Oracle® Banking Platform Collections and Recovery

Batch Execution Guide

Release 2.12.0.0.0

F43567-01

May 2022

ORACLE®

Oracle Banking Platform Collections and Recovery Batch Execution Guide, Release 2.12.0.0.0

F43567-01

Contents

List of Tables

Preface

This document helps you to understand the sequence in which the batches should be executed. It also helps you to improve the batch performance.

This preface contains the following topics:

Audience

This document is intended for the following audience:

- Implementation Team
- Consulting Team
- Development Team

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.

Organization of the Guide

This document contains:

Chapter 1 About this Guide

This chapter provides details about applicability of this guide.

Chapter 2 Introduction

This chapter presents an overview of the business process task flow.

Chapter 3 Batch Execution Sequence and Details

This chapter explains the sequence in which the batches should be executed.

Chapter 4 Technical Recommendations

This chapter recommends parameter values for better performance.

Related Documents

For more information, see the following documentation:

- For the configuration details to be done as part of day zero activity, see the Oracle Banking Platform Collections and Recovery Day Zero Setup Guide.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 About this Guide

This guide is applicable for the following products:

- Oracle Banking Platform (Oracle Banking Collections and Oracle Banking Recovery)
- Oracle Banking Enterprise Default Management (Oracle Banking Enterprise Collections and Oracle Banking Enterprise Recovery)

References to Oracle Banking Platform or OBP in this guide apply to all the above mentioned products.

2 Introduction

Oracle Banking Collections and Oracle Banking Recovery lets you integrate with the OBP Core Banking application.

Oracle Banking Collections processes all incoming delinquent and non-delinquent (preemptive) accounts from core banking.

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

Business Process Task flow

Following steps describe the task flow:

1. Oracle Banking Collections receives delinquent accounts from different host systems.
2. Cases are created for these delinquent accounts and are tracked through Promise To Pay (PTP).
3. When payments are applied to these accounts, case life cycle tracker brings these cases to closure if outstanding amount for the account is zero.
4. When all cases for an account are closed, Oracle Banking Collections marks these accounts as not in Collections.
5. Post charge-off, the accounts move into Oracle Banking Recovery. The recovery system then acts as a System of Records for all such accounts.
6. Further as a recovery system, Oracle Banking Recovery supports applying payments, expenses, and interest charging on such accounts.

Features

Following are the features of the batch processes:

- If the Batch Business Date is provided by the user through UI or command line, then batch should take this date and execute, else the default System Date will be used.
- Currently, DateUtility has two methods, getPostingDate and getCollectionSystemDateTime. For BATCH mode, the getPostingDate would return the date set as per above point.
- During implementation, you can use getPostingDate to retrieve value passed while batch job submission. This value can be used for No Activity check in No Activity Monitoring Algorithm.
- During implementation, you need to ensure that correct Business or Posting Date is provided while submitting the Job. This can be performed manually through UI or by writing your own script and using it to feed the Auto Scheduler. The onus of this would lie with implementation.

General Instructions for Implementation Team

Following are the general instructions for the implementation team to be followed while implementing:

1. All the batches need to provide the Batch Business Date while submitting the batch, else system date time would be considered for processing.

2. In batch mode, DateUtility.getPosting Date() would return batch business Date, if provided on UI, else it would return system date time. This would be applicable to all batches related to Oracle Banking Collections and Oracle Banking Recovery.
3. In online mode, DateUtility.getPosting Date() would return posting date after fetching it from host.
4. Your script should call DateUtility.getPosting Date() in online mode to fetch OBP posting date and pass the same as every batch parameter.
5. To get the posting date during implementation, you need to follow the below points:
    a. Write a Shell Script -> Call local Java Client -> Call Web service embedding the application service for Date.
    b. Update batch parameter files.
    c. If there is no application service, then you can put down your own app service which can call Date utility. By default, every app service in OBP is exposed as web service and hence new app service will also have corresponding web service.

3 Batch Execution Sequence and Details

The following table indicates the applicability of batches for Oracle Banking Collections and Oracle Banking Recovery:

Table 3–1 Batch applicability for Collections and Recovery

Serial No.	Batches	Product Name
1	File Parsing Batch (C1-FIPRS)	Collections
2	Parse Feeder Entity Batch (C1-PFENT)	Collections
3	Delinquency Identification Batch (C1-DELID)	Collections
4	Move Delinquent Account To Feeder Batch (C1-MVDEL)	Collections
5	Validate Incoming Data from Host Batch (C1-VALFD)	Collections
6	Create Entity Batch (C1-CRENT)	Collections
7	Update Host Data Batch (C1-UPENT)	Collections and Recovery
8	Increment Day Past Due Batch (C1-INCDP)	Collections and Recovery
9	Feeder Payment Batch (C1-FDPAY)	Collections
10	Derived Field Batch (C1-DRFLD)	Collections and Recovery
11	Promise To Pay Monitor Batch (C1-PTPM)	Collections and Recovery
12	Cure Monitor Batch (C1-FINCO)	Collections
13	Account Abandon Monitor Batch (WRITEOFF)	Collections
14	Strategy Monitor Batch (C1-CSMB)	Collections and Recovery
15	Case Status Automatic Transition Batch (C1-CSTRN)	Collections and Recovery
16	Queue Allocation Monitor Batch (C1-ALOCM)	Collections and Recovery
17	Collection Statistics Batch (C1-COLST, C1-CLDCS)	Collections
18	Suspend Activity Monitor Batch (C1-SUSMN)	Collections and Recovery
19	Bulk Contact Creation Batch (C1-BLKCC)	Collections and Recovery
20	Contact Processing Batch (C1-CNTPR)	Collections and Recovery
21	Display Priority Monitor Batch (C1-PRMON)	Collections and Recovery
22	User Allocation Monitor Batch (C1-USALC)	Collections and Recovery
23	Treatment Activity Monitor Batch (C1-TRMON)	Collections
24	Event Manager Batch (C1-EVENT)	Collections and Recovery
25	Case Lock Unlock Monitor - Unlock Cases (C1-CSCL)	Collections and Recovery

Serial No.	Batches	Product Name
26	Dialer Results Upload (C1-DLRRS)	Collections and Recovery
27	Agency or Vendor Upload (C1-VNDUP)	Collections and Recovery
28	Supervisory Statistics Batch Process (C1-SPSTM)	Collections and Recovery
29	Case List Table Refresh (C1-CLST)	Collections and Recovery
30	Account Time Barred Batch (C1-ACTBR)	Collections and Recovery
31	Account Charge off Batch (C1-ACTCH)	Collections
32	Partial Charge Off Batch (C1- PCHGO)	Collections
33	Dialer and IVR (O/B Robot) Extracts (C1-DIEXT)	Collections and Recovery
34	Account Dialer Extract Batch (C1-DLEXT)	Collections and Recovery
35	Case Lock Unlock Monitor – Task Creation (C1-CSTD)	Collections and Recovery
36	Account Setup in Recovery Batch (C1-ACTSU)	Recovery
37	Write off Monitor Batch - Recovery (C1-WRTOF)	Recovery
38	Apply Debt Sale Payments Batch - Recovery (C1-DBTSL)	Recovery
39	Account Closure Monitor Batch – Recovery (C1-PAYOF)	Recovery
40	Vendor Communication Outbound Interface Batch – Recovery (C1-VNDCM)	Recovery
41	Monthly Interest Computation Batch – Recovery (C1-INTCP)	Recovery
42	Monthly Account Statement Batch – Recovery (C1-RSTMT)	Recovery
43	Recovery Party warning indicator update Batch (C1-WIUPD)	Recovery
44	Settlement Offer Creation Batch (C1-STLOF)	Recovery

The following table indicates the sequence in which the batches should be executed:

Table 3–2 Batch Sequence Execution

Batches	Batch Sequence
End of Day Batches	Following batches should be run before EOD in the specified order: ■ PTP Tracking ■ Cure Monitor ■ Write-off Monitor ■ Charge Off Monitor

Batches	Batch Sequence
	■ Partial Chargeoff Batch ■ Time Barred Batch ■ Dialer Results Upload ■ Agency or Vendor Upload ■ Write Off Monitor ■ Account Closure Monitor ■ Apply Debt Sale Payments ■ Collection Statistics ■ Case Lock Unlock Monitor - Unlock Cases
Beginning of Day Batches	Following batches should be run after EOD in the specified order: ■ File Parsing Batch ■ Parse Feeder Entity Batch ■ Delinquency Identification for OBP ■ Move delinquent accounts to Collections feeder tables for OBP ■ Validate Incoming data from host ■ Create Entity ■ Increment DPD ■ Derived fields ■ Account Setup in Recovery ■ Suspend Activity Monitor ■ Bulk Contact Creation ■ Contact Processing ■ Strategy Monitor ■ Treatment Activity Monitor ■ Settlement Offer Creation Batch ■ Agency or Vendor Upload – Settlement Offer ■ Queue Allocation Monitor ■ Display Priority Monitor ■ User Allocation Monitor ■ Queue and Agent Statistics ■ Vendor Communication Outbound Interface Batch ■ Account Dialer Extracts

Batches	Batch Sequence
	■ Event Manager
Multiple times during the Day Batches	Following batches should be run multiple times during the day in the specified order: ■ Case Status Automatic Transition Batch ■ Update Entity ■ Recovery Party warning indicator update Batch ■ Payment handling ■ Case Lock Unlock Monitor – Task Creation ■ Case List Table Refresh (C1-CLST)
End Of Month Batches	■ Interest Calculation ■ Monthly Account Statement Batch

3.1 File Parsing Batch (C1-FIPRS)

The **File Parsing Batch (C1-FIPRS)** is first in series of two batches, which will transfer data from the external file to collections feeder (staging) tables. This batch will read the data from the external file uploaded on the server and move all the data to an intermediately table.

Setup Prerequisites

Following are the prerequisites for the setup:

1. External data file present on the server. Server path is mentioned in the feature configuration C1-FIPR.

Dependent Batch

- Parse Feeder Entity Batch

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–3 File Parsing Batch (C1-FIPRS)

Parameter	Mandatory (Yes or No)	Description
Source Host ID	Yes	Comma separated list of Host IDs of the source file to be picked up for processing.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

The process status of the files processed in the batch can be checked in the table - CI_RQST_FDR_DATA_FILES. Successful file processing is marked as 'C' and unsuccessful will be marked as 'P'.

The request data that is not processed can be checked in the error table - ERR$_CI_RQST_FDR_DATA.

After the batch completion, there will be a single row for each request record in the table - CI_RQST_FDR_ DATA.

3.2 Parse Feeder Entity Batch (C1-PFENT)

The Parse Feeder Entity Batch (C1-PFENT) is second in series of two batches, which will transfer data from the external file to collections feeder (staging) tables. This batch will read the data from the intermediately staging table CI_RQST_FDR_DATA and move the respective data to collections main feeder tables.

Setup Prerequisites

Following are the prerequisites for the setup:

1. First File Parsing Batch (C1-FIPRS) need to be executed successfully.

Dependent Batch: Not Applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–4 Parse Feeder Entity Batch (C1-PFENT)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

The data from the intermediary table - CI_RQST_FDR_DATA should be moved to the feeder data into respective feeder tables for each feeder entity.

The request data that is successfully processed and moved into respective feeder table is marked as 'S' in the intermediary table.

If processing fails for the request data, it's process status is marked as 'F' in the intermediary table.

3.3 Delinquency Identification Batch (C1-DELID)

The **Delinquency Identification (C1-DELID)** batch is used to identify delinquent accounts for OBP. The filter conditions passed as parameters to the batch scan all accounts in OBP and mark the account as delinquent when a filter condition is satisfied. The filters must be defined in order of their priority that is, filter with highest priority must be defined first.

Setup Prerequisites

Following are the prerequisites for the setup:

1. Views created on the OBP tables should be available in Oracle Banking Collections and Oracle Banking Recovery database schema.

   Delinquency is identified by using data from the following views/tables:

- FLX_DD_COL_DATA_XF_VW (CASA/LOC Accounts)
- FLX_DD_COL_DATA_TOD_XF_VW (CASA/LOC Accounts)
- FLX_LN_COL_FD_ACCT_VW (TL Accounts)
- FLX_AC_COL_FD_ACCT_ARS_VW (TL Accounts)
- FLX_AC_COL_FD_ACCT_PAY_TRACK_VW (TL Accounts)
- CI_LN_COL_FD_ACCT_OPTMZD (TL Accounts)
- CI_AC_COL_FD_ACCT_ARS_STAGE (TL Accounts)

Updatable views are:

- FLX_DD_COL_DATA_XF_UPD_ACCT_VW (CASA/LOC Accounts)
- FLX_DD_COL_DATA_XF_UPD_EXTN_VW (CASA/LOC Accounts)
- FLX_LN_COL_ACCT_UPDATE_VW (TL Accounts)
- FLX_DD_COL_BILL_DATA_XF_VW (CASA/LOC Accounts)

2. Facts based on which the filters are created, must be available in the system.
3. Filters to identify delinquent accounts for OBP data should be created. The product pre-ships five delinquency identification filters as mentioned below:
   - **TL filters**
     - DebitBalance
     - NonPayment
     - PartialPayment
     - LoanUnpaidBillsODAmt
     - LoanTrackerUnpaidBills
   - **CASA filters**
     - Drawal_LimitPeriod
     - Drawal_ExpiryOfLimit
     - Unpaid Bills

Dependent Batch

- Move delinquency accounts to Feeder (Collections)
- Incoming Data from Host

Multi-threaded: No

You can specify the following parameters while executing this batch:

Table 3–5 Delinquency Identification Batch (C1-DELID)

Parameter	Mandatory (Yes or No)	Description
Delinquency Filter	Yes	Used to specify the filters that are used to identify delinquent accounts. The filters are defined in order of their priority that is, from highest priority to lowest priority. Multiple filters are separated by a comma. For integration with OBP Host, this filter is product shipped. See the Oracle Banking Platform Collections and Recovery Day Zero Setup Guide for details of this filter.
Type Of Account	Yes	Used to specify the type of account on which the filters are executed. The valid values are CASA or TL.
DPD Holiday Factor	No	Days Past Due based on Holidays to be considered while identifying accounts for delinquency.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

On successful completion of this batch, it updates delinquency flags in host for the accounts filtered by the condition builder. If an account is identified as not delinquent, then this batch process updates the Delinquency Flag as 'N' while the Collections Flag still remains as 'Y'. The Delinquency Reason and Delinquency Start Date are updated as NULL during this stage.

3.4 Move Delinquent Account To Feeder Batch (C1-MVDEL)

The **Move Delinquent Account to Feeder (C1-MVDEL)** batch is used to move the delinquent accounts to Collections feeder tables. This batch should be run once the Delinquency Identification batch completes. It uses a stored procedure (AP_COLL_FDR_DATA_EXTRACT) to move data from OBP to collection feeder tables.

Setup Prerequisites

Following are the prerequisites for the setup:

1. Views created in OBP tables to fetch various details of delinquent customers should be available with associated data. List of views used are mentioned below:
    - FLX_COL_ACCT_DATA_XF
    - FLX_AC_COL_FD_ACCT_ARS_VW
    - FLX_LN_COL_FD_SCH_VW
    - FLX_AC_COL_FD_ACCT_PAY_TRACK_VW
    - FLX_COL_ACCT_HRDSHIP_VW
    - FLX_COL_ACCT_WARN_IND_DATA_XF
    - FLX_PI_COL_FD_ACCT_PER_VW

- FLX_PI_COL_FD_PER_VW
- FLX_PI_COL_FD_PARTY_IDENT_VW
- FLX_PI_COL_FD_PER_NAME_VW
- FLX_PI_COL_FD_PER_ADDR_VW
- FLX_PI_COL_FD_PER_WARN_IND_VW
- FLX_PI_COL_FD_EMP_PROF_VW
- FLX_PI_COL_FD_CONTACT_PREF_VW
- FLX_LM_COL_FD_COL_ENTITY_VW
- FLX_LM_COL_FD_COLLATERAL_VW
- FLX_LM_COL_FD_COL_PARTY_VW
- FLX_LM_COL_FD_COL_CHRG_VW
- FLX_LM_COL_FD_COL_GRNTR_VW
- FLX_LM_COL_FD_INSR_DTLS_VW
- FLX_DD_COL_BILL_DATA_XF_VW
- FLX_LZ_UZ_SCRA_HIST_DTLS_VW (Localized view)
- FLX_AC_COL_FD_ACCT_PAY_TRACK_VW
- FLX_LZ_UZ_PI_COL_FD_PER_VW (Localized view)
- FLX_LZ_LM_COL_FD_COL_AUTO_VW(Localized view)

2. Configuration or mapping tables mentioned below should be properly defined.
    - **Source to Collection Class Mapping: CI_HOST_MAIN_CUST** - To identify main customer, financially responsible and guarantor relationship types.
    - **Source to Contract type mapping: CI_HOST_SA_TYPE** - To derive contract type to be used for product class, product group.
    - **Source to product group mapping: CI_HOST_PROD_CL** - To derive customer class, collection class and debt class for product class and product group.
    - **CI_BANK_MST** - List of banks created in host
    - **CI_BANK_BRANCH_MST** - List of bank branches created in host
    - **CI_PRIM_NAMETYPE** - Primary name type for host.

Dependent Batch

- Validate Feeder batch
- Create Entity batch

Multi-threaded: No

You can specify the following parameters while executing this batch:

Table 3–6 Move Delinquent Account To Feeder Batch (C1-MVDEL)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.

Post Execution Check / Clean Up

On successful completion of this batch, the data is inserted into the feeder tables.

Verify CI_FDR_PER and CI_FDR_ACCT tables for party and accounts pulled from host.

Whenever C1-MVDEL encounters any error during execution, batch will abort by logging the issue in the error log table listed below.

err$_CI_FDR_ACCT

err$_CI_FDR_ACCT_ARS

err$_CI_FDR_REPAYMENT_SCH

err$_CI_FDR_ACCT_HARDSHIP_DTLS

err$_CI_FDR_ACCT_WARNING_IND

err$_CI_FDR_ACCT_PER

err$_CI_FDR_PER

err$_CI_FDR_PER_ID

err$_CI_FDR_PER_NAME

err$_CI_FDR_PARTY_WARNING_IND

err$_CI_FDR_PER_EMPLOYMENT_PRO

err$_CI_FDR_PER_ADDR

err$_CI_FDR_CONTACT_PREF

err$_CI_FDR_COLLATERAL_ENTITY

err$_CI_FDR_COLLATERAL

err$_CI_FDR_COLLATERAL_PARTY

err$_CI_FDR_COLLATERAL_CHRG

err$_CI_FDR_COLLATERAL_GRNTR

err$_CI_FDR_INSR_DTLS

3.5 Validate Incoming Data from Host Batch (C1-VALFD)

The **Incoming Data from Host (C1-VALFD)** batch is used to validate the records in feeder tables. It uses a stored procedure (AP_COLL_FDR_DATA_VALIDATE) to validate the data.

Validation includes availability of lookup data, admin data and configuration related data. PROCESS_ STATUS column in each feeder table is used to identify if the record can be processed. Records that fail validation are marked as 'F' and message category or number will be updated against these records.

Other than validation, this batch updates some columns like division, contract type, and so on (derived from configuration tables) in CI_FDR_ACCT table.

Setup Prerequisites

Following are the prerequisites for the setup:

1. Delinquent accounts should be available in the feeder tables
2. Admin data and lookups

Dependent Batch

- Create entity batch

Multi-threaded: No

You can specify the following parameters while executing this batch:

Table 3–7 Validate incoming Data from Host Batch (C1-VALFD)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.

Post Execution Check / Clean Up

On successful completion of this batch, feeder data is marked failed for the records that fail validation.

Check PROCESS_STATUS, MESSAGE_CAT_NBR and MESSAGE_NBR column in CI_FDR_PER table

3.6 Create Entity Batch (C1-CRENT)

The **Create Entity (C1-CRENT)** batch is used to select records from feeder tables and create entries for delinquent accounts.

Entity creation involves following entity objects:

- Party / Person (and related data)
- Account (and related data)
- Party account relationship
- Mainline
- Collateral
- Insurance

Setup Prerequisites

Following is the prerequisite for the setup:

1. Validated data should be available in feeder tables. Records with PROCESS_STATUS = 'P' will be processed.

Dependent Batch

- Strategy Monitor batch
- Increment DPD batch

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–8 Create Entity Batch (C1-CRENT)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.

Post Execution Check / Clean Up

On successful completion of this batch, party and account data is created in Collections. TO DO is created for failed records from feeder tables. Records for successfully created data are deleted from feeder tables.

Verify CI_PER and CI_ACCT for party and accounts created in Collections.

3.7 Update Host Data Batch (C1-UPENT)

The **Update Host Data (C1-UPENT)** batch is used to update the Collections tables based on the updates received from host systems in the feeder tables.

Setup Prerequisites

Following is the prerequisite for the setup:

1. Data is available in feeder tables with updates from hosts. CI_FDR_HOST_UPDATES table should have list of table names whose updates have come from host.

Dependent Batch: Not Applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–9 Update Host Data Batch (C1-UPENT)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Parameter	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.
Feeder Table Group Id	No	Enter a valid feeder table group id as configured in the C1-FDRGRPS feature config. Multiple group ids can be provided as comma separated values.
Feeder Table Name	No	Enter a valid feeder table name for processing. Multiple table names can be provided as comma separated values.

Post Execution Check / Clean Up

On successful execution of this batch, updated records are moved to the collection tables.

Records in CI_FDR_HOST_UPDATES should be deleted for successfully processed updates.

3.8 Increment Day Past Due Batch (C1-INCDP)

The **Increment Day Past Due (C1-INCDP)** batch is used to increment the DPD for an account if no updates are received from the host system. In addition, this batch also updates the bucket value for all accounts even if the accounts DPD is not incremented.

Setup Prerequisites

Following is the prerequisite for the setup:

1. Bucket configuration should be available in CI_COLLECT_BUCKET_MST table.

Dependent Batch

- Display Priority Monitor

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–10 Increment Day Past Due Batch (C1-INCDP)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.
Exclude Source Host ID	No	Used to specify the source host IDs that are to be excluded from the batch processing. Multiple source host IDs are placed in single quotes and are separated by comma. For example, 'A', 'B', 'C'. If you leave this field blank, then the batch will be executed for all the host IDs.

Post Execution Check / Clean Up

On successful execution of this batch, DPD values are updated or are incremented by the difference between system date and last DPD update date. This can be verified in CI_PARTY_COLLECT table.

Also, the account buckets are updated and can be verified in CI_ACCT_BUCKET table.

3.9 Feeder Payment Batch (C1-FDPAY)

The **Feeder Payment (C1-FDPAY)** batch is used to create payments that have failed during online payment creation and are received from the host system.

Setup Prerequisites

Following are the prerequisites for the setup:

1. Account for which payment is received from host must be available in Collections.
2. Collections case should be in open status and at least one active contract should be present on account.
3. FT freeze algorithm should be attached to adjustment type.

Dependent Batch

- PTP Tracking

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–11 Feeder Payment Batch (C1-FDPAY)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.

Post Execution Check / Clean Up

On successful completion of this batch, payment is marked, accounting entries are passed, and overdue balances are updated. Feeder records are deleted.

Payment records successfully created can be verified from CI_PAY, CI_PAY_EXTN, CI_PAY_EVENT tables.

3.10 Derived Field Batch (C1-DRFLD)

The **Derived Field (C1-DRFLD)** batch is used to update columns derived from Collections data. The following derived fields are updated:

- Secured Y/N
- PPI Insured Y/N
- LMI Insured Y/N

- Co-borrowers Exist Y/N
- Installment & Non-Installment Amounts
- Total Payment since last 03 Months
- Days since Account Opened
- Days since Charge-off
- Days Since Dispute Claimed
- Days with Agency
- Days since Bankruptcy Filed
- Account Bankruptcy Switch
- Guarantor Switch
- Account level Service member Flags
- Arrearage Plan Overdue Amount
- Days Past Due in Arrearage
- Confirm Plan Overdue Amount
- Confirm Plan Days Past Due
- Days Since Last Contact
- Days Since Last Payment
- Days Since Last Settlement Offer

Any other business logic to be derived other than the ones mentioned above can be done using an algorithm created on DerivedFieldAlgorithmSpot and passing it to the batch.

Setup Prerequisites

Following are the prerequisites for the setup:

1. Incoming data from host must be available in Collections.
2. Derived field exclusion setup (CI_DERIVED_EXCLUSION) should be available.

Dependent Batch: Not Applicable

Multi-threaded: No

You can specify the following parameters while executing this batch:

Table 3–12 Derived Field Batch (C1-DRFLD)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Algorithm Code	No	Used to specify an algorithm name, which will be invoked by the batch process for additional logic to be performed.
Validation Date	Yes	Enter SYSTEM DATE to perform date validation with system date

Parameter	Mandatory (Yes or No)	Description
		or POSTING DATE to perform date validation with business date.
Timezone Derivation Algorithm Code	No	Used to specify an algorithm name, which will be invoked by the batch process and will update timezone of the person.

Post Execution Check / Clean Up

On successful completion of this batch, the derived fields are updated. Verification can be done in CI_ACCT_EXTN and CI_ACCT_DERIVED_FACT tables.

3.11 Promise To Pay Monitor Batch (C1-PTPM)

The **Promise to Pay Monitor** batch is used to monitor pending and active PTPs. This batch monitors the PTPs payer and the service-level agreement of the same debt class. This batch process also determines if a PTP has been kept, broken, or active.

Setup Prerequisites

Following are the prerequisites for the setup:

1. Person, Account, and Contract must be created for the delinquent accounts.
2. PTP should have been created on accounts for processing.

Dependent Batch

- For EOD, Strategy monitor is dependent.

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–13 Promise To Pay Monitor Batch (C1-PTPM)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

On successful completion of this batch, PTP status is updated for accounts. Verification can be done in CI_PTP table.

3.12 Cure Monitor Batch (C1-FINCO)

The **Cure Monitor (C1-FINCO)** batch is used to close cases and move account out of Collections.

Setup Prerequisites

Following are the prerequisites for the setup:

1. Condition builder must be created to filter the account to be finalized.
2. Host API should be available to mark account Not in Collections.

Dependent Batch

- Strategy Monitor Batch

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–14 Cure Monitor Batch (C1-FINCO)

Parameter	Mandatory (Yes or No)	Description
Filter Name	Yes	Used to specify the filter ID to fetch the accounts that need to be cured.
Promise To Pay Cancellation Reason	Yes	Used to specify the cancellation reason to cancel PTP.
Cure Monitor View	Yes	Used to view the name of cure monitor.
Cure Entity Algorithm Code	No	Used to specify the algorithm to close the contract if there are no open cases on an account.
Customer Level in collections switch	No	Used to specify to cure account if it is linked to a party level case.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

On successful completion of this batch, accounts are marked as Not in Collections in host and in Collections. Contract will be stopped, case will be moved to final status and active PTP is cancelled.

Verification can be done from CI_CASE, CI_PTP and CI_ACCT_EXTN tables.

3.13 Account Abandon Monitor Batch (WRITEOFF)

The **Account Abandon Monitor (WRITEOFF)** batch is used to request for a complete account write-off either in the Straight through Processing (STP) or in Non-STP (Manual) modes.

Setup Prerequisites

Following is the prerequisite for the setup:

1. Set up Rule Engine and Condition builder. Derived facts will be used to retrieve write off type from rule engine.

Dependent Batch: Not Applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–15 Account Abandon Monitor Batch (WRITEOFF)

Parameter	Mandatory (Yes or No)	Description
Account Write Off View	Yes	Used to filter accounts for write off.
Filter Name	Yes	Used to specify the filter name configured in the Filter Definition screen (Back Office > Rules) to filter accounts that needs to be written off.
Rule Name	Yes	Used to specify the rule name configured in the Rule Author screen (Back Office > Rules) to provide write off type.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

On successful completion of this batch, write off request date is updated in Collections andrequest is sent to host for account write off. Verification can be done in the CI_ACCT_EXTN table.

3.14 Strategy Monitor Batch (C1-CSMB)

The **Strategy Monitor (C1-CSMB)** batch is used to create cases for delinquent accounts based on the case type retrieved from the rule. If TAM details are configured in the rule, these will be updated for the new case being created. This batch creates only account level cases.

The possible output expected from the rule is:

- Collection Strategy (Case Type)
- Re-evaluation days
- TAM Matrix ID
- TAM_REVIEW_DT

Setup Prerequisites

Following are the prerequisites for the setup:

1. Rules / Rule set for case creation must be defined in the rules engine.
2. Case types with status should be defined. Algorithms should be attached to case status.
3. PTP Tracking batch should be complete.
4. CI_ADM_RVW_SCH should have records with NEXT_CR_RVW_DT <= system date.
5. C1-DFLTCFG: If no case type is returned by the Rules engine, a default case type is assigned to the account. Default case type is configurable using this feature configuration.

Dependent Batch

- Case Status Automatic Transition batch
- Queue Allocation batch

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–16 Strategy Monitor Batch (C1-CSMB)

Parameter	Mandatory (Yes or No)	Description
Rule to retrieve case type and re-evaluation days	Yes	Used to specify the Rule Set ID or Rule ID to retrieve case type and re-evaluation days.
isRuleSet flag	Yes	Used to specify a value to call the rule set. Enter 'Y' to call a Rule Set or enter 'N' to call a Rule.
Rule Fact Population Algorithm	Yes	Used to specify an algorithm code for rule fact population. (C1-BRLSR algorithm type pre-shipped)
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

On successful completion of this batch, cases are created for delinquent accounts. If CSAM algorithm is attached to the case status, then it will trigger and take actions as defined in the CSAM admin configuration.

Cases created can be verified in CI_CASE, CI_CASE_EXTN, CI_CASE_PARTY and CI_CASE_ ASSOCIATION tables.

3.15 Case Status Automatic Transition Batch (C1-CSTRN)

The **Case Status Automatic Transition (C1-CSTRN)** batch is used to call the algorithm that determines whether a case state should be transitioned to a new state.

Setup Prerequisites

Following is the prerequisite for the setup:

1. Case type status should be defined.

Dependent Batch

- Queue Allocation Monitor batch

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–17 Case Status Automatic Transition Batch (C1-CSTRN)

Parameter	Mandatory (Yes or No)	Description
List of Case Types to be excluded	No	Enter comma separated case type to be excluded from Case Status Auto Transition.
Restrict To Cases in Status Code	No	Used to specify the case status code to limit the process to cases in this status.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before the run terminates.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

Cases will be moved to the next status. Verify this from CI_CASE table.

3.16 Queue Allocation Monitor Batch (C1-ALOCM)

The **Queue Allocation Monitor (C1-ALOCM)** batch is used to allocate cases to queues. Cases which are to be allocated to the queues are filtered by the condition builder attached to the allocation group and are assigned in round-robin method to the queues that are linked to the allocation group.

Setup Prerequisites

Following are the prerequisites for the setup:

1. Queues and Allocation Groups should be configured.
2. Overflow queue should be attached to allocation groups.
3. Queue allocation algorithm should be attached to allocation group.
4. Condition builder to filter accounts for allocation should be attached to allocation group.
5. Feature configuration - C1-DFLTCFG: If no case type is returned by the Rules engine, a default case type is assigned to the account. Default case type is configurable using this feature configuration

Dependent Batch

- User Allocation batch
- Display Priority Monitor batch

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–18 Queue Allocation Monitor Batch (C1-ALOCM)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

On successful completion of this batch, cases are allocated to queues.

Verification can be done in CI_QUEUE_ASSIGN table.

3.17 Collection Statistics Batch (C1-COLST, C1-CLDCS)

The **Collection Statistics (C1-COLST)** batch is used to create statistics from collection data. The following columns are updated:

- PTP's Broken (Derived from CI_PTP table)
- PTP's Kept (Derived from CI_PTP table)
- Consecutive PTP's Broken (Derived from CI_PTP table)
- Peak OD days (Derived from CI_PARTY_COLLECT.DAYS_PAST_DUE)
- PTP's taken (Derived from CI_PTP table)
- Last acted agent (Derived from CI_CASE_LOG.USER_ID)

C1-CLDCS batch is used to generate delinquency and cycle string.

Setup Prerequisites

Following are the prerequisites for the setup:

1. Account and Customer data should be available in Collections.
2. Bucket Admin configuration should be available.

Dependent Batch: Not Applicable

Multi-threaded:

C1-COLST: No

C1-CLDCS: Yes

You can specify the following parameters while executing this batch:

Table 3–19 Collection Statistics Batch (C1-COLST, C1-CLDCS)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before the run terminates.

Post Execution Check / Clean Up

Statistics data will be updated based on data available. Verification can be done in CI_COLLECT_STAT table.

3.18 Suspend Activity Monitor Batch (C1-SUSMN)

The **Suspend Activity Monitor (C1-SUSMN)** batch is used to identify activities that must be stopped or suspended when a set of conditions is satisfied for a customer or an account. This batch also removes the Suspend flag if conditions are not satisfied.

Setup Prerequisites

Following is the prerequisite for the setup:

1. Condition builder must be set up and linked to suspend activities in suspend activity admin setup.

Dependent Batch

- Strategy Monitor batch
- Queue Allocation Monitor batch
- Case Status Automatic Transition batch
- Bulk Contact Creation batch

Multi-threaded: No

You can specify the following parameters while executing this batch:

Table 3–20 Suspend Activity Monitor Batch (C1-SUSMN)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before the run is terminated.
Suspend Activity View Name	Yes	Indicates the name of the view, based on which the records are selected for suspension. CI_SUSPEND_ACTIVITY_VW view is pre-shipped from product.

Post Execution Check / Clean Up

On successful completion of this batch, flags are set for accounts or customers that satisfy the filter conditions defined in the suspend activities.

Verification can be done from CI_SUSP_ACCT_ACTY and CI_SUSP_PER_ACTY tables.

3.19 Bulk Contact Creation Batch (C1-BLKCC)

The **Bulk Contact Creation (C1-BLKCC)** batch is used to generate contacts (of type Letters, E-mail, and SMS) in volume based on certain conditions.

Setup Prerequisites

Following is the prerequisite for the setup:

1. Bulk Contact admin configuration should be defined and condition builder to filter accounts should be attached to each record.

Dependent Batch

- Contact Processing batch

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–21 Bulk Contact Creation Batch (C1-BLKCC)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before the run terminates.
Bulk Contact View Name	No	Used to specify the view name to be queried for list of accounts. CI_BULK_CONTACT_VW view is pre-shipped from product.
Bulk Contact View Name for Settlement Offer	No	Used to specify the view name to be queried for list of Recovery Accounts to create Settlement Offer Letter. CI_BULK_CONTACT_SETT_OFFER_VW view is pre-shipped from product. This view will be used by Process Codes which have Settlement Offer Checked in Bulk Contact Admin.

It is mandatory to provide at least one view name in the parameter.

Post Execution Check / Clean Up

On successful completion of this batch, customer contacts are created for contact processing batch to generate associated letters.

Verification can be done from CI_CC table.

3.20 Contact Processing Batch (C1-CNTPR)

The **Contact Processing (C1-CNTPR)** batch is used to select un-processed customer contacts and invoke the letter template extract algorithm, which generates associated Letters, SMS, or e-mails.

Setup Prerequisites

Following is the prerequisite for the setup:

1. For each contact type, an extraction algorithm should be associated from contact type configuration screen.

Dependent Batch: Not Applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–22 Contact Processing Batch (C1-CNTPR)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.
Process Failed Contacts	No	Used to specify the option to process failed contacts. Enter 'Y' to process failed contacts and 'N' to cancel processing failed contacts.

Post Execution Check / Clean Up

On successful completion of this batch, all contacts that have been processed are marked as processed.

Verification can be done in CI_CC table. Records failed during processing can be verified from NON_ DELIVERY_REASON in CI_CC table.

3.21 Display Priority Monitor Batch (C1-PRMON)

The **Display Priority Monitor (C1-PRMON)** batch is used to set priority of a case. This is used while displaying cases on the user interface.

Setup Prerequisites

Following are the prerequisites for the setup:

1. Define **Display Order** and **Arranged By** for queue from queue details admin setup.
2. Define work list required at queue level.

Dependent Batch: Not Applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–23 Display Priority Monitor Batch (C1-PRMON)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.

Post Execution Check / Clean Up

On successful completion of this batch, the display order of all open cases is set. Verification can be done in the CI_QUEUE_ASSIGN table.

3.22 User Allocation Monitor Batch (C1-USALC)

The **User Allocation Monitor (C1-USALC)** batch is used to allocate cases to users, teams, or vendors associated within queues.

Setup Prerequisites

Following are the prerequisites for the setup:

1. Allocation group and queues should be configured.
2. User allocation algorithm should be mapped to queues.
3. Collection users, teams or vendors should be mapped to queues in queue details screen.

Dependent Batch: Not Applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–24 User Allocation Monitor Batch (C1-USALC)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.

Post Execution Check / Clean Up

On successful completion of this batch, cases are allocated to users, teams, or vendors.

Verification can be done in CI_USER_ASSIGN and CI_VENDOR_ASSIGN for users and vendors respectively.

3.23 Treatment Activity Monitor Batch (C1-TRMON)

The **Treatment Activity Monitor (C1-TRMON)** batch enables the Collections system to provide a framework that helps in the following activities:

- Associate cases to Treatment Activity Monitor decision matrix.
- Invoke Treatment Activity Monitor decision matrix.
- Perform actions based on the Treatment Activity Monitor decision matrix outcomes.
- Modify or Remove Treatment Activity Monitor decision matrix associations.
- Batch picks up cases which are not on hold and where TAM_REVIEW_DT <= business date.

Setup Prerequisites

Following is the prerequisite for the setup:

1. Cases should be available with TAM MATRIX ID's.

Dependent Batch: Not Applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–25 Treatment Activity Monitor Batch (C1-TRMON)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Case Type	No	Used to specify the case type to be used for treatment activity monitor batch.
Treatment Activity Monitor Algorithm Code	Yes	Used to specify the TAM algorithm code to be used for processing of all the selected cases.

Post Execution Check / Clean Up

Based on the condition defined in TAM algorithm, necessary actions will be taken. TO DO will be created if there is an error while processing the algorithm.

3.24 Event Manager Batch (C1-EVENT)

The **Event Manager (C1-EVENT)** batch is used to execute all events attached to condition category and perform all outcomes attached to events.

Setup Prerequisites

Following is the prerequisite for the setup:

1. Validate events are configured for condition category.

Dependent Batch: Not Applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–26 Event Manager Batch (C1-EVENT)

Parameter	Mandatory (Yes or No)	Description
Event Manager Batch Entity	Yes	Value should be one of the following: Enter Value = GLCU: Events with EventType = 'Global' and ConditionOn = 'CUST' Enter Value = GLAC: Events with EventType = 'Global' and ConditionOn = 'ACCT' Enter Value = GLCS: Events with EventType = 'Global' and ConditionOn = 'CASE' Enter Value = CACS: Events with EventType = 'CASE' and ConditionOn = 'CASE'
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

On successful completion of this batch, all outcomes attached to condition events are executed, log entry is created for successful executed outcomes in the Event Log table with status as SUCCESS. Log entry is also created for failed records in Event Log table with status as FAILED.

3.25 Case Lock Unlock Monitor - Unlock Cases (C1-CSCL)

While a user is working on a case, it gets locked for that user. There can be scenarios where the user who locks the case did not release the case for other users to make an update. Like system crash, closure of browser or intentional delay.

Setup Prerequisites: Not Applicable

Dependent Batch: Not Applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–27 Case Lock Batch (C1-CSCL)

Parameter	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Time in minutes	Yes	Used to compare the time with the locked time
Locked by User	No	Optional to filter cases created by entered user
Case Type	No	Optional to filter cases created by entered Case Type

Post Execution Check / Clean Up

On successful completion of this batch, the cases which are locked for more than the stipulated amount of time will get unlocked.

3.26 Dialer Results Upload (C1-DLRRS)

Dialer and IVR systems provide a response file for all the calls made during the course of the day. This information needs to be uploaded in Collections, that is the Staging area. This batch program will read the data from the staging area and upload the results.

Setup Prerequisites

Following is the prerequisite for the setup:

1. A new master (admin) screen is needed for mapping the IVR and dialer results. There will be one set of configuration for Dialer and one for IVR file. This will be done from the same screen.

Dependent Batch:

- Not Applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–28 Dialer Results Upload (C1-DLRRS)

Parameter Name	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates
Execution Status	No	Enter a value ('E' or 'ERROR') to execute all the records in staging area having status as 'ERROR'

Post Execution Check / Clean Up:

- For Entity-Contact, contacts will be created and action results will be logged against those contacts.
- For Entity-Case, action results will be logged against all the cases related to a customer or account.
- On failure TO DO will be created for failed records.

3.27 Agency or Vendor Upload (C1-VNDUP)

A bank may hire vendors, third party agencies to do various collection activities. In Offline Communication mode, vendors will follow up with the customer, execute tasks, and share the results and records in the form of a file.

The information from the file is uploaded in a temporary staging area. The Vendor Upload batch processes the information in the temporary staging area and upload it into collections or recovery system.

Vendors can provide updates on below entities to Oracle Banking Collections:

- Follow Up
- Promise to Pays
- Settlement

Setup Prerequisites

Following is the prerequisite for the setup:

1. Vendor ID provided should be registered.
2. User ID provided should be either configured against registered Vendor ID or will be the same as registered Vendor ID.
3. Program that uploads data from Vendor files in Collection or Recovery staging table.

Dependent Batch:

- Not Applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–29 Agency or Vendor Upload (C1-VNDUP)

Parameter Name	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates
Vendor Upload Entity	Yes	Enter one of the entity names to be uploaded to Collections or Recovery: Followup, PTP, or Settlement
Process Failed Records Switch	No	Y: Process the Records failed in a previous batch run N: Do Not Process the Records failed in a previous batch run
toDoType	No	Task Type Code for task to be created in case offer goes for approval. **Note:** This parameter is applicable only when Vendor Upload Entity is Settlement.
queueCode	No	Queue to which task created above will be assigned. **Note:** This parameter is applicable only when Vendor Upload Entity is Settlement.

Post Execution Check / Clean Up:

- Follow-up is created and action results are logged against the case related to a customer or account.
- PTP is created for the Account associated with the Case.
- Settlement Offer should be created on failure, TO DO will be created for failed records.

3.28 Supervisory Statistics Batch Process (C1-SPSTM)

This batch process is used for calculating the supervisory statistics. This batch calculates the important Queue and Agent level statistics at the Beginning of Day.

Setup Prerequisites

Following is the prerequisite for the setup:

1. Queue and User level allocation is complete.

Dependent Batch:

- Not Applicable.

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–30 Supervisory Statistics Batch Process(C1-SPSTM)

Parameter Name	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Average Days for Queue	Yes	Enter the average days for Queue.
Average Days for Queue Members		Enter the Days for Queue Members.

Post Execution Check / Clean Up:

- This batch should populate the Queue and Agent level statistics.
- On failure, TO DO will be created for failed records.

3.29 Case List Table Refresh (C1-CLST)

This batch refreshes the data of CI_LIST_MODE_UPDATE and and CI_CASE_ADVSRCH_UPDATE tables as part of BOD,MOD and EOD process.

Setup Prerequisites

Following is the prerequisite for the setup:

1. This batch should run at the end of BOD, MOD and EOD batches.
2. Feature Configuration C1-CLST should be set to Y.

Dependent Batch:

- Not Applicable.

Multi-threaded: No

You can specify the following parameters while executing this batch:

Table 3–31 Case List Table Refresh (C1-CLST)

Parameter Name	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.

Post Execution Check / Clean Up:

- CI_LIST_MODE_UPDATE contains same data as obtained by querying CI_LIST_MODE_UPDATE_VW.

- CI_CASE_ADVSRCH_UPDATE contains the same data as obtained by querying CI_CASE_ ADVSRH_VW.
- On failure no data will be inserted in CI_LIST_MODE_UPDATE and CI_CASE_ADVSRCH_UPDATE tables.

3.30 Account Time Barred Batch (C1-ACTBR)

This batch identifies all accounts which are 'Charged Off' that have not been sold or paid off. Based on the rule defined – it will stamp the Time Barred Date based on the state in which the account belongs and the debt category of the account.

Setup Prerequisites

Following is the prerequisite for the setup:

1. This batch should run at the end of EOD batches.

Dependent Batch:

- Derived Field Batch and Account Charge off Batch

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–32 Account Time Barred Batch (C1-ACTBR)

Parameter Name	Mandatory (Yes or No)	Description
Time Barred Entity	Yes	View name to be used as Time Barred Entity (CI_CHARGE_OFF_ VW)
Validation Date	No	Validation Date. By default - System Date. Example: Posting Date or System Date
Rule Id	No	Rule Id for Time Barred Batch
To Do Type	No	To Do Type for Time Barred Batch
Queue Code	No	Queue Code for Time Barred Batch
No. Of Days Before Time Barred	No	No. Of Days Before Time Barred for Time Barred Batch
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates

Post Execution Check / Clean Up:

- It should update time barred date which meets the rule criteria into CI_ACCT_EXTN
- On failure task should create for failed accounts.

3.31 Account Charge off Batch (C1-ACTCH)

This batch identifies all accounts that need to be fully charged off using a filter condition. Out of all the identified accounts, it then evaluates each account to decide if the account needs to be charged off manually or via STP. A rule is used for same. For STP Accounts, Host Account Charge off Service is called. For Manual Accounts, a task needs to be created in the Oracle Banking Collections Task List.

Setup Prerequisites

Following is the prerequisite for the setup:

1. This batch should run at the end of EOD batches.

Dependent Batch:

- Not applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–33 Account Charge off Batch (C1-ACTCH)

Parameter Name	Mandatory (Yes or No)	Description
Charge Off Entity	Yes	View name to be used as Charge Off Entity (CI_CHARGE_OFF_ VW).
Filter Id	Yes	Filter Id to be used in Charge Off Entity
Rule Id	Yes	Rule Id for Charge Off Batch
Task Type	Yes	Task Code on which task will be created
Queue Code Name	Yes	Queue to which task will be assigned
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates

Post Execution Check / Clean Up:

- Charge off accounts on the basis of filter criteria and rule condition met. For STP : Charge off initiated for accounts. For Manual: Task will be created for the accounts and can be view in the task list screen.
- On failure task should create for failed accounts.

3.32 Account Partial Charge off Batch (C1-PCHGO)

This batch identifies all accounts that need to be partially charged off using a rule condition.

Based on % value returned in the rule this batch performs partial chargeoff of Loan accounts in the Host.

Setup Prerequisites

Following is the prerequisite for the setup:

1. This batch should run at the end of EOD batches.

Dependent Batch:

- Not applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–34 Account Partial Charge off Batch (C1-PCHGO)

Parameter Name	Mandatory (Yes or No)	Description
Rule Id	Yes	Rule Name for Partial Chargeoff.
Account Filtering Algorithm	Yes	Algorithm Code required for filtering accounts.(Algorithm Type BATCHJOBPARM)
Task Type	Yes	Task Code on which task will be created
Queue Code Name	Yes	Queue to which task will be assigned
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch

Post Execution Check / Clean Up:

- Valid Account should be partially Chargeoff in the host(NP013 Screen)
- On failure task should create for failed accounts.

3.33 Dialer Extract Batch (C1-DIEXT)

This batch program will fetch account ID's from views satisfying the condition given in the filter. This filter needs to be defined in OBP. After fetching these records, batch program will insert it into staging area for Dialer.

Setup Prerequisites

Following is the prerequisite for the setup:

1. Lookup (C1_EXTRACT_STATUS, CHANNEL_TYPE) should be created before running this batch.

Dependent Batch:

- Not applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–35 Dialer Extract Batch (C1-DIEXT)

Parameter Name	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the

Parameter Name	Mandatory (Yes or No)	Description
		batch.
MAX-ERRORS	No	Enter a value here to override the maximum number of errors allowed before run terminates.
VIEW-NAME	Yes	View to Fetch acct_id based on extract conditions
OVERRIDE-FLAG	No	Optional. Expected values Y/N. Default value is N. The flag is used to control insert/updates in staging table.
EOD-FLAG	Yes	Values (Y/N). If this flag is 'Y' then records in the staging area should be deleted, else if flag is 'N' then records should not be deleted.
EXTRACT-TYPE	Yes	If provided then all set of conditions (filters) specified in EXTRACT-CONDITIONS will be executed in order of priority defined in the batch parameter.
EXTRACT-CONDITIONS	Yes	Used to provide filter condition.

Post Execution Check / Clean Up:

- On successful completion of this batch, records will be inserted in the CI_DIALER_ EXTRACTS table.
- On failure, TO DO will be created for failed accounts.

3.34 Account Dialer Extract Batch (C1-DLEXT)

This batch identifies all accounts that need to be sent to the Dialer using a filter condition. The accounts with cases on hold and with suspension Do Not Include in Dialer File are automatically excluded.

Setup Prerequisites

Following is the prerequisite for the setup:

1. This batch should run at the end of EOD batches.

Dependent Batch:

- Not applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–36 Account Dialer Extract Batch (C1-DLEXT)

Parameter Name	Mandatory (Yes or No)	Description
Dialer Extract View Name	Yes	Dialer Extract View Name(CI_DIALER_EXTRACT_VW)
Extraction Type	Yes	Enter value as Dialer or IVR

Parameter Name	Mandatory (Yes or No)	Description
Extract Condition Filters	Yes	Extract Condition Filters
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates

Post Execution Check / Clean Up:

- Data pushed to below tables as per the filter conditions. CI_ACCT_DILR_EXTRCT and CI_DIALER_EXTRACTS_CONTACT
- On failure task should create for failed accounts.

3.35 Case Lock Unlock Monitor – Task Creation(C1-CSTD)

While user is working on a case system locks the case in context for other users to work on. There can be scenarios where the user who locks the case did not release the case for other users to make an update. Like system crash, closure of browser or intentional delay.

In these scenarios this batch will create a Task for the administrator who can release the locked cases via Supervisory Functions

Setup Prerequisites

Proper mapping to Task Type and Queue and user as provided in batch parameter.

Dependent Batch:

- Not applicable

Multi-threaded: Yes

You can specify the following parameters while executing this batch:

Table 3–37 Case Lock Unlock Monitor – Task Creation (C1-CSTD)

Parameter Name	Mandatory (Yes or No)	Description
Locked by User	No	Batch can create tasks for cases locked by specific user
Time in Minutes	Yes	Batch will check for cases where lock time is more than specified time (in minutes)
Case Type	No	Batch can create ToDos for cases of specific Case type
Queue Code	Yes	Queue to which task should get assigned to
User to Assign the Task	No	User to which task should get assigned to
Task Type	Yes	Task will get created of this task type

Parameter Name	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.

Post Execution Check / Clean Up:

On successful completion of this batch, task for supervisor of a user who has the cases which are locked for more than the stipulated amount of time will be created.

3.36 Account Setup in Recovery Batch (C1-ACTSU)

This batch identifies all accounts which are charged off on the previous day and moved into the Recovery system.

Setup Prerequisites

Following is the prerequisite for the setup:

- Update Entity Batch Run.

Dependent Batch

- Not applicable

Multi-threaded Yes

You can specify the following parameters while executing this batch:

Table 3–38 Account Setup in Recovery Batch (C1-ACTSU)

Parameter Name	Mandatory (Yes or No)	Description
Case Category	No	Case Categories (Comma Separated) for which review date not to be set
Dispute Warning Indicator	No	Dispute Warning Indicators (Comma Separated) for which Indicator to be set on account
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.

Post Execution Check / Clean Up

On successful completion of this batch the accounts which are charged off on the previous day should be created in recovery and would be visible in Recovery landing page. Pricing Plan and Interest Plan (if applicable) should be created on such accounts.

3.37 Write off Monitor Batch (C1-WRTOF)

This batch identifies all accounts which are marked for write-off and closes them in Recovery system. The account status is set as Closed-Written Off . If the account has an active settlement offer, the account status is set as Closed – Settled & Written Off.

Setup Prerequisites

Following is the prerequisite for the setup:

- Mark For Write Off Flag should be set to Y for the account to be processed.

Dependent Batch

- Not applicable

Multi-threaded Yes

You can specify the following parameters while executing this batch:

Table 3–39 Write Off Monitor Batch-Recovery (C1-WRTOF)

Parameter Name	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.

Post Execution Check / Clean Up

On successful completion of this batch, accounts status is marked as Closed-Written off. Pricing and Interest Plan are also closed for such accounts.

3.38 Apply Debt Sale Payments Batch (C1-DBTSL)

This batch identifies all accounts which are marked for debt sale and closes them in Recovery system. The account status is set as Closed-Debt Sold.

Additionally, it applies the payment on such accounts with the amount received by the vendor before closing them.

Setup Prerequisites

Following is the prerequisite for the setup:

- Payment information of debt sold account should be present in CI_DEBT_SALE_INSTRUCTIONS table.

Dependent Batch

- Not applicable

Multi-threaded Yes

You can specify the following parameters while executing this batch:

Table 3–40 Apply Debt Sale Payments Batch-Recovery (C1-DBTSL)

Parameter Name	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.

Post Execution Check / Clean Up

On successful completion of this batch, accounts status is marked as Closed-Debt Sold.

Successful payment should be applied on the account which can be verified on the Recovery landing page.

3.39 Account Closure Monitor Batch (C1-PAYOF)

This batch identifies all accounts for which all the outstanding amount is paid or is below threshold and closes them in Recovery system. The account status is set as Closed-PayOff.

Setup Prerequisites

Following is the prerequisite for the setup:

- Condition builder must be created to filter the account to be finalized.

Dependent Batch

- Not applicable

Multi-threaded Yes

You can specify the following parameters while executing this batch:

Table 3–41 Account Closure Monitor Batch- Recovery (C1-PAYOF)

Parameter Name	Mandatory (Yes or No)	Description
Filter Name	Yes	Used to specify the filter ID to fetch the accounts that need to be Closed.
Promise To Pay Cancellation Reason	Yes	Used to specify the cancellation reason to cancel PTP.
Pay Off Monitor View	Yes	Used to view the name of Pay off monitor.
Pay Off Entity Algorithm Code	No	Used to specify the algorithm to close the contract if there are no open cases on an account.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

On successful completion of this batch, accounts are marked as Closed-PayOff. Contract will be stopped, case will be moved to final status and active PTP is cancelled.

Verification can be done from CI_CASE, CI_PTP and CI_ACCT_EXTN tables.

3.40 Vendor Communication Outbound Interface Batch (C1-VNDCM)

This batch is used for creating files, which can be handed over to the vendors.

Setup Prerequisites

Following is the prerequisite for the setup:

- Proper File extraction algorithm to be provided by implementation team which will be attached on Vendor Service Type Screen.

Dependent Batch

- Not applicable

Multi-threaded Yes

You can specify the following parameters while executing this batch:

Table 3–42 Vendor Communication Outbound Interface Batch- Recovery (C1-VNDCM)

Parameter Name	Mandatory (Yes or No)	Description
Vendor Ids	No	Enter Vendor ID’s if specific Vendor's data need to be processed.
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

On successful completion of this batch, Extract files created for each vendor in staging area.

3.41 Monthly Interest Computation Batch (C1-INTCP)

This batch is run on month end and it calculates recovery interest on the accounts.

Setup Prerequisites

Following is the prerequisite for the setup:

- Interest Rule Setup in Define Recovery Bank Policy Screen

Dependent Batch

- Not applicable

Multi-threaded Yes

You can specify the following parameters while executing this batch:

Table 3–43 Monthly Interest Computation Batch- Recovery (C1-INTCP)

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

On successful completion of this batch, recovery interest is computed on the account. It can be verified in Financial Logs section on the Recovery screen.

3.42 Monthly Account Statement Batch (C1-RSTMT)

This batch is run on month end and logs the financial data of Recovery accounts in the statement table.

Setup Prerequisites

Following is the prerequisite for the setup:

- Not applicable

Dependent Batch

- Not applicable

Multi-threaded Yes

You can specify the following parameters while executing this batch:

Table 3–44 Monthly Account Statement Batch- Recovery (C1-RSTMT)

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

On successful completion of this batch, all the financial data of accounts is logged in table CI_REC_ACCT_ STMT_SMRY.

3.43 Recovery Party Warning Indicator Update Batch (C1-WIUPD)

This batch applies the impact of Party Level Risk Indicator on account when a risk indicator is added or removed from the recovery account's party.

Setup Prerequisites

Following is the prerequisite for the setup:

- Update Entity Batch

Dependent Batch

- Not applicable

Multi-threaded Yes

You can specify the following parameters while executing this batch:

Table 3–45 Recovery Party warning indicator update Batch (C1-WIUPD)

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check / Clean Up

On successful completion of this batch, risk indicator impacts are applied on the account.

3.44 Settlement Offer Creation Batch (C1-STLOF)

This batch creates the Settlement Offer on the applicable Recovery Accounts.

Setup Prerequisites

Following is the prerequisite for the setup:

- Strategy Monitor Batch

Dependent Batch

- Not applicable

Multi-threaded Yes

You can specify the following parameters while executing this batch:

Table 3–46 Settlement Offer Creation Batch (C1-STLOF)

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Enter a value here to override the maximum number of errors allowed before run terminates.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
Rule Id or Ruleset Id	Yes	Rule id or Ruleset Id configured for Settlement Offer creation process.

Parameter Name	Mandatory (Yes or No)	Description
Filter Id	Yes	Filter Id configured for excluding accounts on which settlement offers are to be created.
View Name	Yes	Settlement Offer Input view. Product Shipped view CI_SETTLEMENT_OFFER_VW
isRuleSet flag	Yes	Rule Set Flag Y/N

Post Execution Check / Clean Up

On successful completion of this batch, Settlement Offer will get created on applicable recovery accounts.

4 Technical Recommendations

This section recommends parameter values to achieve maximum performance. This will vary with different hardware set. The actual performance would depend on the number of CPUs and RAM available on the application server, and many other hardware parameters.

The following recommendations must be treated as guidelines and not as the actual values.

To improve the overall batch performance of 64-bit application server on Linux 64-bit environment, we recommend you to make changes in the following files:

Table 4–1 Recommendations for Files to be Modified

File Name	Change From	Change To
hibernate.properties	hibernate.ucp.inactive_ connection_timeout=300	hibernate.ucp.inactive_connection_ timeout=600
threadpoolworker.sh	MEM_ARGS="-Xms512m -Xmx1024m -XX:MaxPermSize=768m "	MEM_ARGS="-Xms8192 -Xmx8192m -XX:MaxPermSize=768m "

Note

These settings are not applicable for 32-bit application servers on Windows, AIX, and Linux platforms.

Recommended Thread Count for a Batch:

Thread count for each batch that supports multi-threading should be as per the number of available CPU and volume of data being handled. More details are available in benchmarking document.

Common Input for All Batches:

E-mail Address: This will be the E-mail ID of the person to whom notification is to be sent in case of TO-DO created by batches.

User ID: This will be a valid user if in collections and will be used as user id executing the batch.

This user ID should also be mapped to OFFLINE_CHANNEL_USER_NAME property in /config/properties/CredentialPropertyStore.properties file.