

# Oracle® Banking Collections and Recovery

## Batch Execution Guide



Release 14.7.3.0.0

F96492-01

February 2024

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Banking Collections and Recovery Batch Execution Guide, Release 14.7.3.0.0

F96492-01

Copyright © 2023, Oracle and/or its affiliates.

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

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

Contributors: (contributor), (contributor)

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

1	Introduction	
2	Batch Execution Sequence Diagram	
3	Batch Execution Details	
3.1	Case Creation Batch	3-1
3.2	Segmentation Batch	3-2
3.3	Increment DPD Batch	3-5
3.4	Tasks Initiation Batch	3-6
3.5	User Allocation Batch	3-8
3.6	Fees and Charges Batch	3-12
3.7	Dashboard Data Population Batch	3-14
3.8	Promise Tracking Batch	3-15
3.9	Case Closure Batch	3-17
3.10	Outbound File Generation Batch	3-19
3.11	Correspondence Batch	3-20
4	Batch Configuration Details	
4.1	Defining workflow using Netflix conductor	4-1
4.2	Pre-requisites to deploy conductor process	4-3
4.3	Branch EOD configuration through common core maintenance	4-3
4.4	Success and Failure analysis	4-3
4.5	Invoking EOD by using APIs	4-5
4.5.1	Upload workflow on the conductor	4-5
4.5.2	Get the workflow details updated on conductor	4-6
4.5.3	Initiate workflow	4-6
4.5.4	Track status of initiated workflow	4-7
4.5.5	Crossing a milestone task	4-7



# 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 and Recovery.

## 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 and Recovery License Guide - On-Premise](#)
- [Oracle Banking Collections and Recovery Security Guide](#)
- [Oracle Banking Collections and Recovery Maintenance User Guide](#)
- [Oracle Banking Collections and Recovery Transactions User Guide](#)

# Conventions

The following text conventions are used in this document:

---

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	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

Oracle Banking Collection and Recovery 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:

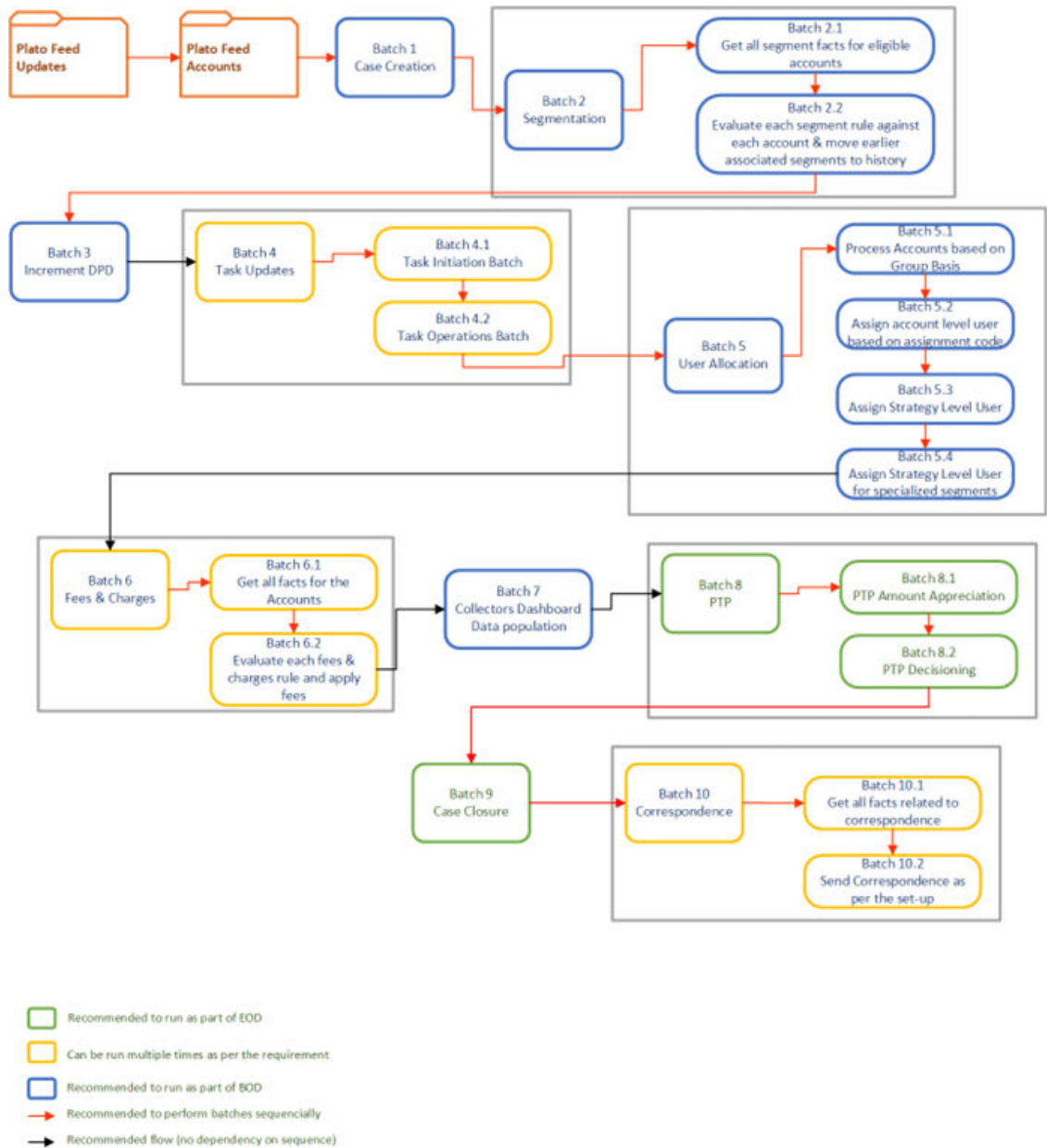
1. Oracle Banking Collections receives delinquent accounts from different host systems.
2. Cases are created for their primary party and accounts are linked to the case.
3. Accounts are then classified under different segments and various strategies are applied to recover the overdue amount.
4. When 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 and Recovery.

Figure 2-1 Batch\_Sequence





# 3

## Batch Execution Details

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

- [Case Creation 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 through 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.
appld	CRSTRGY	Yes	The application ID of the host service.
branchCode	BCR	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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.
appld	CRSTRGY	Yes	The application ID of the host service.
contextRoot	obcr-strategy-services	Yes	Context path of the service.
timestamp	YYYYMMDDHHMMSS. SSSS	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 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 3-3 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRSEG	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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.
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 3-5 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userid	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRSEG	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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.
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'

## 3.3 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.

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 3-7 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRENTITY	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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.
excludePPC	TPH	No	If multiple product processor needs to be skipped, then we need to pass their codes as comma separated values.

**Table 3-8 (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'

## 3.4 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 3-9 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRTASK	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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.
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 3-11 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRSEG	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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.

**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'

## 3.5 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 3-13 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userid	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRUSERMGMT	Yes	The application ID of the host service.



**Table 3-13 (Cont.) Mandatory Headers**

Parameter Name	Value	Mandatory	Description
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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	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 3-15 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRUSERMGMT	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID of the Request.

**Table 3-15 (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 3-16 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.
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 3-17 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the Requesting user.

**Table 3-17 (Cont.) Mandatory Headers**

Parameter Name	Value	Mandatory	Description
appld	CRUSERMGMT	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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.
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 3-19 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRUSERMGMT	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.

**Table 3-19 (Cont.) Mandatory Headers**

Parameter Name	Value	Mandatory	Description
entityId	DEFAULTENTITY	Yes	EntityID 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.
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'

## 3.6 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 3-21 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userid	ADMINUSER1	Yes	User Id of the Requesting user.
Appld	CRSEG	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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	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 3-23 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userid	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRSEG	Yes	The application ID of the host service.

**Table 3-23 (Cont.) Mandatory Headers**

Parameter Name	Value	Mandatory	Description
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID of the Request.
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.

**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'

## 3.7 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 3-25 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
UserId	ADMINUSER1	Yes	User Id of the Requesting user.

**Table 3-25 (Cont.) Mandatory Headers**

Parameter Name	Value	Mandatory	Description
Appld	CRSEG	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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.
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'

## 3.8 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 3-27 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userid	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRSEG	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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.

- **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 3-29 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userid	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRSEG	Yes	The application ID of the host service.



**Table 3-29 (Cont.) Mandatory Headers**

Parameter Name	Value	Mandatory	Description
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID of the Request.
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.
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'

## 3.9 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 3-31 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userid	ADMINUSER1	Yes	User Id of the Requesting user.
Appld	CRSEG	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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 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'

## 3.10 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 3-33 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRSEG	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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.
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'

## 3.11 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 3-35 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRSEG	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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.

- **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 3-37 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
userid	ADMINUSER1	Yes	User Id of the Requesting user.
appld	CRSEG	Yes	The application ID of the host service.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	EntityID 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.
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:

- [Defining workflow using Netflix conductor](#)
- [Pre-requisites to deploy conductor process](#)
- [Branch EOD configuration through common core maintenance](#)
- [Success and Failure analysis](#)
- [Invoking EOD by using APIs](#)

### 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.

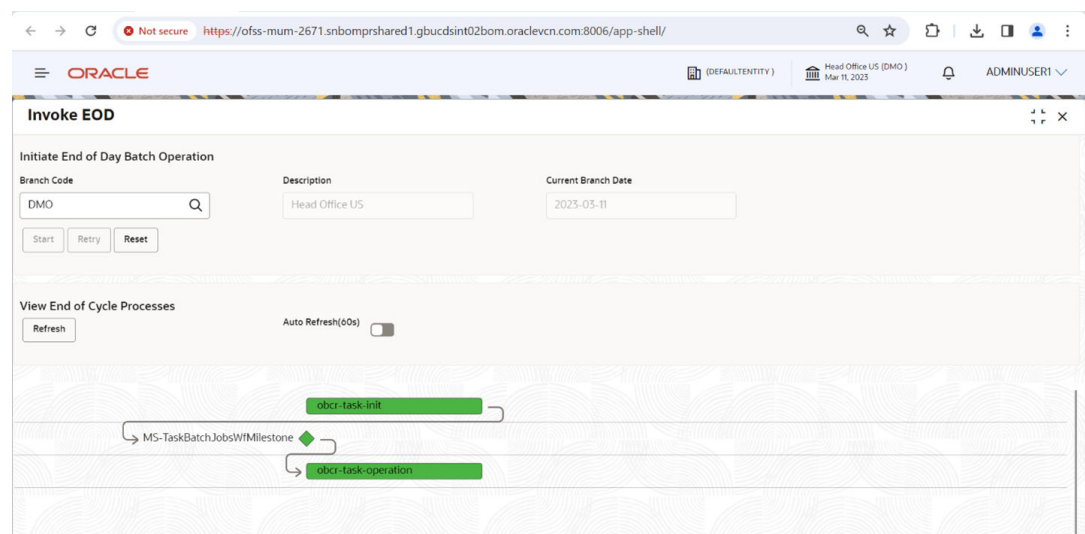
## 4.3 Branch EOD configuration through common core maintenance

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

## 4.4 Success and Failure analysis

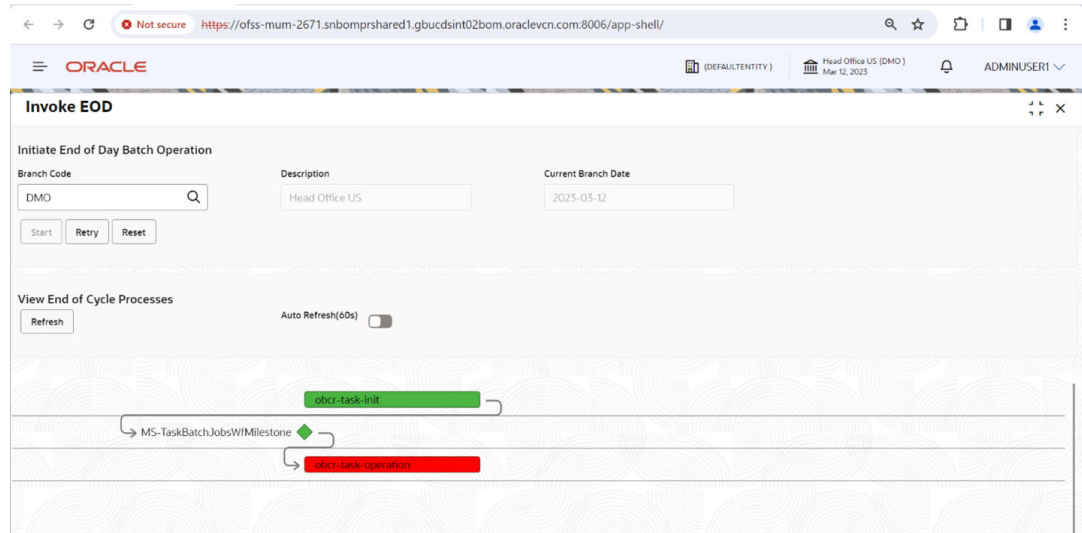
1. Navigate to the branch EOD screen. Core Maintenance → Branch EOD → Invoke EOD.
2. Enter the branch code.
3. If the batches have successfully run for that day, below is how it would appear. Here, **obcr-task-init** and **obcr-task-operation** are two batches which have completed successfully, hence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



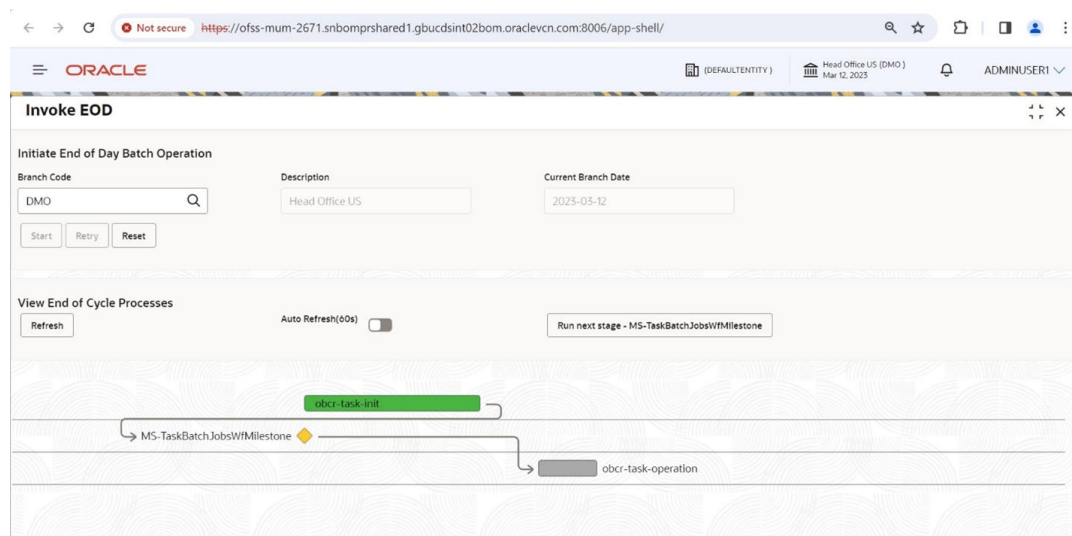
In this case, please visit the PLATO\_BATCH schema, and execute the below query.

```
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

If you are using their 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

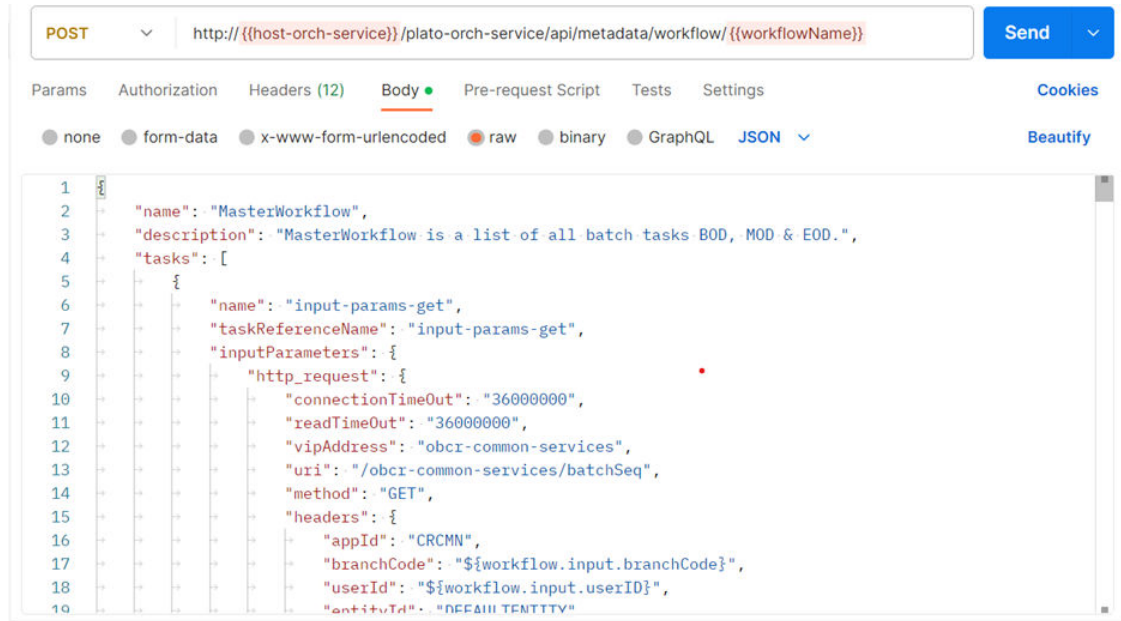
`http://{{host-orch-service}}/plato-orch-service/api/metadata/workflow/  
{{workflowName}}`

Following headers to be considered:

Table 4-1 Headers

Parameter Name	Value	Mandatory	Description
Content-Type	application/json	Yes	-
appld	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	-
appld	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	-
appld	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	-
appld	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	-
appld	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