Oracle® Banking Collections
Batch Execution Guide

Release 14.7.4.0.0

G11171-01

June 2024

ORACLE®

Oracle Banking Collections Batch Execution Guide, Release 14.7.4.0.0

G11171-01

Contents

1 Introduction

2 Batch Execution Sequence Diagram

3 Batch Execution Details

4 Batch Configuration Details

ORACLE

Index

Preface

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

This section consists of following topics:

- Audience
- Documentation Accessibility
- Diversity and Inclusion
- Related Resources
- Conventions

Audience

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

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 - On-Premise
- Oracle Banking Collections Security Guide
- Oracle Banking Collections Maintenance User Guide
- Oracle Banking Collections Transactions User Guide

Conventions

6

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.

ORACLE

1

Introduction

Oracle Banking Collection 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 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

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

Figure 2-1   Batch_Sequence
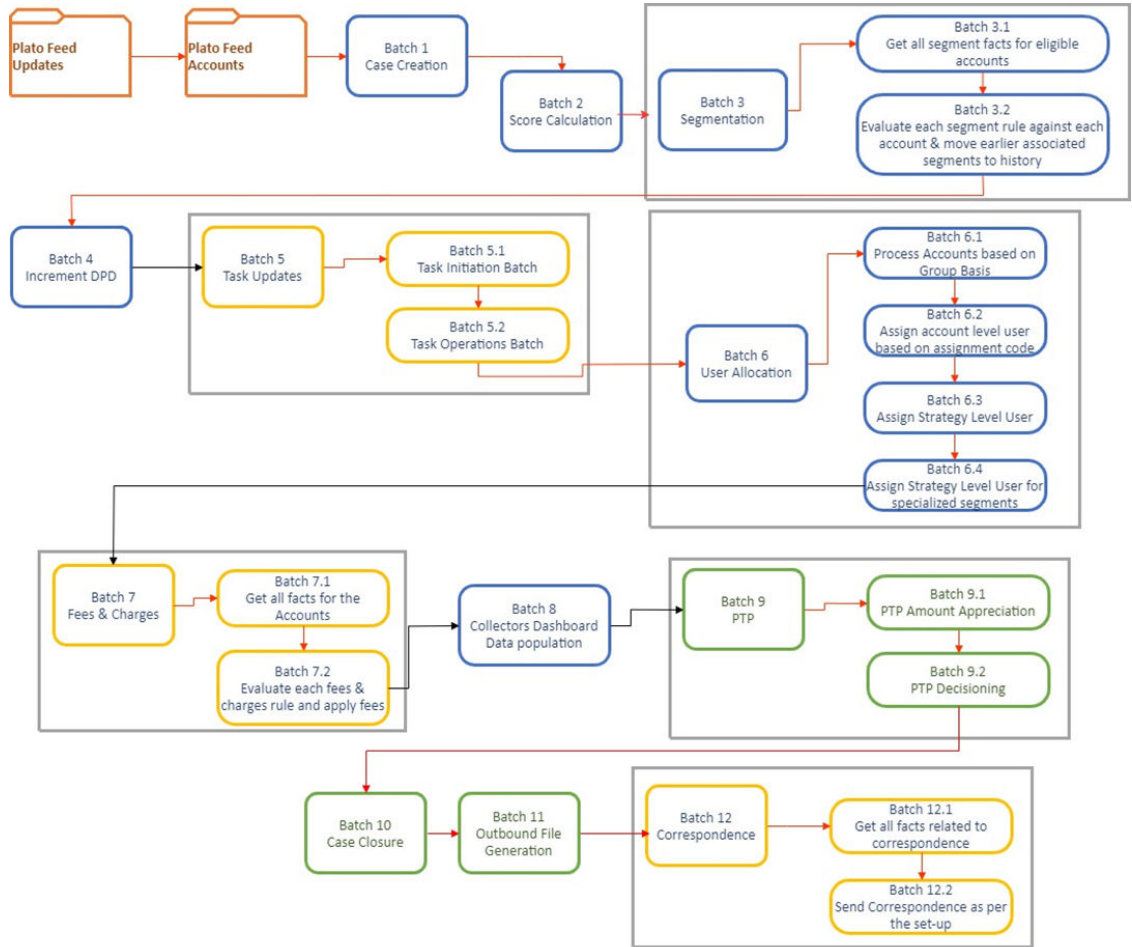
3

Batch Execution Details

This section explains the various batch processes in the below sections:

- Case Creation Batch
- Score Calculation Batch
- Segmentation Batch
- Increment DPD Batch
- Tasks Initiation Batch
- User Allocation Batch
- Fees and Charges Batch
- Dashboard Data Population Batch
- Promise Tracking Batch
- Case Closure Batch
- Outbound File Generation Batch
- Correspondence Batch

3.1 Case Creation Batch

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 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 3-1   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	COLLADMIN1	Yes	User Id of the requesting user.
appId	CRSTRTGY	Yes	The application ID of the host service.
branchCode	BCR	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	Entity ID of the request.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-2   Parameters

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

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 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 3-3   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRENTITY	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-4   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'

3.3 Segmentation Batch

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 3-5   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRSEG	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-6   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>:// obedm-segmentation-batch-services/jobLauncher/runBatch/

Table 3-7   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRSEG	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-8   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'

3.4 Increment DPD Batch

Oracle Banking Collections 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.

ORACLE

If the product processor sends blank value in this field, then in Oracle Banking Collections, 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 3-9   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRENTITY	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-10   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.
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'

3.5 Tasks Initiation Batch

- **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

    Table 3-11   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRTASK	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.
Accept	Application/json	Yes	Content Type of the request.

    Specify following parameters while executing this batch:

    Table 3-12   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.

- **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

Table 3-13   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRSEG	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-14   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'

3.6 User Allocation Batch

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 3-15   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRUSERMGMT	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-16   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.

Table 3-16   (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:**
  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 3-17   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRUSERMGMT	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-18   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.

Table 3-18 (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 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

Table 3-19 Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRUSERMGMT	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-20 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.

Table 3-20   (Cont.) Parameters

Parameter Name	Value	Mandatory	Description
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

Table 3-21   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRUSERMGMT	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-22   Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	The branch code of requesting branch.

Table 3-22   (Cont.) Parameters

Parameter Name	Value	Mandatory	Description
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'

3.7 Fees and Charges Batch

Oracle Banking Collections 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 3-23   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
AppId	CRSEG	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.

Table 3-23   (Cont.) Mandatory Headers

Parameter Name	Value	Mandatory	Description
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-24   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 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 3-25   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRSEG	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-26   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'

3.8 Dashboard Data Population Batch

Oracle Banking Collections 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 3-27   Mandatory Headers

Parameter Name	Value	Mandatory	Description
UserId	ADMINUSER1	Yes	User Id of the requesting user.
AppId	CRSEG	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-28   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 CR_DASHBOARD 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'

3.9 Promise Tracking Batch

Oracle Banking Collections 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

Table 3-29   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRSEG	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.

Table 3-29   (Cont.) Mandatory Headers

Parameter Name	Value	Mandatory	Description
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-30   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

Table 3-31   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRSEG	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-32   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'

3.10 Case Closure Batch

Oracle Banking Collections 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 3-33   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
AppId	CRSEG	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-34   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.
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'

3.11 Outbound File Generation Batch

Oracle Banking Collections 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 3-35   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRSEG	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-36   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'

3.12 Correspondence Batch

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 3-37   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRSEG	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-38   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

    This batch evaluates all active correspondence template rules against each account and perform communication via letter or email or sms as per 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 3-39   Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the requesting user.
appId	CRSEG	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.
Accept	Application/json	Yes	Content Type of the request.

Specify following parameters while executing this batch:

Table 3-40   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.
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'

4
Batch Configuration Details

This section consists of the following:

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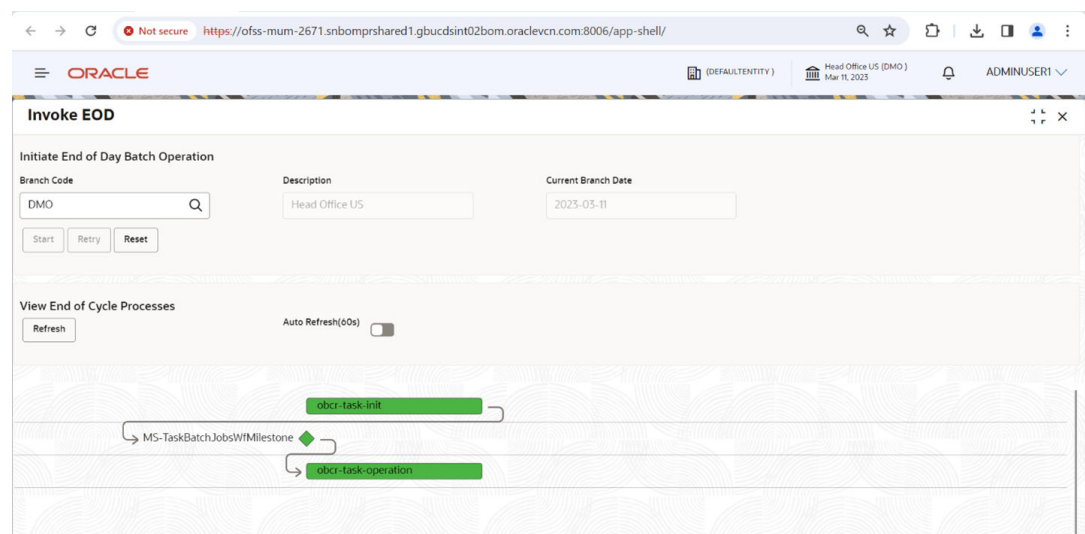
4. If there is any failure in the batch, below is how it would appear. Here, **obcr-task-operation** batch has failed, hence it shows in red.

Figure 4-2   Invoke EOD - Processing
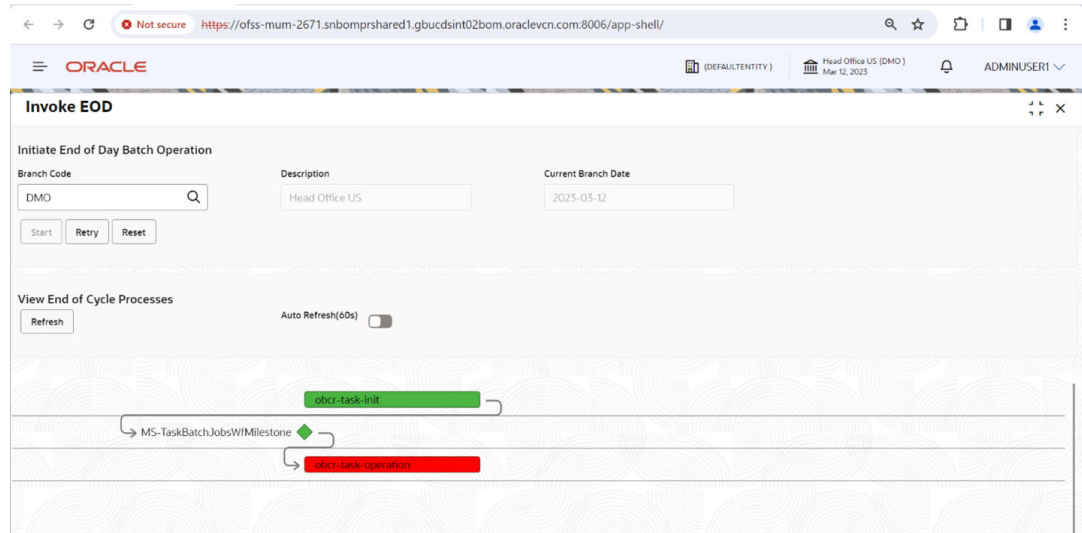

In this case, please visit the PLATO_BATCH schema, and execute the below query.

```
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
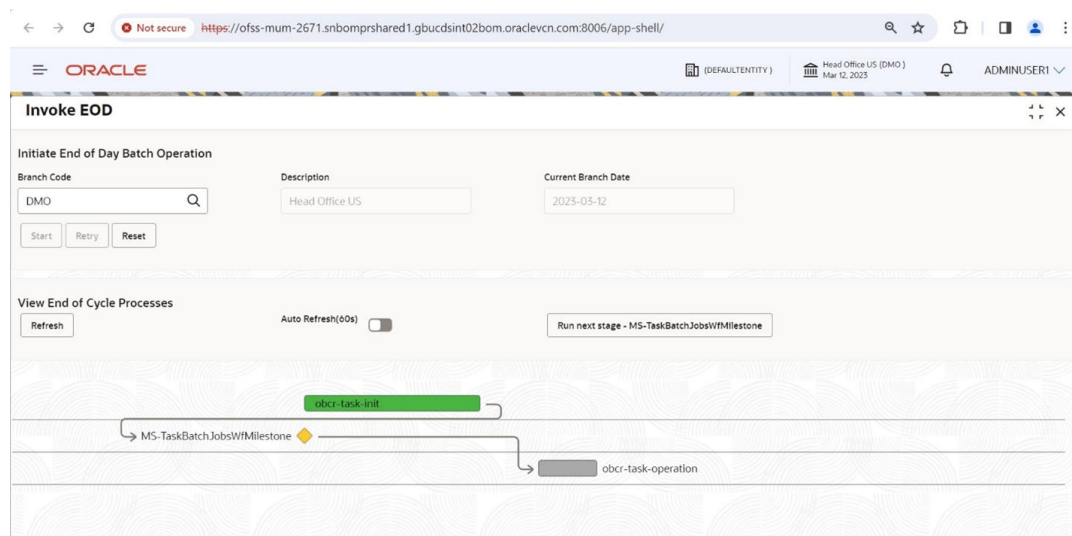
4.5 Invoking EOD by using APIs

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

- 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

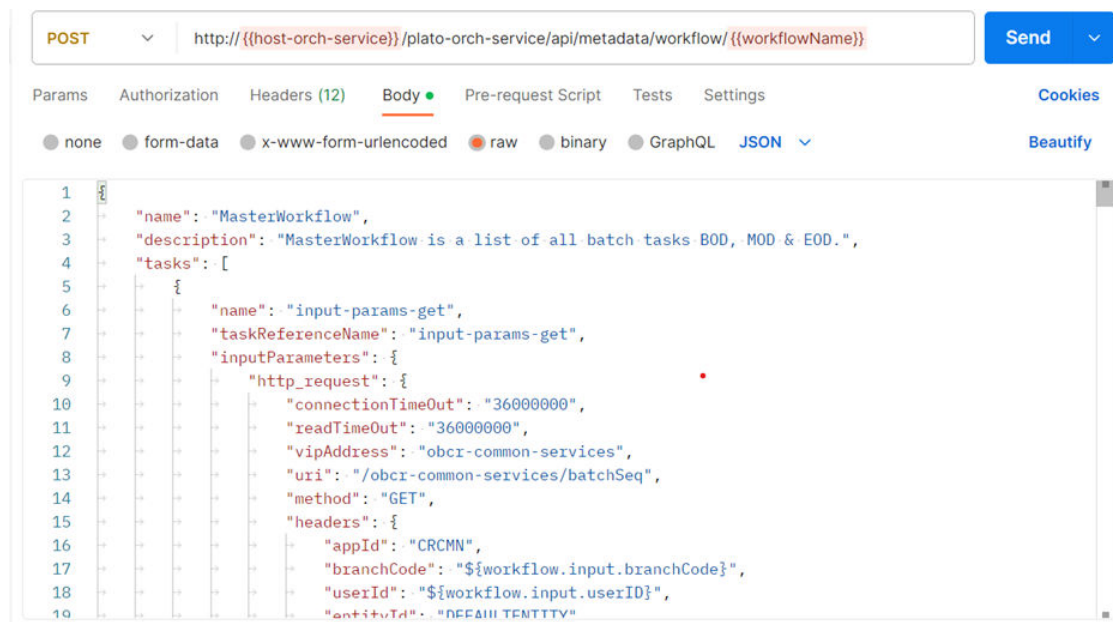Below endpoint should be invoked from the postman tool.

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