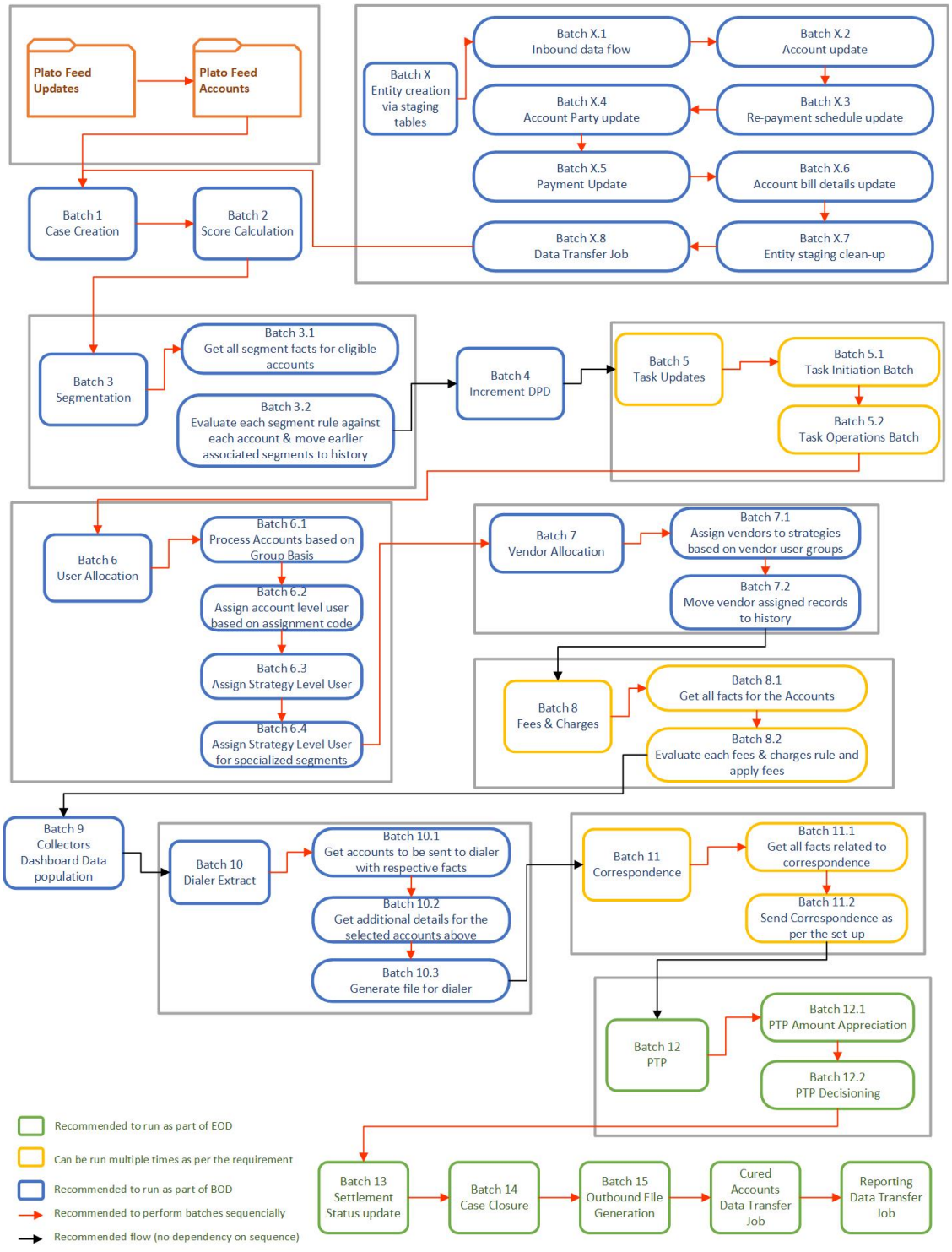
2

Batch Execution Sequence Diagram

This topic describes the information about the Batch Execution Sequence Diagram.

The below illustration explains the batch execution sequence for Oracle Banking Collections Cloud Service.

Figure 2-1 Batch_Sequence



3

Batch Execution Details

This section explains the various batch processes that are executed in Oracle Banking Collections Cloud Service once the delinquent account is onboarded.

Inbound data processing:

Oracle Banking Collections Cloud Service can receive delinquent accounts from the host system in two different ways.

- Option 1:
  Host system can send the delinquent accounts data in various files as per the agreed format. Please refer to the **File Interface Specification** document for the details. (file-interface-specification-guide.doc)

  These files can then be uploaded in Oracle Banking Collections Cloud Service to on-board the delinquent accounts.
- Option 2:
  Oracle Banking Collections Cloud Service can receive delinquent accounts from the host system in the staging area.

  Various batches will be run as mentioned below to process the delinquent account details received in the staging area to Oracle Banking Collections Cloud Service after all relevant validations.

Mandatory headers which are applicable to all batches are listed below:

Table 3-1 Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	Entity ID of the request.
Accept	application/json	Yes	Defines the expected response format.
Content-Type	application/json	Yes	Defines the request body format.
multiEntityAdmin	N	Yes	Indicates if multi-entity admin mode is enabled (Y/N).
Connection	keep-alive	Yes	Keeps the connection open for better.
Accept-Language	en-US	Yes	Requests responses Language.
env	cloud	Yes	Specifies the environment.
tenantId	nonprod	Yes	Identifies the tenant environment.
tenantPdb	dvcXXXatpXXX	Yes	Specifies the Pluggable Database (PDB) for the tenant.
tenantSvc	devtestbankXX	Yes	Defines the database service name for the tenant.

Details of batches:

- Entity Creation
  This topic provides information on Entity Creation of Batch Execution.

- Batch Execution
  This section explains the various batch processes of Oracle Banking Collections Cloud Service.

3.1 Entity Creation

This topic provides information on Entity Creation of Batch Execution.

This topic contains the following sub-topics:

- Inbound Entity Creation
  This topic provides information about the Inbound Entity Creation process
- Account Update
  This topic provides information about the Account Update process
- Re-payment Schedule Update
  This topic provides information about the Re-payment Schedule Update process.
- Account Party Update
  This topic provides information about the Account Party Update process.
- Payment Update
  This topic provides information about the Payment Update process.
- Account Bill Details Update
  This topic provides information about the Account Bill Details Update process.
- Entity Staging Clean-up
  This topic provides information about the Entity Staging Clean-up process.
- Data Transfer Job
  This topic provides information about the Data Transfer Job process.

3.1.1 Inbound Entity Creation

This topic provides information about the Inbound Entity Creation process

This batch process inbound data from staging area like account and account party relationship and persist a new account and customer relationship details entities in Oracle Banking Collections Cloud Service. Customer related details are fetched from Oracle Banking Party based on the Customer id provided by the host system.

Job Name: obcr-inbound-entity-creation-batch

Dependent Batch/Job: Not Applicable

Multi-threaded: No

URI: <protocol>://obcr-entity-services/jobLauncher/runBatch/

Mandatory Headers:

Table 3-2 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRENTITY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-3 Parameters

Parameter Name	Value	Mandatory	Description
runDate	16062019	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.

Success/Failure Details:

- User can verify the record level status in the following tables in CR_ENTITY schema. DM_INBOUND_ENTITY_CREATION_DRIVER
- For a successful processing of a record, the below condition should be satisfied. ERROR_CODE = **Success** and PROCESS_RESULT = **1**

3.1.2 Account Update

This topic provides information about the Account Update process

This batch will process all the accounts received as an update from the source system in the staging area. These accounts should already be existing in the Oracle Banking Collections Cloud Service. Account will be updated in Oracle Banking Collections Cloud Service after all the validations.

Job Name: obcr-account-update-batch

Dependent Batch/Job: Not Applicable

Multi-threaded: Yes

Multi-stream: Yes

URI: <protocol>://obcr-entity-services/jobLauncher/runBatch/

Table 3-4 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRENTITY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-5 Parameters

Parameter Name	Value	Mandatory	Description
runDate	16062019	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.

Success/Failure Details:

- User can verify the record level status in the following tables in CR_ENTITY schema. DM_ACCOUNT_UPDATE_DRIVER
- For a successful processing of a record, the below condition should be satisfied. ERROR_CODE = **Success** and PROCESS_RESULT = **1**

3.1.3 Re-payment Schedule Update

This topic provides information about the Re-payment Schedule Update process.

This batch will process re-payment schedule received for accounts from the source system, in the staging area, for an account already exists in Oracle Banking Collections Cloud Service.

If any existing re-payment schedule already exists for an account then this batch will delete the existing re-payment schedule and update the revised re-payment schedule as received in the staging area.

Job Name: obcr-repaysch-update-batch

Dependent Batch/Job: Not Applicable

Multi-threaded: Yes

Multi-stream: Yes

URI : <protocol>://obcr-entity-services/jobLauncher/runBatch/

Table 3-6 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRENTITY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-7 Parameters

Parameter Name	Value	Mandatory	Description
runDate	16062019	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.

Success/Failure Details:

- User can verify the record level status in the following tables in CR_ENTITY schema. DM_REPAYSCH_UPDATE_DRIVER
- For a successful processing of a record, the below condition should be satisfied. ERROR_CODE=**Success** and PROCESS_RESULT = **1**

3.1.4 Account Party Update

This topic provides information about the Account Party Update process.

This batch will process all the updates received from the source system, in the staging area, related to either addition or modification of existing account party relationship details.

This batch will support create, update, and delete operations on account party relationship in Oracle Banking Collections Cloud Service.

Job Name: obcr-account-party-update-batch

Dependent Batch/Job: Not Applicable

Multi-threaded: Yes

Multi-stream: Yes

URI: <protocol>://obcr-entity-services/jobLauncher/runBatch/

Mandatory Headers:

Table 3-8 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRENTITY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-9 Parameters

Parameter Name	Value	Mandatory	Description
runDate	16062019	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.

Success/Failure Details:

- User can verify the record level status in the following tables in CR_ENTITY schema. DM_ACCOUNT_PARTY_REL_UPDATE_DRIVER
- For a successful processing of a record, the below condition should be satisfied. ERROR_CODE = **Success** and PROCESS_RESULT = **1**

3.1.5 Payment Update

This topic provides information about the Payment Update process.

This batch will process new payment transactions received from source system, in the staging area, for an account already exists in Oracle Banking Collections Cloud Service.

This batch will not support either update or delete of already existing payment transactions in Oracle Banking Collections Cloud Service.

Job Name: obcr-payment-update-batch

Dependent Batch/Job: Not Applicable

Multi-threaded: Yes

Multi-stream: Yes

URI: <protocol>://obcr-entity-services/jobLauncher/runBatch/

Mandatory Headers:

Table 3-10 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRENTITY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-11 Parameters

Parameter Name	Value	Mandatory	Description
runDate	16062019	Yes	It must be in ddmmyyyy format.
branchCode	GB2	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.

Success/Failure Details:

- User can verify the record level status in the following tables in CR_ENTITY schema. DM_PAYMENT_UPDATE_DRIVER
- For a successful processing of a record, the below condition should be satisfied. ERROR_CODE = **Success** and PROCESS_RESULT = **1**

3.1.6 Account Bill Details Update

This topic provides information about the Account Bill Details Update process.

This batch will process newly received bill details or modification to any existing bill details in the staging area from the source system and persist bill details in Bill detail entity.

This batch will support create, update, and delete operations on account bill details in Oracle Banking Collections Cloud Service.

Job Name: obcr-bill-details-update-batch

Dependent Batch/Job: Not Applicable

Multi-threaded: Yes

Multi-stream: Yes

URI: <protocol>://obcr-entity-services/jobLauncher/runBatch/

Mandatory Headers:

Table 3-12 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRENTITY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-13 Parameters

Parameter Name	Value	Mandatory	Description
runDate	16062019	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.

Success/Failure Details:

- User can verify the record level status in the following tables in CR_ENTITY schema. DM_BILL_DETAIL_UPDATE_DRIVER
- For a successful processing of a record, the below condition should be satisfied. ERROR_CODE = **Success** and PROCESS_RESULT = **1**

3.1.7 Entity Staging Clean-up

This topic provides information about the Entity Staging Clean-up process.

This batch will clean up the data from the staging area like account, account party, payment, re-payment, bill details and will make staging area ready for the next batch run.

Prerequisite: All staging table for cleanup activity to be configured in lookup for lookup code **ENTITY_STAGING_TABLES**

Job Name: obcr-entity-staging-cleanup-batch

Dependent Batch/Job: Not Applicable

Multi-threaded: No

Multi-stream: No

URI: <protocol>://obcr-entity-services/jobLauncher/runBatch/

Mandatory Headers:

Table 3-14 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRENTITY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-15 Parameters

Parameter Name	Value	Mandatory	Description
runDate	16062019	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.

Success/Failure Details:

- User can verify the record level status in the following tables in CR_ENTITY schema. DM_ENTITY_STAGING_DRIVER

- For a successful processing of a record, the below condition should be satisfied. ERROR_CODE = **Success** and PROCESS_RESULT = **1**

3.1.8 Data Transfer Job

This topic provides information about the Data Transfer Job process.

All the records processed in the above batches with its status i.e success or failure (with failure reason) will be informed to the source system via this job. Source system can then analyze the records with failed status, correct the same and re-send again for processing.

This Job having two categories such as app configuration and trigger endpoint. Through the App configuration job, source and destination schema tables and column details are mapped. And Trigger endpoint job will transfer data from the source driver tables to destination system.

Table 3-16 Mandatory Headers for Configuration endpoint:

Parameter Name	Value	Mandatory	Description
appId	PLATOFDT	Yes	The application ID of the host service.

Table 3-17 Mandatory Headers for Trigger endpoint:

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRDATAEXCHNG	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	Entity ID of the request.
Content-Type	application/json	Yes	Content Type of the request.

Source Schema: CR_ENTITY

Source tables:

- DM_INBOUND_ENTITY_CREATION_DRIVER
- DM_ACCOUNT_UPDATE_DRIVER
- DM_ACCOUNT_PARTY_REL_UPDATE_DRIVER
- DM_PAYMENT_UPDATE_DRIVER
- DM_REPAYSCH_UPDATE_DRIVER
- DM_BILL_DETAIL_UPDATE_DRIVER

Destination Schema: As provided by the host system.

Destination Tables: As provided by the host system.

Specify following parameters while executing this batch:

Table 3-18 Data Transfer sample configuration:

Endpoint	Configurations
Host location	http://host:port/plato-fast-data-transfer-service/configurations/applications/ObrlDataResponse/ObcrDataAck
Method	POST
Header	appId:PLATOFDT entityId:DEFAULTENTITY Content-Type:application/json branchCode:BCR userId:AUTOUSER

Table 3-18 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
Request Body	{ "sourceJNDI": "jdbc/CR_ENTITY", "destinationJNDI": "jdbc/SLPRCORE", "continueOnErrorFlag": "N", "ignoreFailedDropsFlag": "N", "readPageSize": "1000", "tablesPerJob": "6", "chunkSize": "100", "jobWriterTypeFlag": "J", "deployScriptWhileCreate": "N", "truncateFlag": "Y", "srcTruncateFlag": "N", "streamsPerTable": "2", "customConfigMasters": [{ "srcTable": "DM_INBOUND_ENTITY_CREATION_DRIVER", "destTable": "SLP_TM_IN_COLL_RESPONSE_DATA", "customConfigDatas": [{ "key": "ACCOUNT_NBR", "value": "CONTRACT_REF_NO", "configType": "COLUMN_MAPPING" }, { "key": "BRANCH_CODE", "value": "BRANCH_CODE", "configType": "COLUMN_MAPPING" }, { "key": "RUN_DATE", "value": "RUN_DATE", "configType": "COLUMN_MAPPING" }, { "key": "SEQ_NO", "value": "SEQ_NO", "configType":

Table 3-18 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations

	"COLUMN_MAPPING"
	},
	{
	"key":
	"ERROR_CODE",
	"value":
	"ERROR_CODE",
	"configType":
	"COLUMN_MAPPING"
	},
	{
	"key":
	"PROCESS_RESULT",
	"value":
	"PROCESS_RESULT",
	"configType": "COLUMN_MAPPING"
	}
	]
	}
	}

Trigger endpoint

Table 3-19 Trigger endpoint

Endpoint	Configurations
Host location	http:// host:port/obcr-data-exchange-services/transfers/ ObrlDataResponse/ObcrDataAck? runDate=02042018&operation=T&branchCode=BCR
Method	POST
Header	appId:PLATOFDT entityId:DEFAULTENTITY Content-Type:application/json branchCode:BCR userId:AUTOUSER

Table 3-19 (Cont.) Trigger endpoint

Endpoint	Configurations

Request Body	<pre>[
	    {
	        "tableName": "SLP_TM_IN_COLL_RESPONSE_DATA",
	        "truncateFlag": "N",
	        "srcTableName":
	"DM_INBOUND_ENTITY_CREATION_DRIVER",
	        "srcTruncateFlag": "N",
	        "srcWhereClause": ""
	    }
	}</pre>

3.2 Batch Execution

This section explains the various batch processes of Oracle Banking Collections Cloud Service.

This topic contains the following sub-topics:

- Case Creation Batch
  This topic provides information about the Case Creation Batch process.
- Score Calculation Batch
  This topic provides information about the Score Calculation Batch process.
- Segmentation Batch
  This topic provides information about the Segmentation Batch process.
- Increment DPD Batch
  This topic provides information about the Increment DPD Batch process.
- Tasks Initiation Batch
  This topic provides information about the Tasks Initiation Batch process.
- User Allocation Batch
  This topic provides information about the User Allocation Batch process.
- Vendor Allocation Batch
  This topic provides information about the Vendor Allocation Batch process.
- Fees and Charges Batch
  This topic provides information about the Fees and Charges Batch process.
- Dashboard Data Population Batch
  This topic provides information about the Dashboard Data Population Batch process.
- Dialer Extract Batch
  This topic provides information about the Dialer Extract Batch process.
- Correspondence Batch
  This topic provides information about the Correspondence Batch process.
- Promise Tracking Batch
  This topic provides information about the Promise Tracking Batch process.

- Settlement Status Update Batch
  This topic provides information about the Settlement Status Update Batch process.
- Case Closure Batch
  This topic provides information about the Case Closure Batch process.
- Outbound File Generation Batch
  This topic provide information about the Outbound File Generation Batch process.
- Cured Account Data Transfer Job
  This topic provides information about the Cured Account Data Transfer Job process.
- Reporting Data Population Job
  This topic provides information about the Reporting Data Population Job process.

3.2.1 Case Creation Batch

This topic provides information about the Case Creation Batch process.

When a new account comes into collections, case creation batch will create a case on the primary party linked to this account based on certain conditions.

Case is always created on the Customer. For all new accounts received in collections, system will first check whether any existing active case is running on the primary Customer of the account. If yes, then the account will be linked to the same Case number, else, a new case number will be generated and account will be linked to it. At any given point of time, only a single active case will be running on a Customer across all product processors.

Setup Prerequisites

Following is the prerequisite for the setup:

New Accounts are available in Oracle Banking Collections Cloud Service though inbound file processing.

Job Name: obcrCaseCreateBatch

Dependent Batch/Job: Not Applicable

Multi-threaded: No

URI: <protocol>://obcr-strategy-services/jobLauncher/run/

Mandatory Headers:

Table 3-20 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRSTRTGY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-21 Parameters

Parameter Name	Value	Mandatory	Description
microServiceName	obcr-strategy-services	Yes	Service that consists the case creation batch.

Table 3-21 (Cont.) Parameters

Parameter Name	Value	Mandatory	Description
appId	CRSTRTGY	Yes	The application ID of the host service.
contextRoot	obcr-strategy-services	Yes	Context path of the service.
timestamp	YYYYMMDDHHMMSS.SSSSS	Yes	Each batch run should have unique parameters so the current timestamp is to be added.
collStartDt	YYYY-MM-DD	No	Optional parameter Collection Start Date of the accounts to be mapped. If no date is passed Application Date is considered.

Success/Failure Details:

- Fetch the latest Job Instance Id from the table PLATO_BATCH_JOB_INSTANCE in PLATOBATCH schema with the job name obcrCaseCreateBatch.
- Check the Job_instance_id in PLATO_BATCH_JOB_EXECUTION the status and exit_code will be marked COMPLETED.
- Additionally, verify the entries in DM_CASE and DM_CASE_ACCOUNT_ASSO tables in CR_STRTGY schema for delinquent accounts.

3.2.2 Score Calculation Batch

This topic provides information about the Score Calculation Batch process.

Borrowers' risk or behavior scores are used by banks and financial institutions to define the right treatment strategy in collections. The **Behavior score** or **Risk score** is a clear indication of borrowers capacity or ability to payback his overdue amount. The inbuilt scoring engine of Oracle Banking Collections Cloud Service uses quantitative scorecard models to calculate borrowers' behavior scores. The model can be built of borrower's multiple financial attributes and provides the application updates a borrower's account with a score generated from the model, which is built using their multiple financial attributes. Accounts can be segmented based on their risk profile using this score.

Job Name: obcr-opds-integration-batch

Dependent Batch/Job: obcrCaseCreateBatch

Multi-threaded: Yes

URI: <protocol>://obcr-entity-services/jobLauncher/runBatch/

Table 3-22 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRENTITY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-23 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.

Success/Failure Details:

Verify the record level status in DM_ACCOUNT_OPDS_DRIVER table in CR_ENTITY schema. For a successful processing of a record, the below condition should be satisfied.

ERROR_CODE = ‘Success’ and PROCESS_RESULT = ‘1’

3.2.3 Segmentation Batch

This topic provides information about the Segmentation Batch process.

This batch will assign either one or multiple segments to new accounts and update (add/remove) segments to existing accounts in collections. Segments will be assigned/removed based on the selections criteria maintained in segment maintenance. Each segment can have multiple active strategies. If a segment is created, then all its active strategies are created on the account. If an account already has active segment/s, however during re-segmentation, if active segment is not part of the newly identified segments, then this batch will close those segments for that account.

Segmentation batch is currently run as three sub-batches.

- **Batch 1:**
  This batch will get all segment related facts for all active accounts which are in collections and having next review date <= application date or override date.

  Job Name: obcr-segment-account-facts

  Setup Prerequisites

  Following is the prerequisite for the setup:

  Data is available in table DM_ACCOUNT_REVIEW.

  Dependent Batch/Job: Not Applicable

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>:// obcr-segmentation-batch-services/jobLauncher/runBatch/

  Table 3-24 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRSEG	Yes	The application ID of the host service.

  Specify following parameters while executing this batch:

Table 3-25 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

- **Batch 2:**
  This batch evaluates all active segment rules against each account and assigns/removes segment.

  Job Name:obcr-account-segmentation

  Setup Prerequisites

  Following is the prerequisite for the setup:

  Batch 1 which is obcr-segment-account-facts must be completed before this batch.

  Dependent Batch/Job: obcr-segment-account-facts

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>:// obcr-segmentation-batch-services/jobLauncher/runBatch/

Table 3-26 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRSEG	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-27 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.
noOfDaysForNextRun	1	No	It is required if you want to exclude accounts for few days from re-segmentation. Its default value is 1.

Success/Failure Details:

User can verify the record level status in DM_SEGMENT_ACCOUNTS and DM_SEGMENT_ACCOUNT_FACTS tables in CR_SEG schema. For a successful processing of a record, the below condition should be satisfied.

ERROR_CODE = ‘Success’ and PROCESS_RESULT = ‘1’

3.2.4 Increment DPD Batch

This topic provides information about the Increment DPD Batch process.

Oracle Banking Collections Cloud Service receives delinquent accounts data from multiple product processors via file or online services.

Last DPD Update Date is an optional field in the ‘Account Details’ entity. If the product processor sends the value in this field, the same should be updated in Oracle Banking Collections Cloud Service.

If the product processor sends blank value in this field, then in Oracle Banking Collections Cloud Service, the value should be set as current business date.

This batch will increment the DPDs for all accounts in collections by the difference between Previous Business Date and Current Business Date.It is recommended to run as part of BOD.

This batch can be switched off for specific product processor, if required.

Job Name: obcr-incrementdpd-batch

Dependent Batch/Job: Not Applicable

Multi-threaded: Yes

URI: <protocol>://obcr-entity-services/jobLauncher/runBatch/

Table 3-28 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRENTITY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-29 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.
excludePPC	TPH	No	If multiple product processor needs to be skipped, then we need to pass their codes as comma separated values.

Table 3-29 (Cont.) Parameters

Parameter Name	Value	Mandatory	Description
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

Success/Failure Details:

User can verify the record level status in DM_PPC_DRIVER table in CR_ENTITY schema. For a successful processing of a record, the below condition should be satisfied.

ERROR_CODE = ‘Success’ and PROCESS_RESULT = ‘1’

3.2.5 Tasks Initiation Batch

This topic provides information about the Tasks Initiation Batch process.

- **Batch 1:**

  1. Segmentation batch will update segment creation date for all accounts where new segments are stamped. Tasks are initiated based on the rules defined for all strategies of these newly assigned segments.
  2. Tasks for strategies and segments which are de-linked from the account (based on segment end date) are moved to history table.

  Job Name: obcr-task-init

  Setup Prerequisites

  Following is the prerequisite for the setup:

  Recommended to run after segmentation batch.

  Dependent Batch/Job: Not Applicable

  Multi-threaded: Yes

  Multi-stream: Yes

  URI: <protocol>:// obcr-task-services/jobLauncher/runBatch/

  Table 3-30 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRTASK	Yes	The application ID of the host service.

  Specify following parameters while executing this batch:

  Table 3-31 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code.

Table 3-31 (Cont.) Parameters

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	The user running the batch.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

- **Batch 2:**

1. This batch will move the status of the tasks based on the rules defined.
2. New Tasks will be opened based on the wait period, dependencies and so on.
3. Tasks will be escalated or marked as expired based on the periods specified.

Job Name: obcr-task-operation

Dependent Batch/Job: obcr-task-init

Multi-threaded: Yes

Multi-stream: Yes

URI: <protocol>:// obcr-task-services/jobLauncher/runBatch/

Table 3-32 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRTASK	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-33 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code.
userId	ADMINUSER1	Yes	The user running the batch.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

Success/Failure Details:

User can verify the record level status in DM_TASK_WRKFLW_OPERATN_DRIVR and DM_ACCOUNT_STRATEGY table in CR_TASK schema. For a successful processing of a record, the below condition should be satisfied.

ERROR_CODE = ‘Success’ and PROCESS_RESULT = ‘1’

3.2.6 User Allocation Batch

This topic provides information about the User Allocation Batch process.

This batch will assign accounts and their respective tasks to users. Based on group basis at product processor level, different types of user assignments are done as mentioned below:

- **Customer:** All the strategies & their respective tasks of all the accounts of a customer will be assigned to a single collector.
- **Accounts:** All the strategies & their respective tasks of an account will be assigned to a collector.
- **Ignoring product processor group basis (at Segment Maintenance):** Strategies & their respective tasks will be assigned to different collectors based on collector skill-set.

Use Allocation batch is run in 4 sub-batches.

- **Batch 1:**
  Function Description:

  1. Batch will consider all accounts where segmentation is not yet done and filter the accounts based on group basis i.e either Account or Customer.
  2. For Accounts where group basis is Customer and account's primary party having other accounts already assigned in collections then assign same user to this account.
  3. For Accounts where group basis is Account OR group basis is Customer where primary party of the account do not have any existing account in collections, assignment code will be stamped based on the assignment rules and priority defined.

  Job Name: obcr-account-assignment-batch

  Setup Prerequisites

  Following is the prerequisite for the setup:

  1. New Accounts available in the system with segmentation done.
  2. Re-segmentation of the existing accounts in done.

  Dependent Batch/Job: Segmentation Batch

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>:// obcr-user-management-services/jobLauncher/runBatch/

Table 3-34 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRUSERMGMT	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-35 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	The branch code of requesting branch.
userId	ADMINUSER1	Yes	The user id of the user triggering the batch.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

- **Batch 2:**
  Functional Description:

  1. Batch will consider all the accounts from batch 1 on which account assignment code was stamped.
  2. Based on the assignment method specified in the assignment code (i.e round robin method), users are assigned for these accounts.

  Job Name: obcr-account-assign-batch

  Setup Prerequisites:

  Batch 1 which is obcr-account-assignment-batch must be completed before this batch.

  Dependent Batch/Job: obcr-account-assignment-batch

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>:// obcr-user-management-services/jobLauncher/runBatch/

Table 3-36 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRUSERMGMT	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-37 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	The branch code of requesting branch.
userId	ADMINUSER1	Yes	The user ID of the user triggering the batch.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

- **Batch 3:**
  This batch will assign strategy level users to all accounts where ignore group basis is maintained as ‘No’ for a segment and stamp assignment code maintained at the segment level for all accounts where ignore group basis is ‘Yes’.

  Functional Description:

  1. This batch will consider all accounts where new segment is added or deleted during segmentation batch.
  2. For all accounts where segment ignore group basis in ‘No’ , assign the account level user already assigned in batch 2 above for all strategies of that segment.
     For all accounts where segment ignore group basis in ‘Yes’ , stamp the assignment code against each strategy as defined in the segment maintenance.

  Job Name: obcr-strategy-assignment-batch

  Setup Prerequisites:

  Following is the prerequisite for the setup:

  1. Batch 1 & Batch 2 must be completed before this batch.
  2. Data is available in output tables of segmentation batch.

  Dependent Batch/Job: obcr-account-assign-batch

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>:// obcr-user-management-services/jobLauncher/runBatch/

Table 3-38 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRUSERMGMT	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-39 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	The branch code of requesting branch.
userId	ADMINUSER1	Yes	The user ID of the user triggering the batch.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

- **Batch 4:**
  Functional Description:

  1. Batch will consider all the accounts from batch 3 on which account assignment code was stamped.

2. Based on the assignment method specified in the assignment code (i.e round robin method), users are assigned at strategy level.

Job Name: obcr-strategy-assign-batch

Setup Prerequisites:

Batch 1, Batch 2 & Batch 3 must be completed before this batch.

Dependent Batch/Job: obcr-strategy-assignment-batch

Multi-threaded: Yes

Multi-stream: Yes

URI : <protocol>:// obcr-user-management-services/jobLauncher/runBatch/

Table 3-40 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRUSERMGMT	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-41 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	The branch code of requesting branch.
userId	ADMINUSER1	Yes	The user ID of the user triggering the batch.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

Success/Failure Details:

- User can verify the record level status in the following tables in CR_USERMGMT schema.
  - DM_ACCOUNT_ALLOCATION_DRIVER
  - DM_ACCOUNT_ASSIGN_DRIVER
  - DM_STRATEGY_ASSIGNMENT_DRIVER
  - DM_STRATEGY_ASSIGN_DRIVER
- For a successful processing of a record, the below condition should be satisfied. ERROR_CODE = ‘Success’ and PROCESS_RESULT = ‘1’

3.2.7 Vendor Allocation Batch

This topic provides information about the Vendor Allocation Batch process.

Vendor Allocation Batch is currently run as two sub-batches.

- **Batch 1:**

After the completion of user allocation batch, user will be assigned at strategy level.

This batch will consider only those records from the above set of records where user id belongs to any of the vendor user group and fetch all the relevant details for these accounts and persist the data.

Job Name: obcr-vendor-account-assign-batch

Dependent Batch/Job: User Assignment

Multi-threaded: Yes

Multi-stream: Yes

URI : <protocol>:// obcr-vendor-management-services /jobLauncher/runBatch/

Table 3-42 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRVNDRMGMT	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-43 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.

- **Batch 2:**
  This batch will consider records from strategy assign history table where user id belongs to any of the vendor user group and check whether this account and user combination exists in vendor assign table. If record exists in vendor assign table then delete from this table and move it to vendor account assign history table.

  Job Name: obcr-vendor-account-history-batch

  Dependent Batch/Job: obcr-vendor-account-facts

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>: //obcr-vendor-management-services/jobLauncher/runBatch/

Table 3-44 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRVNDRMGMT	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-45 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.

Success/Failure Details:

User can verify the record level status in the following tables in CR_VNDRMGMT schema.

- DM_VENDOR_ACCOUNT_ASSIGN_DRIVER
- DM_VENDOR_ACCOUNT_ASSIGN_HISTORY_DRIVER

3.2.8 Fees and Charges Batch

This topic provides information about the Fees and Charges Batch process.

Oracle Banking Collections Cloud Service has the capability to apply fees and charges on the accounts based on the predefined conditions.

This batch currently runs as two sub-batches.

- **Batch 1:**
  This batch will get all fees and charges related facts for all active accounts which are in collections.

  Job Name: obcr-feescharges-account-facts

  Setup Prerequisites

  NA

  Dependent Batch/Job: Not Applicable

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>:// obcr-fees-charges-services/jobLauncher/runBatch/

Table 3-46 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRFEECHRG	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-47 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.

Table 3-47 (Cont.) Parameters

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User id of the user.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

- **Batch 2:**
  This batch evaluates all active fees and charges rules against each account and apply the appropriate fees and charges on the applicable accounts.

  Job Name: obcr-feescharges-calculation

  Setup Prerequisites

  Following is the prerequisite for the setup:

  Batch 1 which is obcr-feescharges-account-facts must be completed before this batch.

  Dependent Batch/Job: obcr-feescharges-account-facts

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>://obcr-fees-charges-services/jobLauncher/runBatch/

Table 3-48 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRFEECHRG	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-49 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

Success/Failure Details:

User can verify the record level status in DM_FEECHARGE_ACCOUNTS_DRIVER and DM_FEECHARGE_ACCOUNTS_FACTS_DRIVER tables in CR_FEECHRG schema. For a successful processing of a record, the below condition should be satisfied.

ERROR_CODE = ‘Success’ and PROCESS_RESULT = ‘1’

3.2.9 Dashboard Data Population Batch

This topic provides information about the Dashboard Data Population Batch process.

Oracle Banking Collections Cloud Service has the capability to calculate the historical data and key performance indicators to be displayed on the collector's dashboard.

Job Name: obcr-dashboard-data-population

Setup Prerequisites

NA

Dependent Batch/Job: Not Applicable

Multi-threaded: Yes

Multi-stream: Yes

URI : <protocol>://obcr-dashboard-services/jobLauncher/runBatch/

Table 3-50 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRDASHBOARD	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-51 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.
pastDays	180	No	Number of past days to be considered while calculating historical and performance data of collectors.

Success/Failure Details:

User can verify the record level status in DM_COLLECTOR_DASHBOARD_DRIVER table in CR_DASHBOARD schema. For a successful processing of a record, the below condition should be satisfied.

ERROR_CODE = 'Success' and PROCESS_RESULT = '1'

3.2.10 Dialer Extract Batch

This topic provides information about the Dialer Extract Batch process.

- **Batch 1:**
  Dialer Account Filter batch:
  - This batch will pick up all accounts from DM_TASK table that have call type tasks either in Open or WIP status and follow-up date for these tasks is as per the parameter defined.
  - In the next step fact values are fetched for each of the above account and then filtered further based on rules defined for the selection criteria.
  - These accounts are then saved.

  Job Name: obcr-dialer-accts-filter-batch

  Dependent Batch/Job: Not Applicable

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>://obcr-data-exchange-services/jobLauncher/runBatch/

Table 3-52 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRDATAEXCHNG	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-53 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	The branch code of the project.
userId	ADMINUSER1	Yes	The user ID of the user triggering the batch.
followUpDays	1	No	Number of Maximum Follow-up days (On Task) from Current Business Date to be considered.
campaignCodes	SEGMENT_EC	Yes	The campaign code will be the condition(rule) which needs to be evaluated to identify the accounts to be sent to dialer system. Multiple comma separated rules are supported.

Table 3-53 (Cont.) Parameters

Parameter Name	Value	Mandatory	Description
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

- **Batch 2:**
  Dialer Account Filter batch:
  - Additional data is fetched by making API calls to different services for all the accounts of batch 1 (Dialer account filter batch) above.
  - Data persisted in the DB.

  Job Name:obcr-dialer-data-extract-batch

  Dependent Batch/Job: obcr-dialer-accts-filter-batch

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>://obcr-data-exchange-services/jobLauncher/runBatch/

Table 3-54 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRDATAEXCHNG	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-55 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	The branch code of the project.
userId	ADMINUSER1	Yes	The user ID of the user triggering the batch.

- **Batch 3:**
  Dialer File Generation Batch:
  - Account and Customer numbers are picked for all accounts stored in batch 2 (Dialer Data Extract Batch) to store in the Driver table (DM_DIALER_FILEGEN_DRIVER) Rest of the details are fetched from DM_DIALER_EXTRACT_DATA for each account in the driver.
  - All the records along with column headers are stored in the file as per the details mentioned in the properties file.

  Job Name: obcr-dialer-file-gen-batch

  Dependent Batch/Job: obcr-dialer-data-extract-batch

  Multi-threaded: No

Multi-stream: No

URI : <protocol>://obcr-data-exchange-services/jobLauncher/runBatch/

Table 3-56 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRDATAEXCHNG	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-57 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	The branch code of the project.
userId	ADMINUSER1	Yes	The user ID of the user triggering the batch.

Success/Failure Details:

User can verify the record level status in the following tables in CR_DATAEXCHNG schema.

- DM_DIALER_ACCTS_DRIVER
- DM_DIALER_FILEGEN_DRIVER

For a successful processing of a record, the below condition should be satisfied.
ERROR_CODE = ‘Success’ and PROCESS_RESULT = ‘1’

3.2.11 Correspondence Batch

This topic provides information about the Correspondence Batch process.

Correspondence batch currently runs in two sub-batches.

- **Batch 1:**
  This batch will get all correspondence related facts for all active accounts which are in collections.

  Job Name: obcr-communication-facts-batch

  Setup Prerequisites

  Following is the prerequisite for the setup:

  Data is available in table DM_ACCOUNT from hosts.

  Dependent Batch/Job: Not Applicable

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>:// obcr- correspondence -services/jobLauncher/runBatch/

Table 3-58 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRCORR	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-59 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

- **Batch 2:**
  Job Name: obcr-communication-email-batch

  Each account is evaluated against all active correspondence template rules in this batch, and communication is performed via letter, email, SMS, or WhatsApp based on the configuration maintained.

  Setup Prerequisites

  Following is the prerequisite for the setup:

  Batch 1 which is obcr-communication-facts-batch must be completed first.

  Dependent Batch/Job: obcr-communication-facts-batch

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>:// obcr-correspondence-services/jobLauncher/runBatch/

Table 3-60 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRCORR	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-61 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.

Table 3-61 (Cont.) Parameters

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User id of the user.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.
commMode	“letter,email”	No	It is required if you want to perform certain mode of communication in particular.

Success/Failure Details:

- User can verify the record level status in the following tables in CR_CORR schema.
  - DM_COMM_ACCOUNT_DRIVER
  - DM_COMM_ACCOUNT_FACTS_DRIVER
- For a successful processing of a record, the below condition should be satisfied. ERROR_CODE = ‘Success’ and PROCESS_RESULT = ‘1’

3.2.12 Promise Tracking Batch

This topic provides information about the Promise Tracking Batch process.

Oracle Banking Collections Cloud Service has the capability to monitor all the promises taken from the customer and appropriate the payment received against these promises to mark them as kept or broken.

- **Batch 1: Payment Appropriation Batch**
  This batch will allocate the payments received from product processor against an account which have active promises based on FIFO (First in First Out) method. If amount is reversed from the product processor then payment appropriated will also be reversed, only if promise is still active.

  Job Name: obcr-ptp-appropriation-batch-services

  Dependent Batch/Job: Not Applicable

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>:// obcr-ptp-services/jobLauncher/runBatch/

Table 3-62 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRPTP	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-63 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

- **Batch 2: PTP Decisioning Batch**
  Based on the promise appropriation batch output, system will mark whether the promise is kept or broken. Follow-up date of the linked task is also updated based on the decision.

  Job Name: obcr-ptp-decisioning-batch-services

  Setup Prerequisites

  Following is the prerequisite for the setup:

  The PTP Appropriation batch must have completed its run.

  Dependent Batch/Job: Payment appropriation batch.

  Multi-threaded: Yes

  Multi-stream: Yes

  URI : <protocol>:// obcr-ptp-services/jobLauncher/runBatch/

Table 3-64 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRPTP	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-65 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code.
userId	ADMINUSER1	Yes	The user running the batch.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

Success/Failure Details:

- User can verify the record level status in the following tables in CR_PTP schema.
  - DM_PTP_DECISIONING_DRIVER

  - DM_PTP_APPR_DRIVER
- For a successful processing of a record, the below condition should be satisfied. ERROR_CODE = ‘Success’ and PROCESS_RESULT = ‘1’

3.2.13 Settlement Status Update Batch

This topic provides information about the Settlement Status Update Batch process.

This batch will consider all the accounts where settlement process is initiated i.e where settlement status is **Initiated** and update the settlement status based on the business logic defined. Any of the following settlement status can be marked on the account

- Offer Generated
- Partially Settled
- Fully Settled
- Not Settled

Job Name: obcr-account-settlement-batch

Dependent Batch/Job: obcr-ptp-decisioning-batch-services

Multi-threaded: Yes

Multi-stream: Yes

URI : <protocol>://obcr-settlement-services/jobLauncher/runBatch/

Table 3-66 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRSETLMNT	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-67 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	The user ID of the user triggering the batch.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

Success/Failure Details:

User can verify the record level status in DM_SETTLEMENT_OFFER_DRIVER table in CR_SETLMNT schema. For a successful processing of a record, the below condition should be satisfied.

ERROR_CODE = ‘Success’ and PROCESS_RESULT = ‘1’

3.2.14 Case Closure Batch

This topic provides information about the Case Closure Batch process.

Oracle Banking Collections Cloud Service has the capability to move the active account out of collections based on certain conditions (based on pre-defined rule) and also capability to close the existing cases.

Job Name: obcr-caseclosure-batch

Setup Prerequisites

Following is the prerequisite for the setup:

The facts and rules are created before batch run.

Dependent Batch/Job: Not Applicable

Multi-threaded: Yes

Multi-stream: Yes

URI : <protocol>:// obcr-entity-services/jobLauncher/runBatch/

Table 3-68 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRENTITY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-69 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.
filterId	“ACCT_CURED”	Yes	It is the rule, based on which accounts will be classified as cured.

Table 3-69 (Cont.) Parameters

Parameter Name	Value	Mandatory	Description
exclusionRuleId	“Segment_Code”	No	It is rule to define which accounts needs to be excluded from processing. For example, in the list 10 accounts are classified where overdue_amount is Zero, however out of which we need to exclude accounts where segment is ‘Bankruptcy’ , this rule can be defined and used.

Success/Failure Details:

User can verify the record level status in DM_ACCOUNT_CLOSURE_DRIVER table in CR_ENTITY schema. For a successful processing of a record, the below condition should be satisfied.

ERROR_CODE = ‘Success’ and PROCESS_RESULT = ‘1’

3.2.15 Outbound File Generation Batch

This topic provide information about the Outbound File Generation Batch process.

Oracle Banking Collections Cloud Service has the capability to extract the data of the cured accounts and share it with the product processor in the form of pre-defined file format.

Job Name: obcr-outboundFileGeneration-batch

Setup Prerequisites

Dependent Batch/Job: Case Closure Batch

Multi-threaded: Yes

Multi-stream: Yes

URI : <protocol>:// obcr-entity-services/jobLauncher/runBatch/

Table 3-70 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appId	CRENTITY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-71 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User id of the user.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.
ProductProcessor	“TPH”	No	Specific product processor code for which the outbound file is to be generated.
filePath	"/scratch/work_area/OutFile"	Yes	It is the path on the server where the outbound file needs to be generated.

Success/Failure Details:

User can verify the record level status in DM_ACCOUNT_OUTBOUND_DRIVER table in CR_ENTITY schema. For a successful processing of a record, the below condition should be satisfied.

ERROR_CODE = ‘Success’ and PROCESS_RESULT = ‘1’

3.2.16 Cured Account Data Transfer Job

This topic provides information about the Cured Account Data Transfer Job process.

This job is used to transfer cured account details to the source system. Source system can then analyze the data and process the cured accounts.

This Job having two categories such as app configuration and trigger endpoint. Through the App configuration job, source and destination schema tables and column details are mapped and Trigger endpoint job will transfer data from the source driver table to destination system.

Configuration for this job is similar to what is explained for data transfer job in the section Data Transfer Job.

Mandatory Headers:

Table 3-72 Mandatory Headers for Configuration endpoint:

Parameter Name	Value	Mandatory	Description
appId	PLATOFDT	Yes	The application ID of the host service.

Table 3-73 Mandatory Headers for Trigger endpoint:

Parameter Name	Value	Mandatory	Description
appId	CRDATAEXCHNG	Yes	The application ID of the host service.

Source Schema: CR_ENTITY

Source table: DM_CURED_ACCOUNT

Destination Schema: As provided by the host system.

Destination Tables: As provided by the host system.

3.2.17 Reporting Data Population Job

This topic provides information about the Reporting Data Population Job process.

This job is used to populate data from various tables across schemas into CR_DATAEXCHNG schema, which can then be used to generate various reports.

Configuration for this job is similar to what is explained for data transfer job above.

Mandatory Headers:

Table 3-74 Mandatory Headers for Configuration endpoint:

Parameter Name	Value	Mandatory	Description
appId	PLATOFDT	Yes	The application ID of the host service.

Table 3-75 Mandatory Headers for Trigger endpoint:

Parameter Name	Value	Mandatory	Description
appId	CRDATAEXCHNG	Yes	The application ID of the host service.

Details of source and target schema and tables are mentioned below:

Table 3-76 source and target schema

Source-SCHEMA NAME	Source-TABLE NAMES	Destination-SCHEMA NAME	Destination-TABLE NAMES
CR_ACTION	DM_ACCOUNT_COLLECTION_STATUS	CR_DATAEXCHNG	DM_RPT_ACCOUNT_COLLECTION_STATUS
	DM_ACCOUNT_COLLECTION_STATUS_HISTORY	CR_DATAEXCHNG	DM_RPT_ACCOUNT_COLLECTION_STATUS_HISTORY
	DM_ACTION_RESULTS	CR_DATAEXCHNG	DM_RPT_ACTION_RESULTS
CR_ACTIVITY	DM_ACTIVITY	CR_DATAEXCHNG	DM_RPT_ACTIVITY

Table 3-76 (Cont.) source and target schema

Source-SCHEMA NAME	Source-TABLE NAMES	Destination-SCHEMA NAME	Destination-TABLE NAMES
CR_DASHBOARD	DM_KPI	CR_DATAEXCHNG	DM_RPT_KPI
	DM_KPI_HIST	CR_DATAEXCHNG	DM_RPT_KPI_HIST
CR_ENTITY	DM_ACCOUNT	CR_DATAEXCHNG	DM_RPT_ACCOUNT
	DM_ACCOUNT_DEL_HISTORY	CR_DATAEXCHNG	DM_RPT_ACCOUNT_DEL_HISTORY
	DM_ACCOUNT_PARTY	CR_DATAEXCHNG	DM_RPT_ACCOUNT_PARTY
	DM_COLLATERAL	CR_DATAEXCHNG	DM_RPT_COLLATERAL
	DM_COLLATERAL_LINKAGE	CR_DATAEXCHNG	DM_RPT_COLLATERAL_LINKAGE
	DM_CURED_ACCOUNT	CR_DATAEXCHNG	DM_RPT_CURED_ACCOUNT
	DM_CURED_ACCOUNT_HIST	CR_DATAEXCHNG	DM_RPT_CURED_ACCOUNT_HIST
	DM_PARTY	CR_DATAEXCHNG	DM_RPT_PARTY
	DM_PARTY_NAME	CR_DATAEXCHNG	DM_RPT_PARTY_NAME
	DM_PAYMENT	CR_DATAEXCHNG	DM_RPT_PAYMENT
CR_FEECHRG	DM_FEESCHARGES_ACCT_ASSO	CR_DATAEXCHNG	DM_RPT_FEESCHARGES_ACCT_ASSO
	DM_FEESCHARGES_ACCT_ASSO_HIST	CR_DATAEXCHNG	DM_RPT_FEESCHARGES_ACCT_ASSO_HIST
CR_PTP	DM_PTP	CR_DATAEXCHNG	DM_RPT_PTP
	DM_PTP_HIST	CR_DATAEXCHNG	DM_RPT_PTP_HIST
CR_SEG	DM_SEGMENT_ACCOUNT_ASSO	CR_DATAEXCHNG	DM_RPT_SEGMENT_ACCOUNT_ASSO
	DM_SEGMENT_ACCOUNT_ASSO_HIST	CR_DATAEXCHNG	DM_RPT_SEGMENT_ACCOUNT_ASSO_HIST
	DM_SEGMENT_STGY_ACCOUNT_ASSO	CR_DATAEXCHNG	DM_RPT_SEGMENT_STGY_ACCOUNT_ASSO
CR_STRTGY	DM_CASE	CR_DATAEXCHNG	DM_RPT_CASE
	DM_CASE_ACCOUNT_ASSO	CR_DATAEXCHNG	DM_RPT_CASE_ACCOUNT_ASSO
	DM_CASE_ACCOUNT_ASSO_HIST	CR_DATAEXCHNG	DM_RPT_CASE_ACCOUNT_ASSO_HIST
	DM_CASE_HIST	CR_DATAEXCHNG	DM_RPT_CASE_HIST

Table 3-76 (Cont.) source and target schema

Source-SCHEMA NAME	Source-TABLE NAMES	Destination-SCHEMA NAME	Destination-TABLE NAMES
CR_TASK	DM_TASK	CR_DATAEXCHNG	DM_RPT_TASK
	DM_TASK_ASSIGN_HIST	CR_DATAEXCHNG	DM_RPT_TASK_ASSIGN_HIST
	DM_TASK_HISTORY	CR_DATAEXCHNG	DM_RPT_TASK_HISTORY
CR_USERMGMT	DM_ACCOUNT_ASSIGN	CR_DATAEXCHNG	DM_RPT_ACCOUNT_ASSIGN
	DM_ACCOUNT_ASSIGN_HISTORY	CR_DATAEXCHNG	DM_RPT_ACCOUNT_ASSIGN_HISTORY
	DM_STRATEGY_ASSIGN	CR_DATAEXCHNG	DM_RPT_STRATEGY_ASSIGN
	DM_STRATEGY_ASSIGN_HISTORY	CR_DATAEXCHNG	DM_RPT_STRATEGY_ASSIGN_HISTORY
	DM_TM_USER_GROUP	CR_DATAEXCHNG	DM_TM_RPT_USER_GROUP
	DM_TM_USER_GROUP_MEMBERS	CR_DATAEXCHNG	DM_TM_RPT_USER_GROUP_MEMBERS

4

Batch Configuration Details

This topic provide information about the Batch Configuration Details.

This topic contains the following sub-topics:

4.1 Defining workflow using Netflix conductor

Netflix Conductor is a popular and widely used open source orchestration engine. It uses JSON-based DSL (short for, **domain-specific language**) to define the workflows and workflow steps (tasks). These simple/worker tasks are implemented by application(s) and run in a separate environment from Conductor. These tasks talk to Conductor server through REST client and provides the following notable features: Provides visibility and traceability of workflows.

Below is a standard example of a workflow DSL.

- The tasks section of a DSL is an array of batch jobs to be executed in sequence.
- In the input Parameters section **http_request** to be passed. So, this would contain the standard batch request contents for example, url, headers, body etc.
- Certain set of parameters like branchCode, userId are obtained from the input of request from where this DSL would be invoked to run. Hence, they are referred to as $ {workflow.input.xxx}.
- Certain parameters like runDate are obtained as output from any tasks, preceding the task which is using it.

Standard DSL

```
{     "name": "TaskBatchJobsWf",
               "description": "TaskBatchJobsWf puts the jobs in the order of
execution sequence.",
               "tasks": [
                {
                "name": "obcr-task-init",
                "taskReferenceName": "obcr-task-init",
                "inputParameters": {
                    "http_request": {
                        "connectionTimeOut": "36000000",
                        "readTimeOut":"36000000",
                        "vipAddress":"obcr-task-services",
                        "uri":"/obcr-task-services/jobLauncher/runBatch/",
                        "method":"POST",
```

```
                "headers": {
                    "appId": "CRTASK",
                    "branchCode":"${workflow.input.branchCode}",
                    "userId":"${workflow.input.userID}",
                    "entityId":"DEFAULTENTITY",
                    "Accept":"application/json"
                 },
                    "body": {
                      "jobName":"obcr-task-init",
                      "jobParameters":[
                          {
                            "key": "runDate",
                            "value":"${workflow.variables.runDate}"
                          },
                           {
                             "key": "branchCode",
                             "value":"${workflow.input.branchCode}"
                            },
                            {
                              "key": "userId",
                              "value":"${workflow.input.userID}"
                             }
                             ]
                          }
                        }
                      },
                       "type": "HTTP",
                       "startDelay": 0,
                       "optional": false,
                       "asyncComplete": false
                  }
              ],
              "schemaVersion": 2,
              "restartable": true,
              "workflowStatusListenerEnabled": false
}
```

Below is the SOP to upload and execute the DSL.

- A standard workflow DSL which includes BOD, MOD & EOD workflows has been provided in the release package.
- Please upload this DSL to the conductor server via the app-shell. (Need to find the exact page as we have not followed this way).
- Next, navigate to Core Maintenance → Branch EOD → Configure EOD and add this workflow to the EOD.
- Once configured, please navigate to Core Maintenance → Branch EOD → Invoke EOD, enter the branch code and start the workflow by clicking on start.
- Further steps are provided in Success and Failure Analysis section (refer 2.4.4).
  The above steps can be followed through the attached Postman collections as well. Batch Sequencing Guide.postman_collection.json.

  Execute the 'Post Workflow' postman request to upload the DSL to the conductor server. The DSL would be passed in the body of the request.

- Next, navigate to Core Maintenance → Branch EOD → Configure EOD and add this workflow to the EOD.
- Once configured, please execute the 'Initiate Workflow' postman request to execute the DSL. The request would have branchCode and userId in the request body and the workflow name as the URL path variable.

4.2 Pre-requisites to deploy conductor process

PLATO-O and **PLATO-ORCH-SERVICE** services should be up and registered in the Eureka registry.

For the installation of **PLATO-O** and **PLATO-ORCH-SERVICE**, refer to Oracle Banking Microservices Platform Foundation Installation Guide.

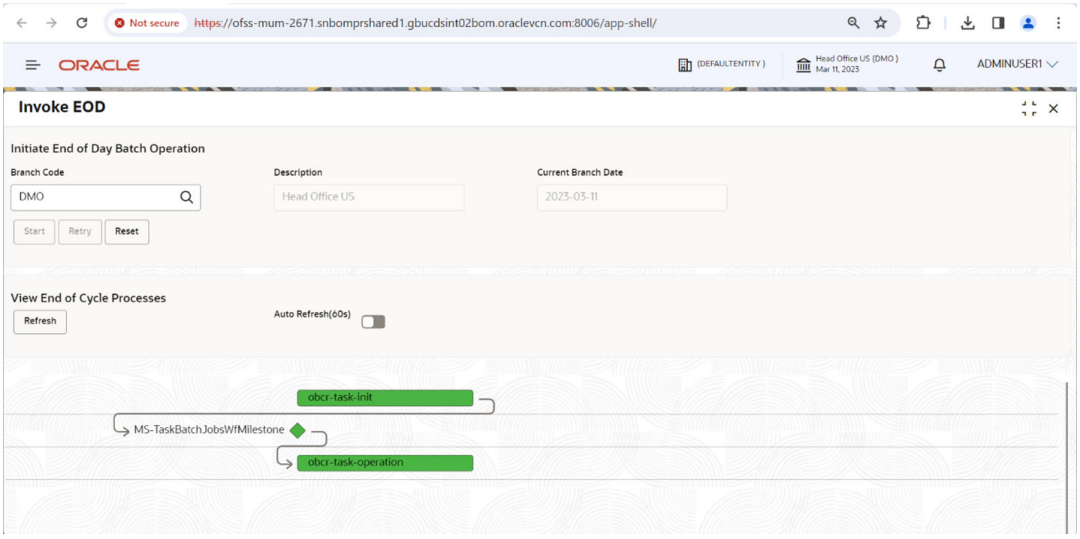
4.3 Branch EOD configuration through common core maintenance

For configuring and invoking branch EOD through common core, refer to Oracle Banking Common Core User Guide.

4.4 Success and Failure analysis

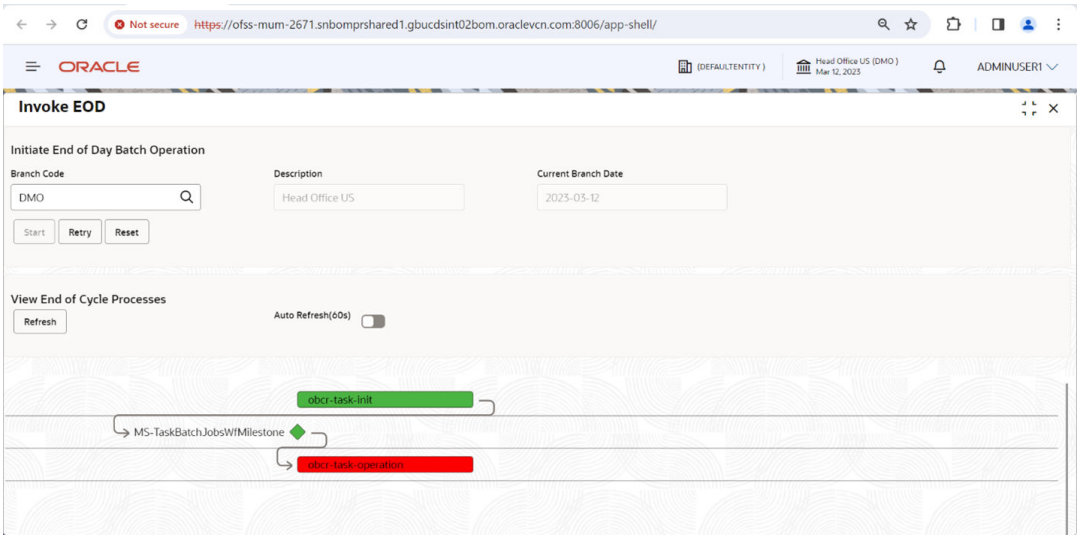
1. Navigate to the branch EOD screen. Core Maintenance → Barnch EOD → Invoke EOD.
2. Enter the branch code.
3. If the batches have successfully run for that day, below is how it would appear. Here, **obcr-task-init** and **obcr-task-operation** are two batches which have completed successfully, hence both are showing green.

Figure 4-1 Invoke EOD - Branch Code



4. If there is any failure in the batch, below is how it would appear. Here, **obcr-task-operation** batch has failed, hence it shows in red.

Figure 4-2 Invoke EOD - Processing



In this case, please visit the PLATO_BATCH schema, and execute the below query.

```
Select * from PLATO_BATCH_STREAM_RESULTS where JOB_NAME =
'<FAILED_JOB_NAME>';
```

Here, you would be able to find out which of the streams of the job are in status **A** (Aborted) or **S** (Suspended).

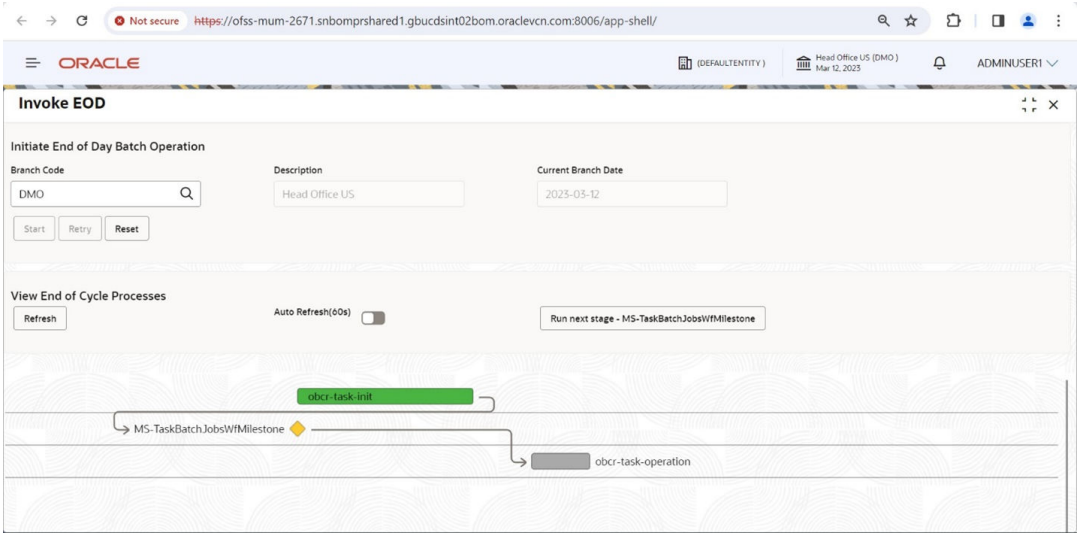
a. If any of the streams are in **A** status then have a look at the logs for the service, fix the bug and just run the batches again. It would pick-up the execution from the same point where it was aborted.

b. If any of the streams are in **S** status then follow below steps.

- Have a look at the logs and fix the issue.
- Delete all records from **PLATO_BATCH_STREAM_RESULTS** table where **JOB_NAME = '<FAILED_JOB_NAME>';**
- Check the **PLATO_BATCH_STEP_EXECUTION** and **PLATO_BATCH_JOB_EXECUTION** table for any stray records corresponding to this failed job and delete them.
- Once done, run the batch again.

5. If there is a milestone present in the execution flow, then the execution would pause at that point, waiting for user input. A button will be made available to the user to click and proceed with the execution. The text on this button will proceed with **Run next stage**. The same is shown in the diagram below.

Figure 4-3 Invoke EOD - Run next stage



4.5 Invoking EOD by using APIs

Whenever you (customer) are using your own software to run EOD, following end points needs to be invoked.

This topic contains the following sub-topics:

- Upload workflow on the conductor
- Get the workflow details updated on conductor
- Initiate workflow
- Track status of initiated workflow
- Crossing a milestone task

4.5.1 Upload workflow on the conductor

Below endpoint should be invoked from the postman tool.

HTTP Method: POST

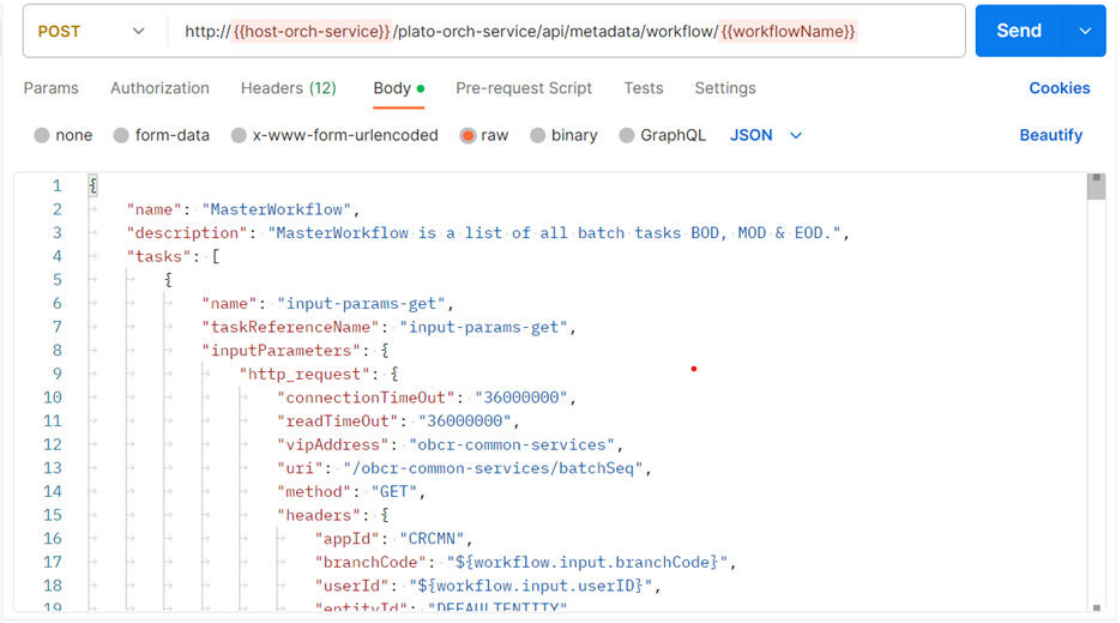
```
http://{{host-orch-service}}/plato-orch-service/api/metadata/workflow/
{{workflowName}}
```

Following headers to be considered:

Table 4-1 Headers

Parameter Name	Value	Mandatory	Description
Content-Type	application/json	Yes	-
appId	platoorch	Yes	The application ID of the host service
entityId	DEFAULTENTITY	Yes	EntityID of the Request.
multiEntityAdmin	N	Yes	-

Figure 4-4 JSON_Code workflow



4.5.2 Get the workflow details updated on conductor

Below endpoint should be invoked from the postman tool.

HTTP Method: GET

```
http://{{host-orch-service}}/plato-orch-service/api/metadata/workflow/
{{workflowName}}
```

Following headers to be considered:

Table 4-2 Headers

Parameter Name	Value	Mandatory	Description
Content-Type	application/json	Yes	-
appId	platoorch	Yes	The application ID of the host service.
entityId	DEFAULTENTITY	Yes	EntityID of the Request.

4.5.3 Initiate workflow

Below endpoint should be invoked from the postman tool.

HTTP Method: POST

```
http://{{host-orch-service}}/plato-orch-service/api/workflow/{{workflowName}}
```

Returns: This endpoint creates a new workflow and returns its id. This id can further be used to track the status of this workflow.

Following headers to be considered:

Table 4-3 Headers

Parameter Name	Value	Mandatory	Description
Content-Type	application/json	Yes	-
appId	platoorch	Yes	The application ID of the host service.
branchcode	DMO	Yes	Branch Code of the requesting user.
Userid	ADMINUSER1	Yes	User Id of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID of the Request.
multiEntityAdmin	N	Yes	-

Specify following parameters:

Table 4-4 Parameters

Parameter Name	Value	Mandatory	Description
branchcode	DMO	Yes	Branch Code of the requesting user.
Userid	ADMINUSER1	Yes	User Id of the requesting user.

4.5.4 Track status of initiated workflow

Below endpoint should be invoked from the postman tool.

HTTP Method: GET

```
http://{{host-orch-service}}/plato-orch-service/api/workflow/{{workflowId}}
```

Following headers to be considered:

Table 4-5 Headers

Parameter Name	Value	Mandatory	Description
Content-Type	application/json	Yes	-
appId	platoorch	Yes	The application ID of the host service.
entityId	DEFAULTENTITY	Yes	EntityID of the Request.

4.5.5 Crossing a milestone task

Below endpoint should be invoked from the postman tool.

HTTP Method: GET

```
https://{{host-orch-service}}/cmc-branch-services/brancheod/task/update/
{{branchCode}}/{{workflowId}}/{{taskId}}
```

Following headers to be considered:

Table 4-6 Headers

Parameter Name	Value	Mandatory	Description
Content-Type	application/json	Yes	-
appId	CMNCORE	Yes	The application ID of the host service.
branchcode	DMO	Yes	Branch Code of the requesting user.
userid	ADMINUSER1	Yes	User Id of the user.
entityId	DEFAULTENTITY	Yes	EntityID of the Request.

Specify following parameters:

Table 4-7 Parameters

Parameter Name	Value	Mandatory	Description
branchcode	DMO	Yes	Branch Code of the requesting user.
Userid	ADMINUSER1	Yes	User Id of the requesting user.

Glossary

Index