

Oracle® Banking Collections

Batch Execution Guide



Release 14.9.0.0.0

G61048-01

August 2026

ORACLE®

Copyright © 2022, 2026, Oracle and/or its affiliates.

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	Entity Creation	2
3.1.1	Inbound for Mandatory Entities Creation	2
3.1.2	Inbound for Non-Mandatory Entities Creation	3
3.1.3	Update Entity Batch	4
3.1.4	Data Transfer Job	6
3.2	Batch Execution	26
3.2.1	Case Creation Batch	27
3.2.2	Score Calculation Batch	28
3.2.3	Segmentation Batch	29
3.2.4	Increment DPD Batch	30
3.2.5	Tasks Initiation Batch	31
3.2.6	User Allocation Batch	33
3.2.7	Vendor Allocation Batch	37
3.2.8	Fees and Charges Batch	38
3.2.9	Dashboard Data Population Batch	40
3.2.10	Dialer Extract Batch	41
3.2.11	Customer Correspondence Batch	44
3.2.12	Internal Correspondence Batch	45
3.2.13	Promise Tracking Batch	47
3.2.14	Settlement Status Update Batch	48
3.2.15	Case Closure Batch	49
3.2.16	Outbound File Generation Batch	50
3.2.17	Cured Account Data Transfer Job	51
3.2.18	Task Prioritization Batch	52
3.2.19	Report Data Population Job	53

4 Batch Configuration Details

4.1	Define workflow using Netflix conductor	1
4.2	Pre-requisites to deploy conductor process	3
4.3	Branch EOD configuration through common core maintenance	3
4.4	Success and Failure analysis	3
4.5	Invoking EOD by using APIs	5
4.5.1	Upload workflow on the conductor	6
4.5.2	Get the workflow details updated on conductor	6
4.5.3	Initiate workflow	7
4.5.4	Track status of initiated workflow	8
4.5.5	Crossing a Milestone Task	8

Index

Preface

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

This section consists of following topics:

- [Purpose](#)
- [Before You Begin](#)
- [Audience](#)
- [Documentation Accessibility](#)
- [Critical Patches](#)
- [Diversity and Inclusion](#)
- [Related Resources](#)
- [Conventions](#)
- [Screenshot Disclaimer](#)
- [Acronyms and Abbreviations](#)
- [Basic Actions](#)
- [Symbols and Icons](#)

Purpose

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

Before You Begin

Kindly refer to our getting started user guide for common elements, including Symbols and Icons, Conventions Definitions, and so forth.

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.

Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at [Critical Patches, Security Alerts and Bulletins](#). All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by [Oracle Software Security Assurance](#).

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

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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which user supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that user enter.

Screenshot Disclaimer

Personal information used in the interface or documents is dummy and does not exist in the real world. It is only for reference purposes.

Acronyms and Abbreviations

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

Table Acronyms and Abbreviations

Abbreviation	Description
DDA	Demand Deposit Accounts
ECA	External Credit Approval
EOD	End of Day
IBAN	International Bank Account Number

Basic Actions

The basic actions performed in the screens are as follows:

Table Basic Actions

Actions	Description
New	Click New to add a new record. The system displays a new record to specify the required data. The fields marked with asterisk are mandatory. This button is displayed only for the records that are already created.
Save	Click Save to save the details entered or selected in the screen.
Unlock	Click Unlock to update the details of an existing record. The system displays an existing record in editable mode. This button is displayed only for the records that are already created.
Authorize	Click Authorize to authorize the record created. A maker of the screen is not allowed to authorize the same. Only a checker can authorize a record. This button is displayed only for the already created records. For more information on the process, refer Authorization Process.
Approve	Click Approve to approve the initiated record. This button is displayed once the user click Authorize.
Audit	Click Audit to view the maker details, checker details of the particular record. This button is displayed only for the records that are already created.
Close	Click Close to close a record. This action is available only when a record is created.
Confirm	Click Confirm to confirm the action performed.
Cancel	Click Cancel to cancel the action performed.
Compare	Click Compare to view the comparison through the field values of old record and the current record. This button is displayed in the widget once the user click Authorize.
View	Click View to view the details in a particular modification stage. This button is displayed in the widget once the user click Authorize.

Table (Cont.) Basic Actions

Actions	Description
View Difference only	Click View Difference only to view a comparison through the field element values of old record and the current record, which has undergone changes. This button is displayed once the user click Compare.
Expand All	Click Expand All to expand and view all the details in the sections. This button is displayed once the user click Compare.
Collapse All	Click Collapse All to hide the details in the sections. This button is displayed once the user click Compare.
OK	Click OK to confirm the details in the screen.

Symbols and Icons

This guide has the following list of symbols and icons.

Table Symbols and Icons - Common

Symbol/Icon	Function
	Minimize
	Maximize
	Close
	Perform Search
	Open a list
	Add a new record
	Navigate to the first record
	Navigate to the last record

Table (Cont.) Symbols and Icons - Common

Symbol/Icon	Function
	Navigate to the previous record
	Navigate to the next record
	Grid view
	List view
	Refresh
	Click this icon to add a new row.
	Click this icon to delete a row, which is already added.
	Calendar
	Alerts

Table Symbols and Icons – Audit Details

Symbol/Icon	Function
	A user
	Date and time
	Unauthorized or Closed status

Table (Cont.) Symbols and Icons – Audit Details

Symbol/Icon	Function
	Authorized or Open status
	Rejected status

Table Symbols and Icons - Widget

Symbol/Icon	Function
	Open status
	Unauthorized status
	Closed status
	Authorized status
	Rejected status
	Modification Number

1

Introduction

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

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.

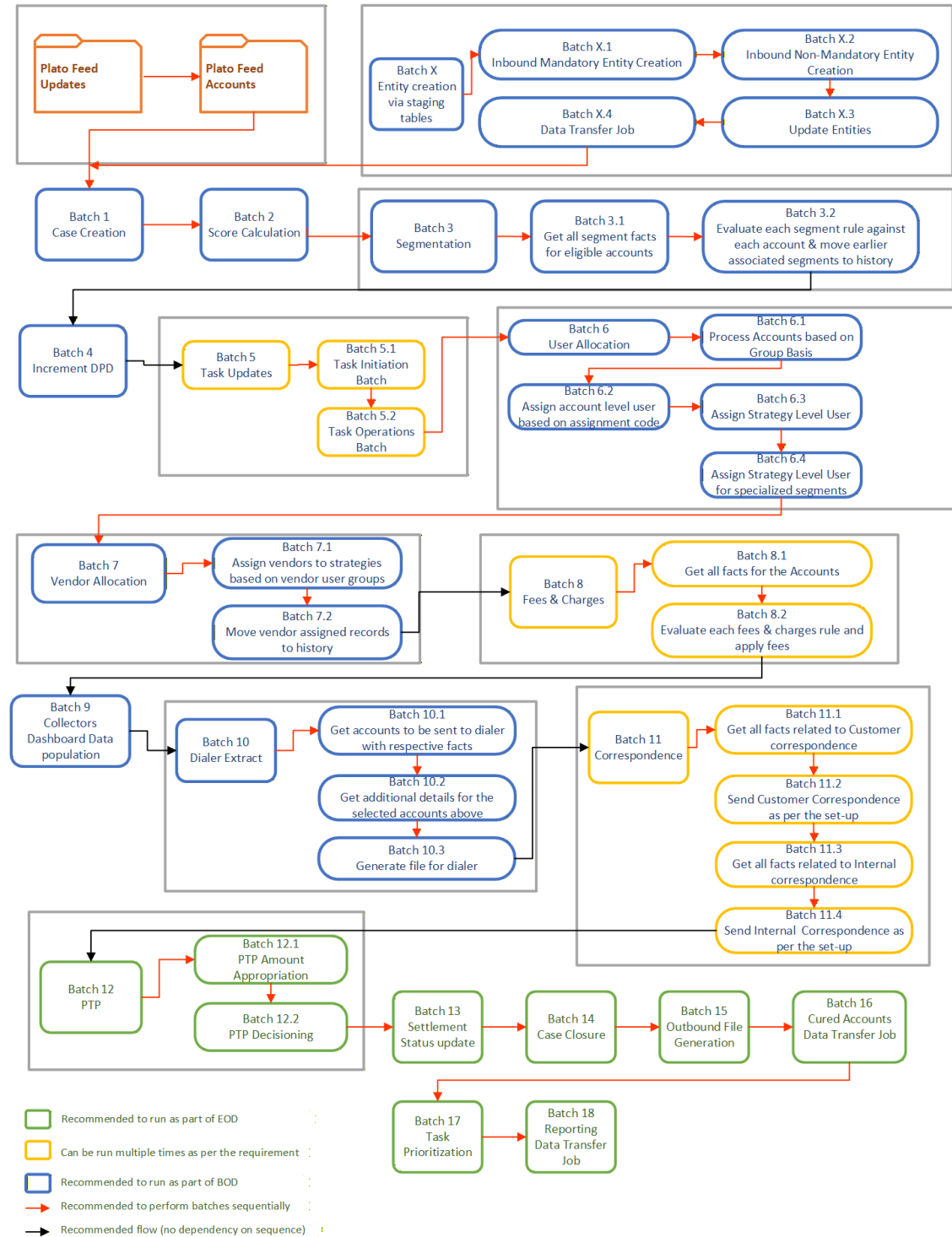
2

Batch Execution Sequence Diagram

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

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

Figure 2-1 Batch_Sequence



3

Batch Execution Details

This section describes the various batch processes used to onboard delinquent accounts, as well as the subsequent OBC batches executed after the account is onboarded.

Inbound data processing:

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

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

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

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

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

Table 3-1 Mandatory Headers

Parameter Name	Value	Mandatory	Description
userId	ADMINUSER1	Yes	User ID of the requesting user.
branchCode	DMO	Yes	Branch Code of the requesting user.
entityId	DEFAULTENTITY	Yes	Entity ID of the request.
Accept	application/json	Yes	Defines the expected response format.
Content-Type	application/json	Yes	Defines the request body format.
multiEntityAdmin	N	Yes	Indicates if multi-entity admin mode is enabled (Y/N).
Connection	keep-alive	Yes	Keeps the connection open for better.
Accept-Language	en-US	Yes	Requests responses Language.

Details of batches:

- [Entity Creation](#)
This topic provides information on the creation of mandatory and non-mandatory entities during the onboarding of a delinquent account, as well as the processing of subsequent updates received.
- [Batch Execution](#)
This section explains the various batch processes of Oracle Banking Collections.

3.1 Entity Creation

This topic provides information on the creation of mandatory and non-mandatory entities during the onboarding of a delinquent account, as well as the processing of subsequent updates received.

This topic contains the following sub-topics:

- [Inbound for Mandatory Entities Creation](#)
This topic describes the information about Mandatory Inbound for Entity Creation.
- [Inbound for Non-Mandatory Entities Creation](#)
This topic describes the information about Inbound for Non-mandatory Entity Creation.
- [Update Entity Batch](#)
This topic describes the information about Update Entity Batch.
- [Data Transfer Job](#)
This topic describes the information about Data Transfer Job.

3.1.1 Inbound for Mandatory Entities Creation

This topic describes the information about Mandatory Inbound for Entity Creation.

This batch processes data of mandatory entities from staging tables to create a new delinquent account with the related party details.

This batch processes data for the following entities:

- Account Details
- Account Party Details
- Party Details
- Party Name Details

JobName: obcrInboundEntityCreationBatch

Dependent Batch/Job: NotApplicable

URI: <protocol>:obcr-entity-data-services/jobLauncher/run/

Mandatory Headers:

Table 3-2 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appld	CRENTITYDATA	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-3 Parameters

Parameter Name	Value	Mandatory	Description
microServiceName	obcr-entity-data-services	Yes	Name of micro service.
appld	CRENTITYDATA	Yes	The application ID of the host service.

Table 3-3 (Cont.) Parameters

Parameter Name	Value	Mandatory	Description
contextRoot	obcr-entity-data-services	Yes	Context root value.
overrideDate	2021-03-07	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.
productProcessrCode	TPH	No	Value of Product Processor is required only if user wants to run this batch for a particular product processor.
timestamp	20260210190234.34300	No	Batch running timestamp in YYYYMMDDHHMMSS.SSSSS

Success/Failure Details:

- User can verify the record level status in the respective staging tables.
- For a successful processing of a record, Process Status will be updated as 'S' and for failure of records the same will be updated as 'F'.

3.1.2 Inbound for Non-Mandatory Entities Creation

This topic describes the information about Inbound for Non-mandatory Entity Creation.

This batch processes data of non-mandatory entities from staging tables (i.e Bill Details, Payments, Repayment Schedule, Arrear Details, Collateral Details etc) to update/insert data related to delinquent accounts already onboarded in Collections.

This batch processes data for the following entities:

- Account Arrears
- Bill Details
- Repayment Schedule
- Payment Details
- Party Address
- Party Contact
- Party Employment Details
- Party Identity Details
- Collateral Details
- Collateral Charge Details
- Collateral Linkage Details
- Collateral Ownership Details

JobName: obcrInbNonManEntityCreationBatch

Dependent Batch/Job: Not Applicable

Multi-threaded: Yes

URI: <protocol>:obcr-entity-data-services/jobLauncher/run/

Mandatory Headers:**Table 3-4 Mandatory Headers**

Parameter Name	Value	Mandatory	Description
appld	CRENTITYDATA	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-5 Parameters

Parameter Name	Value	Mandatory	Description
microServiceName	obcr-entity-data-services	Yes	Name of micro service.
appld	CRENTITYDATA	Yes	The application ID of the host service.
contextRoot	obcr-entity-data-services	Yes	Context root value.
overrideDate	2021-03-07	No	It must be in yyyy-mm-ddformat. It is required to run the batch for a specific day.
productProcessrCode	TPH	No	Value of Product Processor is required only if user wants to run this batch for a particular product processor.
timestamp	20260210190234.34300	No	Batch running timestamp in YYYYMMDDHHMMSS.SSSSS.

Success/Failure Details:

- User can verify the record level status in the respective staging tables.
- For a successful processing of a record, Process Status will be updated as 'S' and for failure of records the same will be updated as 'F'.

3.1.3 Update Entity Batch

This topic describes the information about Update Entity Batch.

Updates to existing entities are received from the host system in the staging area. This batch processes data from all relevant staging tables (e.g., Bill Details, Payments, Repayment Schedule, Arrear Details, Collateral Details, etc.) to insert or update information for delinquent accounts already onboarded in Collections.

This batch processes data for the following entities:

- Account Details
- Party Details
- Party Name Details
- Account Arrears
- Bill Details
- Repayment Schedule
- Payment Details
- Party Address

- Party Contact
- Party Employment Details
- Party Identity Details
- Collateral Details
- Collateral Charge Details
- Collateral Linkage Details
- Collateral Ownership Details

Job Name: obcrInboundEntityUpdateBatch

Dependent Batch/Job: Not Applicable

Multi-threaded: Yes

URI: <protocol>:obcr-entity-data-services/jobLauncher/run/

Mandatory Headers:

Table 3-6 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appld	CRENTITYDATA	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-7 Parameters

Parameter Name	Value	Mandatory	Description
microServiceName	obcr-entity-data-services	Yes	Name of micro service.
appld	CRENTITYDATA	Yes	The application ID of the host service.
contextRoot	obcr-entity-data-services	Yes	Context root value.
overrideDate	2021-03-07	No	It must be in yyyy-mm-ddformat. It is required to run the batch for a specific day.
productProcessrCode	TPH	No	Value of Product Processor is required only if user wants to run this batch for a particular product processor.
Timestamp	20260210190234.34300	No	Batch running timestamp in YYYYMMDDHHMMSS.SSSSS
tableList	DM_ACCOUNT_S TG	No	Value of Staging Table is required only if user wants to run this batch for a particular entity.

Success/Failure Details:

- User can verify the record level status in the respective staging tables.
- For a successful processing of a record, Process Status will be updated as 'S' and for failure of records the same will be updated as 'F'.

3.1.4 Data Transfer Job

This topic describes the information about Data Transfer Job.

All records processed in the above batches, along with their status (success or failure, including failure reasons), will be transferred to a separate schema through this job. A similar job can then be configured to relay these details to the source system, allowing it to review failed records, make necessary corrections, and resend them for processing.

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

Mandatory Headers:

Table 3-8 Mandatory Headers

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

Source Schema: CR_ENTITY

Destination Schema: CR_DATAEXCHNG.

Destination Tables: As provided by the host system.

Specify following parameters while executing this batch:

Table 3-9 Data Transfer sample configuration:

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

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
Request Body	<pre> [{ "srcTable": "DM_ACCOUNT_STG", "destTable": "DM_ACCOUNT_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "ACCT_NBR", "value": "ACCT_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }] }, { "srcTable": "DM_ACCOUNT_ARREARS_STG", "destTable": "DM_ACCOUNT_ARREARS_STG_PROCESS_STATUS", "customConfigDatas": [</pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> { "key": "ACCT_NBR", "value": "ACCT_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "ARS_TYP_CD", "value": "ARS_TYP_CD", "configType": "COLUMN_MAPPING" }, { "key": "REFERENCE_VAL", "value": "REFERENCE_VAL", "configType": "COLUMN_MAPPING" } }], { "srcTable": "DM_ACCOUNT_BILL_DTLS_STG", </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> "destTable": "DM_ACCOUNT_BILL_DTLS_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "ACCT_NBR", "value": "ACCT_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "BILL_GENERATE_DT", "value": "BILL_GENERATE_DT", "configType": "COLUMN_MAPPING" }] }, { "srcTable": "DM_ACCOUNT_PARTY_STG", "destTable": "DM_ACCOUNT_PARTY_STG_PROCESS_STATUS", </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> "customConfigDatas": [{ "key": "ACCT_NBR", "value": "ACCT_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "CUST_NBR", "value": "CUST_NBR", "configType": "COLUMN_MAPPING" }] }, { "srcTable": "DM_ACCOUNT_REPAY_SCH_STG", "destTable": "DM_ACCOUNT_REPAY_SCH_STG_PROCESS_STATUS", "customConfigDatas": [{ </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> "key": "ACCT_NBR", "value": "ACCT_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "INSTALLMENT_DT", "value": "INSTALLMENT_DT", "configType": "COLUMN_MAPPING" }] }, { "srcTable": "DM_PAYMENT_STG", "destTable": "DM_PAYMENT_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "ACCT_NBR", "value": "ACCT_NBR", </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "XREF_NO", "value": "XREF_NO", "configType": "COLUMN_MAPPING" } } }, { "srcTable": "DM_COLLATERAL_CHARGE_STG", "destTable": "DM_COLLATERAL_CHARGE_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "COLLATERAL_CD", "value": "COLLATERAL_CD", "configType": "COLUMN_MAPPING" }], </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "CHRG_CD", "value": "CHRG_CD", "configType": "COLUMN_MAPPING" }] }, { "srcTable": "DM_COLLATERAL_LINKAGE_STG", "destTable": "DM_COLLATERAL_LINKAGE_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "ACCT_NBR", "value": "ACCT_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "COLLATERAL_CD", "value": "COLLATERAL_CD", "configType": "COLUMN_MAPPING" }] }, { "srcTable": "DM_COLLATERAL_OWNER_STG", "destTable": "DM_COLLATERAL_OWNER_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "CUST_NBR", "value": "CUST_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }] } </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "COLLATERAL_CD", "value": "COLLATERAL_CD", "configType": "COLUMN_MAPPING" }] }, { "srcTable": "DM_COLLATERAL_STG", "destTable": "DM_COLLATERAL_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }] } </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "COLLATERAL_CD", "value": "COLLATERAL_CD", "configType": "COLUMN_MAPPING" }] }, { "srcTable": "DM_PARTY_ADDRESS_STG", "destTable": "DM_PARTY_ADDRESS_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "CUST_NBR", "value": "CUST_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "ADDR_TYPE_CD", "value": "ADDR_TYPE_CD", "configType": "COLUMN_MAPPING" } }], { "srcTable": "DM_PARTY_CONTACT_STG", "destTable": "DM_PARTY_CONTACT_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "CUST_NBR", "value": "CUST_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }], </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "CONTACT_POINT_TYP_CD", "value": "CONTACT_POINT_TYP_CD", "configType": "COLUMN_MAPPING" }] }, { "srcTable": "DM_PARTY_EMPLOYMENT_STG", "destTable": "DM_PARTY_EMPLOYMENT_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "CUST_NBR", "value": "CUST_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "EMP_SEQ_ID", "value": "EMP_SEQ_ID", "configType": "COLUMN_MAPPING" }] }, { "srcTable": "DM_PARTY_IDENTITY_STG", "destTable": "DM_PARTY_IDENTITY_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "CUST_NBR", "value": "CUST_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }] } </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "ID_TYPE_CD", "value": "ID_TYPE_CD", "configType": "COLUMN_MAPPING" }] }, { "srcTable": "DM_PARTY_NAME_STG", "destTable": "DM_PARTY_NAME_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "CUST_NBR", "value": "CUST_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }] } </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }, { "key": "NAME_TYP_CD", "value": "NAME_TYP_CD", "configType": "COLUMN_MAPPING" }] }, { "srcTable": "DM_PARTY_STG", "destTable": "DM_PARTY_STG_PROCESS_STATUS", "customConfigDatas": [{ "key": "CUST_NBR", "value": "CUST_NBR", "configType": "COLUMN_MAPPING" }, { "key": "RCD_TYPE", "value": "RCD_TYPE", "configType": "COLUMN_MAPPING" }, { "key": "PRD_PROCESSR_CD", "value": "PRD_PROCESSR_CD", "configType": "COLUMN_MAPPING" }, { "key": "CRET_DTTM", "value": "CRET_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "LAST_UPDT_DTTM", "value": "LAST_UPDT_DTTM", "configType": "COLUMN_MAPPING" }, { "key": "PROCESS_STATUS", "value": "PROCESS_STATUS", </pre>

Table 3-9 (Cont.) Data Transfer sample configuration:

Endpoint	Configurations
	<pre> "configType": "COLUMN_MAPPING" }, { "key": "ERROR_MSG", "value": "ERROR_MSG", "configType": "COLUMN_MAPPING" }] }]</pre>

Trigger endpoint

Table 3-10 Trigger endpoint

Endpoint	Configurations
Host location	https://host:port/obcr-data-exchange-services/transfers/CR_DATAEXCHNG/CR_ENTITY_PROCESS_STATUS?operation=T&branchCode=BCR&runDate=24122024
Header	<pre> appId: CRDATAEXCHNG entityId:DEFAULTENTITY Content-Type:application/json branchCode:BCR userId:AUTOUSER </pre>

Table 3-10 (Cont.) Trigger endpoint

Endpoint	Configurations
Request Body	<pre>[{ "tableName": "DM_ACCOUNT_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_ACCOUNT_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\\'F\\' OR PROCESS_STATUS = '\\'S\\'" }, { "tableName": "DM_ACCOUNT_ARREARS_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_ACCOUNT_ARREARS_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\\'F\\' OR PROCESS_STATUS = '\\'S\\'" }, { "tableName": "DM_ACCOUNT_BILL_DTLS_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_ACCOUNT_BILL_DTLS_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\\'F\\' OR PROCESS_STATUS = '\\'S\\'" }, { "tableName": "DM_ACCOUNT_PARTY_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_ACCOUNT_PARTY_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\\'F\\' OR PROCESS_STATUS = '\\'S\\'" }, { "tableName": "DM_ACCOUNT_REPAY_SCH_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_ACCOUNT_REPAY_SCH_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\\'F\\' OR PROCESS_STATUS = '\\'S\\'" }, { "tableName": "DM_COLLATERAL_CHARGE_STG_PROCESS_STATUS", "truncateFlag": "N",</pre>

Table 3-10 (Cont.) Trigger endpoint

Endpoint	Configurations
	<pre> "srcTableName": "DM_COLLATERAL_CHARGE_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\''F'\'' OR PROCESS_STATUS = '\''S'\''", }, { "tableName": "DM_COLLATERAL_LINKAGE_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_COLLATERAL_LINKAGE_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\''F'\'' OR PROCESS_STATUS = '\''S'\''", }, { "tableName": "DM_COLLATERAL_OWNER_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_COLLATERAL_OWNER_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\''F'\'' OR PROCESS_STATUS = '\''S'\''", }, { "tableName": "DM_COLLATERAL_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_COLLATERAL_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\''F'\'' OR PROCESS_STATUS = '\''S'\''", }, { "tableName": "DM_PARTY_ADDRESS_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_PARTY_ADDRESS_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\''F'\'' OR PROCESS_STATUS = '\''S'\''", }, { "tableName": "DM_PARTY_CONTACT_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_PARTY_CONTACT_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\''F'\'' OR PROCESS_STATUS = '\''S'\''", </pre>

Table 3-10 (Cont.) Trigger endpoint

Endpoint	Configurations
	<pre> }, { "tableName": "DM_PARTY_EMPLOYMENT_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_PARTY_EMPLOYMENT_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\''F'\'' OR PROCESS_STATUS = '\''S'\''", }, { "tableName": "DM_PARTY_IDENTITY_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_PARTY_IDENTITY_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\''F'\'' OR PROCESS_STATUS = '\''S'\''", }, { "tableName": "DM_PARTY_NAME_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_PARTY_NAME_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\''F'\'' OR PROCESS_STATUS = '\''S'\''", }, { "tableName": "DM_PARTY_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_PARTY_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\''F'\'' OR PROCESS_STATUS = '\''S'\''", }, { "tableName": "DM_PAYMENT_STG_PROCESS_STATUS", "truncateFlag": "N", "srcTableName": "DM_PAYMENT_STG", "srcTruncateFlag": "Y", "srcWhereClause": "PROCESS_STATUS = '\''F'\'' OR PROCESS_STATUS = '\''S'\''", }] </pre>

3.2 Batch Execution

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

This topic contains the following sub-topics:

- [Case Creation Batch](#)
This topic describes the information about Case Creation Batch.
- [Score Calculation Batch](#)
This topic describes the information about Score Calculation Batch.
- [Segmentation Batch](#)
This topic describes the information about Segmentation Batch.
- [Increment DPD Batch](#)
This topic describes the information about Increment DPD Batch.
- [Tasks Initiation Batch](#)
This topic describes the information about Tasks Initiation Batch.
- [User Allocation Batch](#)
This topic describes the information about User Allocation Batch.
- [Vendor Allocation Batch](#)
This topic describes the information about Vendor Allocation Batch .
- [Fees and Charges Batch](#)
This topic describes the information about Fees and Charges Batch.
- [Dashboard Data Population Batch](#)
This topic describes the information about Dashboard Data Population Batch.
- [Dialer Extract Batch](#)
This topic describes the information about Dialer Extract Batch.
- [Customer Correspondence Batch](#)
This topic describes the information about Customer Correspondence Batch.
- [Internal Correspondence Batch](#)
This topic describes the information about Internal Correspondence Batch.
- [Promise Tracking Batch](#)
This topic describes the information about Promise Tracking Batch.
- [Settlement Status Update Batch](#)
This topic describes the information about Settlement Status Update Batch.
- [Case Closure Batch](#)
This topic describes the information about Case Closure Batch.
- [Outbound File Generation Batch](#)
This topic describes the information about Outbound File Generation Batch.
- [Cured Account Data Transfer Job](#)
This topic describes the information about Cured Account Data Transfer Job.
- [Task Prioritization Batch](#)
This topic describes the information about Task Prioritization Batch.
- [Report Data Population Job](#)
This topic describes the information about Reporting Data Population Job.

3.2.1 Case Creation Batch

This topic describes the information about 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 3-11 Mandatory Headers

Parameter Name	Value	Mandatory	Description
appld	CRSTRGY	Yes	The application ID of the host service.

Specify following parameters while executing this batch:

Table 3-12 Parameters

Parameter Name	Value	Mandatory	Description
microServiceName	obcr-strategy-services	Yes	Service that consists of 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_STRATEGY schema for delinquent accounts.

3.2.2 Score Calculation Batch

This topic describes the information about 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 3-13 Mandatory Headers

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

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 of the project.
userId	ADMINUSER1	Yes	User ID of the user.

Success/Failure Details:

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

ERROR_CODE = 'Success' and PROCESS_RESULT = '1'

3.2.3 Segmentation Batch

This topic describes the information about 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-services/jobLauncher/runBatch/

Table 3-15 Mandatory Headers

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

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	Branch code of the project.
userId	ADMINUSER1	Yes	User ID of the user.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

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

Job Name:obcr-account-segmentation

Setup Prerequisites

Following is the prerequisite for the setup:

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

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

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-17 Mandatory Headers

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

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	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 user want to exclude accounts for few days from re-segmentation. Its default value is 1.

Success/Failure Details:

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

ERROR_CODE = 'Success' and PROCESS_RESULT = '1'

3.2.4 Increment DPD Batch

This topic describes the information about 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 3-19 Mandatory Headers

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

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	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'

3.2.5 Tasks Initiation Batch

This topic describes the information about 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

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

Table 3-21 Mandatory Headers

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

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

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

Job Name: obcr-task-operation

Dependent Batch/Job: obcr-task-init

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-23 Mandatory Headers

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

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

Success/Failure Details:

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

ERROR_CODE = 'Success' and PROCESS_RESULT = '1'

3.2.6 User Allocation Batch

This topic describes the information about 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 3-25 Mandatory Headers

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

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	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 3-27 Mandatory Headers

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

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	The branch code of requesting branch.
userId	ADMINUSER1	Yes	The user ID of the user triggering the batch.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

- Batch 3:**

This batch will assign strategy level users to all accounts where ignore group basis is maintained as 'No' for a segment and stamp assignment code maintained at the segment level for all accounts where ignore group basis is 'Yes'.

Functional Description:

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

Job Name: obcr-strategy-assignment-batch

Setup Prerequisites:

Following is the prerequisite for the setup:

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

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

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-29 Mandatory Headers

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

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	The branch code of requesting branch.
userId	ADMINUSER1	Yes	The user ID of the user triggering the batch.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

- Batch 4:**

Functional Description:

1. Batch will consider all the accounts from batch 3 on which account assignment code was stamped.
2. Based on the assignment method specified in the assignment code (i.e round robin method), users are assigned at strategy level.

Job Name: obcr-strategy-assign-batch

Setup Prerequisites:

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

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

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-31 Mandatory Headers

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

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	The branch code of requesting branch.
userId	ADMINUSER1	Yes	The user ID of the user triggering the batch.

Table 3-32 (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 the following tables in CR_USERMGMT schema.
 - DM_ACCOUNT_ALLOCATION_DRIVER
 - DM_ACCOUNT_ASSIGN_DRIVER
 - DM_STRATEGY_ASSIGNMENT_DRIVER
 - DM_STRATEGY_ASSIGN_DRIVER
- For a successful processing of a record, the below condition should be satisfied.
ERROR_CODE = 'Success' and PROCESS_RESULT = '1'

3.2.7 Vendor Allocation Batch

This topic describes the information about Vendor Allocation Batch .

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

- Batch 1:**
After the completion of user allocation batch, user will be assigned at strategy level.
This batch will consider only those records from the above set of records where user id belongs to any of the vendor user group and fetch all the relevant details for these accounts and persist the data.

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

Dependent Batch/Job: User Assignment

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-33 Mandatory Headers

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

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.

Table 3-34 (Cont.) Parameters

Parameter Name	Value	Mandatory	Description
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User ID of the user.

- Batch 2:**

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

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

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

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-35 Mandatory Headers

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

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.

Success/Failure Details:

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

- DM_VENDOR_ACCOUNT_ASSIGN_DRIVER
- DM_VENDOR_ACCOUNT_ASSIGN_HISTORY_DRIVER

3.2.8 Fees and Charges Batch

This topic describes the information about 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 3-37 Mandatory Headers

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

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:**

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 3-39 Mandatory Headers

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

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.

Success/Failure Details:

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

ERROR_CODE = 'Success' and PROCESS_RESULT = '1'

3.2.9 Dashboard Data Population Batch

This topic describes the information about 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 3-41 Mandatory Headers

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

Specify following parameters while executing this batch:

Table 3-42 Parameters

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

Success/Failure Details:

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

ERROR_CODE = 'Success' and PROCESS_RESULT = '1'

3.2.10 Dialer Extract Batch

This topic describes the information about Dialer Extract Batch.

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

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

Dependent Batch/Job: Not Applicable

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-43 Mandatory Headers

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

Specify following parameters while executing this batch:

Table 3-44 Parameters

Parameter Name	Value	Mandatory	Description
runDate	01102023	Yes	It must be in ddmmyyyy format.
branchCode	DMO	Yes	The branch code of the project.
userId	ADMINUSER1	Yes	The user ID of the user triggering the batch.
followUpDays	1	No	Number of Maximum Follow-up days (On Task) from Current Business Date to be considered.
campaignCodes	SEGMENT_EC	Yes	The campaign code will be the condition(rule) which needs to be evaluated to identify the accounts to be sent to dialer system. Multiple comma separated rules are supported.
overrideDate	2023-10-01	No	It must be in yyyy-mm-dd format. It is required to run the batch for a specific day.

- **Batch 2:**

Dialer Account Filter batch:

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

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

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

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-45 Mandatory Headers

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

Specify following parameters while executing this batch:

Table 3-46 Parameters

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

- Batch 3:**

Dialer File Generation Batch:

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

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

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

Multi-threaded: No

Multi-stream: No

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

Table 3-47 Mandatory Headers

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

Specify following parameters while executing this batch:

Table 3-48 Parameters

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

Success/Failure Details:

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

- DM_DIALER_ACCTS_DRIVER
- DM_DIALER_FILEGEN_DRIVER

For a successful processing of a record, the below condition should be satisfied.

ERROR_CODE = 'Success' and PROCESS_RESULT = '1'

3.2.11 Customer Correspondence Batch

This topic describes the information about Customer Correspondence Batch.

This 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 3-49 Mandatory Headers

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

Specify following parameters while executing this batch:

Table 3-50 Parameters

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

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

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

Setup Prerequisites

Following is the prerequisite for the setup:

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

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

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-51 Mandatory Headers

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

Specify following parameters while executing this batch:

Table 3-52 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 user want to perform certain mode of communication in particular.

Success/Failure Details:

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

3.2.12 Internal Correspondence Batch

This topic describes the information about Internal Correspondence Batch.

This batch currently runs in two sub-batches.

- Batch 1:**
This batch will get all correspondence related facts for all active accounts and users as per the maintenance.

Job Name: obcr-internal-communication-fact-batch

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-53 Mandatory Headers

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

Specify following parameters while executing this batch:

Table 3-54 Parameters

Parameter Name	Value	Mandatory	Description
runDate	23012026	Yes	It should be in DDMMYYYY.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User ID of the user.
overrideDate	2021-03-07	No	It must be in yyyy-mm- ddformat. It is required to run the batch for a specific day.

- Batch 2:**

Job Name: obcr-internal-communication-email-batch

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

DependentBatch/Job: obcr-internal-communication-fact-batch

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-55 Mandatory Headers

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

Specify following parameters while executing this batch:

Table 3-56 Parameters

Parameter Name	Value	Mandatory	Description
runDate	23012026	Yes	It should be in DDMMYYYY.
branchCode	DMO	Yes	Branch code of the project.
userId	ADMINUSER1	Yes	User ID of the user.
overrideDate	2021-03-07	No	It must be in yyyy-mm- ddformat. It is required to run the batch for a specific day.

Success/Failure Details:

- User can verify the record level status in the respective staging tables.

- For a successful processing of a record, Process Status will be updated as 'S' and for failure of records the same will be updated as 'F'.

3.2.13 Promise Tracking Batch

This topic describes the information about 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

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

Table 3-57 Mandatory Headers

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

Specify following parameters while executing this batch:

Table 3-58 Parameters

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

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

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

Setup Prerequisites

Following is the prerequisite for the setup:

The PTP Appropriation batch must have completed its run.

Dependent Batch/Job: Payment appropriation batch.

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-59 Mandatory Headers

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

Specify following parameters while executing this batch:

Table 3-60 Parameters

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

Success/Failure Details:

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

3.2.14 Settlement Status Update Batch

This topic describes the information about Settlement Status Update Batch.

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

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

Job Name: obcr-account-settlement-batch

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

Multi-threaded: Yes

Multi-stream: Yes

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

Table 3-61 Mandatory Headers

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

Specify following parameters while executing this batch:

Table 3-62 Parameters

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

Success/Failure Details:

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

ERROR_CODE = 'Success' and PROCESS_RESULT = '1'

3.2.15 Case Closure Batch

This topic describes the information about 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 3-63 Mandatory Headers

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

Specify following parameters while executing this batch:

Table 3-64 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'

3.2.16 Outbound File Generation Batch

This topic describes the information about 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 3-65 Mandatory Headers**

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

Specify following parameters while executing this batch:

Table 3-66 Parameters

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

Success/Failure Details:

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

ERROR_CODE = 'Success' and PROCESS_RESULT = '1'

3.2.17 Cured Account Data Transfer Job

This topic describes the information about Cured Account Data Transfer Job.

This task is intended to send cured account information to the source system. The source system can then examine the data and handle the cured accounts.

This job consists of two parts: **App configuration** and **Trigger endpoint**. The app configuration part involves mapping the source and destination schema tables along with their

column details. The trigger endpoint part is responsible for transferring data from the source driver table to the destination system.

Configuration for this job is similar to what is explained for data transfer job in the section [Data Transfer Job](#).

Mandatory Headers:

Table 3-67 Mandatory Headers for Configuration endpoint:

Parameter Name	Value	Mandatory	Description
appld	PLATOFTD	Yes	The application ID of the host service.

Table 3-68 Mandatory Headers for Trigger endpoint:

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

Source Schema: CR_ENTITY

Source table: DM_CURED_ACCOUNT

Destination Schema: As provided by the host system.

Destination Tables: As provided by the host system.

3.2.18 Task Prioritization Batch

This topic describes the information about Task Prioritization Batch.

This batch is responsible for reading the configuration settings and dynamically prioritizing tasks based on predefined criteria, such as the urgency of the account, overdue amounts, and customer risk factors.

Job Name: obcr-task-prioritization-batch

Setup Prerequisites - Following is the prerequisite for the setup:

Dependent Batch/Job: Not Applicable

Multi-threaded: Yes

Multi-stream: Yes

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

Mandatory Headers:

Table 3-69 Mandatory Headers for Configuration endpoint:

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

Specify following parameters while executing this batch:

Table 3-70 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_PRIORITIZATION_DRIVER. For a successful processing of a record, the below condition should be satisfied.

ERROR_CODE = 'Success' and PROCESS_RESULT = '1'

3.2.19 Report Data Population Job

This topic describes the information about Reporting Data Population Job.

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

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

Mandatory Headers:

Table 3-71 Mandatory Headers for Configuration endpoint:

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

Table 3-72 Mandatory Headers for Trigger endpoint:

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

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

Table 3-73 source and target schema

Source-SCHEMA NAME	Source-TABLE NAMES	Destination-SCHEMA NAME	Destination-TABLE NAMES
CR_ACTION	DM_ACCOUNT_COLLECTION_STATUS	CR_DATAEXCHNG	DM_RPT_ACCOUNT_COLLECTION_STATUS

Table 3-73 (Cont.) source and target schema

Source-SCHEMA NAME	Source-TABLE NAMES	Destination-SCHEMA NAME	Destination-TABLE NAMES
	DM_ACCOUNT_COLLECTION_STATUS_HISTORY	CR_DATAEXCHNG	DM_RPT_ACCOUNT_COLLECTION_STATUS_HISTORY
	DM_ACTION_RESULTS	CR_DATAEXCHNG	DM_RPT_ACTION_RESULTS
CR_ACTIVITY	DM_ACTIVITY	CR_DATAEXCHNG	DM_RPT_ACTIVITY
CR_DASHBOARD	DM_KPI	CR_DATAEXCHNG	DM_RPT_KPI
	DM_KPI_HIST	CR_DATAEXCHNG	DM_RPT_KPI_HIST
CR_ENTITY	DM_ACCOUNT	CR_DATAEXCHNG	DM_RPT_ACCOUNT
	DM_ACCOUNT_DEL_HISTORY	CR_DATAEXCHNG	DM_RPT_ACCOUNT_DEL_HISTORY
	DM_ACCOUNT_PARTY	CR_DATAEXCHNG	DM_RPT_ACCOUNT_PARTY
	DM_COLLATERAL	CR_DATAEXCHNG	DM_RPT_COLLATERAL
	DM_COLLATERAL_LINKAGE	CR_DATAEXCHNG	DM_RPT_COLLATERAL_LINKAGE
	DM_CURED_ACCOUNT	CR_DATAEXCHNG	DM_RPT_CURED_ACCOUNT
	DM_CURED_ACCOUNT_HIST	CR_DATAEXCHNG	DM_RPT_CURED_ACCOUNT_HIST
	DM_PARTY	CR_DATAEXCHNG	DM_RPT_PARTY
	DM_PARTY_NAME	CR_DATAEXCHNG	DM_RPT_PARTY_NAME
	DM_PAYMENT	CR_DATAEXCHNG	DM_RPT_PAYMENT
CR_FEECHRG	DM_FEESCHARGES_ACCT ASSO	CR_DATAEXCHNG	DM_RPT_FEESCHARGES_ACCT ASSO
	DM_FEESCHARGES_ACCT ASSO_HIST	CR_DATAEXCHNG	DM_RPT_FEESCHARGES_ACCT ASSO_HIST
CR_PTP	DM_PTP	CR_DATAEXCHNG	DM_RPT_PTP
	DM_PTP_HIST	CR_DATAEXCHNG	DM_RPT_PTP_HIST
CR_SEG	DM_SEGMENT_ACCOUNT ASSO	CR_DATAEXCHNG	DM_RPT_SEGMENT_ACCOUNT ASSO
	DM_SEGMENT_ACCOUNT ASSO_HIST	CR_DATAEXCHNG	DM_RPT_SEGMENT_ACCOUNT ASSO_HIST
	DM_SEGMENT_STGY_ACCOUNT ASSO	CR_DATAEXCHNG	DM_RPT_SEGMENT_STGY_ACCOUNT ASSO
CR_STRATEGY	DM_CASE	CR_DATAEXCHNG	DM_RPT_CASE

Table 3-73 (Cont.) source and target schema

Source-SCHEMA NAME	Source-TABLE NAMES	Destination-SCHEMA NAME	Destination-TABLE NAMES
	DM_CASE_ACCOUNT ASSO	CR_DATAEX CHNG	DM_RPT_CASE_ACCOUNT ASSO
	DM_CASE_ACCOUNT ASSO_HIST	CR_DATAEX CHNG	DM_RPT_CASE_ACCOUNT ASSO_HIST
	DM_CASE_HIST	CR_DATAEX CHNG	DM_RPT_CASE_HIST
CR_TASK	DM_TASK	CR_DATAEX CHNG	DM_RPT_TASK
	DM_TASK_ASSIGN_HIST	CR_DATAEX CHNG	DM_RPT_TASK_ASSIGN_HIST
	DM_TASK_HISTORY	CR_DATAEX CHNG	DM_RPT_TASK_HISTORY
CR_USERM GMT	DM_ACCOUNT_ASSIGN	CR_DATAEX CHNG	DM_RPT_ACCOUNT_ASSIGN
	DM_ACCOUNT_ASSIGN_HISTORY	CR_DATAEX CHNG	DM_RPT_ACCOUNT_ASSIGN_HISTORY
	DM_STRATEGY_ASSIGN	CR_DATAEX CHNG	DM_RPT_STRATEGY_ASSIGN
	DM_STRATEGY_ASSIGN_HISTORY	CR_DATAEX CHNG	DM_RPT_STRATEGY_ASSIGN_HISTORY
	DM_TM_USER_GROUP	CR_DATAEX CHNG	DM_TM_RPT_USER_GROUP
	DM_TM_USER_GROUP_MEMBERS	CR_DATAEX CHNG	DM_TM_RPT_USER_GROUP_MEMBERS

4

Batch Configuration Details

This topic provides information about the Batch Configuration Details.

This topic contains the following sub-topics:

- [Define workflow using Netflix conductor](#)
This topic describes the information about Defining workflow using Netflix conductor.
- [Pre-requisites to deploy conductor process](#)
This topic describes the information about Pre-requisites to deploy conductor process.
- [Branch EOD configuration through common core maintenance](#)
This topic describes the information about Branch EOD configuration through common core maintenance.
- [Success and Failure analysis](#)
This topic describes systematic instructions about the Success and Failure analysis.
- [Invoking EOD by using APIs](#)
This topic describes information about the Invoking EOD by using APIs.

4.1 Define workflow using Netflix conductor

This topic describes the information about 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

This topic describes the information about 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 Micro services Platform Foundation Installation Guide.

4.3 Branch EOD configuration through common core maintenance

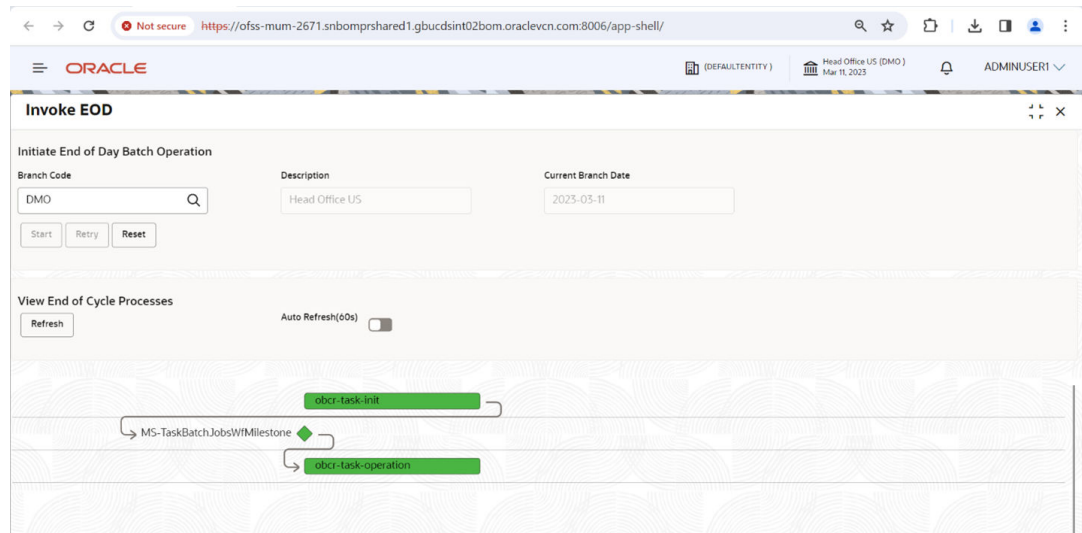
This topic describes the information about Branch EOD configuration through common core maintenance.

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

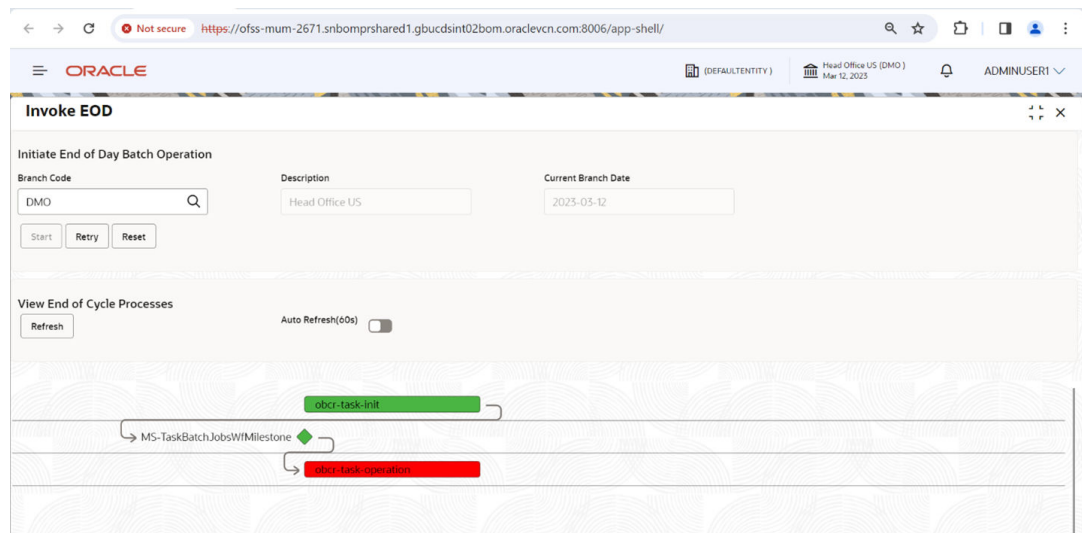
4.4 Success and Failure analysis

This topic describes systematic instructions about the 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.

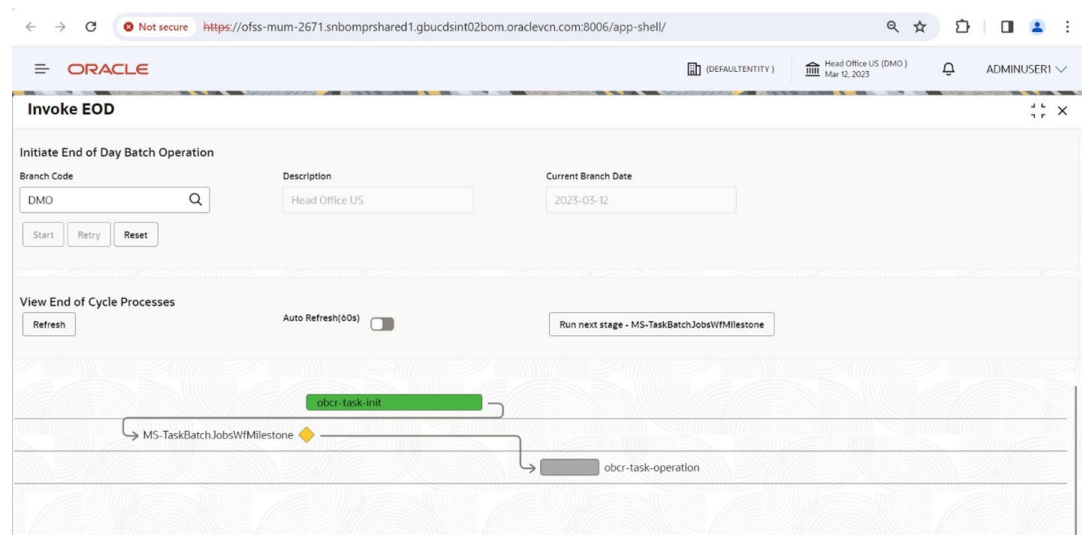
```
Select * from PLATO_BATCH_STREAM_RESULTS where JOB_NAME =
'<FAILED_JOB_NAME>';
```

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

This topic describes information about the Invoking EOD by using APIs.

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

This topic contains the following sub-topics:

- [Upload workflow on the conductor](#)
This topic describes the information about Upload workflow on the conductor.
- [Get the workflow details updated on conductor](#)
This topic describes the information about Get the workflow details updated on conductor
- [Initiate workflow](#)
This topic describes the information about Initiate workflow.
- [Track status of initiated workflow](#)
This topic describes the information about Track status of initiated workflow.
- [Crossing a Milestone Task](#)
This topic describes the information about Crossing a Milestone Task.

4.5.1 Upload workflow on the conductor

This topic describes the information about Upload workflow on the conductor.

Below endpoint should be invoked from the postman tool.

HTTP Method: POST

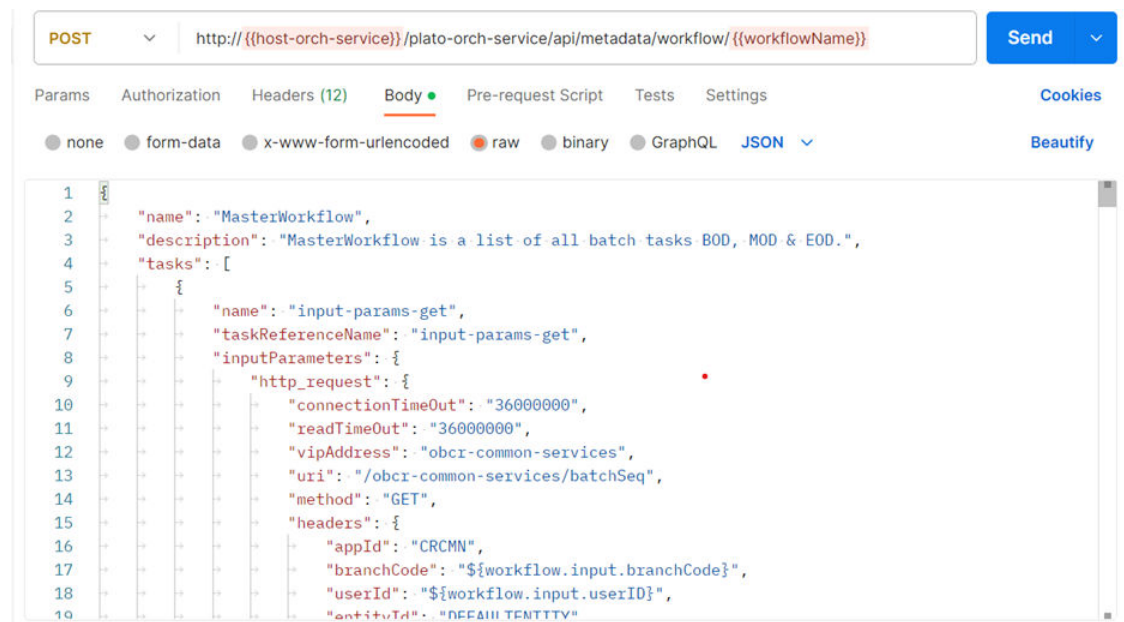
`http://{{host-orch-service}}/plato-orch-service/api/metadata/workflow/{{workflowName}}`

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

This topic describes the information about 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

This topic describes the information about 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.

Table 4-4 (Cont.) Parameters

Parameter Name	Value	Mandatory	Description
Userid	ADMINUSER1	Yes	User ID of the requesting user.

4.5.4 Track status of initiated workflow

This topic describes the information about 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

This topic describes the information about 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	Entity ID 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.

Index

B

Batch Configuration Details, [1](#)
Batch Execution Details, [1](#)
Batch Execution Sequence Diagram, [1](#)
Branch EOD configuration through common core maintenance, [3](#)

C

Case Closure Batch, [49](#)
Case Creation Batch, [27](#)
Correspondence Batch, [44](#)
Crossing a Milestone Task, [8](#)
Cured Account Data Transfer Job, [51](#)

D

Dashboard Data Population Batch, [40](#)
Data Transfer Job, [6](#)
Define workflow using Netflix conductor, [1](#)
Dialer Extract Batch, [41](#)

F

Fees and Charges Batch, [38](#)

I

Inbound for Mandatory Entities Creation, [2](#)
Inbound for Non-Mandatory Entities Creation, [3](#)
Increment DPD Batch, [30](#)
Initiate workflow, [7](#)
Internal Correspondence Batch, [45](#)
Introduction, [1](#)
Invoking EOD by using APIs, [5](#)

O

Outbound File Generation Batch, [50](#)

P

Pre-requisites to deploy conductor process, [3](#)
Promise Tracking Batch, [47](#)

R

Reporting Data Population Job, [53](#)

S

Score Calculation Batch, [28](#)
Segmentation Batch, [29](#)
Settlement Status Update Batch, [48](#)
Success and Failure analysis, [3](#)

T

Task Prioritization Batch, [52](#)
Tasks Initiation Batch, [31](#)
Track status of initiated workflow, [8](#)

U

Update Entity Batch, [4](#)
Upload workflow on the conductor, [6](#)
User Allocation Batch, [33](#)

V

Vendor Allocation Batch, [37](#)