Oracle Financial Services Revenue Management and Billing

OR

Oracle Insurance Revenue Management and Billing

Version 8.1.0.0.0

Batch Guide

Revision 28.0

G53179-01

March 2026

ORACLE®

Oracle Financial Services Revenue Management and Billing/Oracle Insurance Revenue Management and Billing Version 8.1.0.0.0 Batch Guide

> **Note:** To improve the content readability, the above two products are collectively referred to as Oracle Revenue Management and Billing throughout this document.

G53179-01

Copyright Notice

Copyright © 2026, Oracle and/or its affiliates.

License Restrictions

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 de-compilation of this software, unless required by law for interoperability, is prohibited.

Warranty Disclaimer

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.

Restricted Rights Notice

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.

Hazardous Applications Notice

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.

Trademark Notice

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.

Third-Party Content, Products, and Services Disclaimer

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.

iii

Preface

About This Document

This document provides detail information about various batches in Oracle Revenue Management and Billing.

Intended Audience

This document is intended for the following audience:

- End-users
- Implementation Team
- Consulting Team
- Development Team

Organization of the Document

The information in this document is organized into the following sections:

Section No.	Section Name	Description
Section 1	Accruals	Lists and describes the batches used for the **Accruals** feature.
Section 2	Billing	Lists and describes the batches used for the **Billing** feature.
Section 3	Contracts	Lists and describes the batches used for the **Contracts** feature.
Section 4	Deal Pricing Management	Lists and describes the batches used for the **Deal Pricing Management** feature.
Section 5	Deferred Revenue Recognition	Lists and describes the batches used for the **Deferred Revenue Recognition** feature.
Section 6	Earnings Credit Rate	Lists and describes the batches used for the **Earnings Credit Rate** feature.
Section 7	Entity Audit	Lists and describes the batches used for the **Entity Audit** feature.
Section 8	Financial Transactions	Lists and describes the batches used for the **Financial Transactions** feature.
Section 9	FOP Reports	Lists and describes the batches used for the **FOP Reports** feature.
Section 10	Fully-Insured Billing	Lists and describes the batches used for the **Fully-Insured Billing** feature.

Section No.	Section Name	Description
Section 11	Funding Request	Lists and describes the batches used for the **Funding Request** feature.
Section 12	Health Insurance Policy	Lists and describes the batches used for the **Health Insurance Policy** feature.
Section 13	Hold Request	Lists and describes the batches used for the **Hold Request** feature.
Section 14	Inbound Message	Lists and describes the batches used for the **Inbound Message** feature.
Section 15	Offset Request	Lists and describes the batches used for the **Offset Request** feature.
Section 16	Payment Agreement Request	Lists and describes the batches used for the **Payment Agreement Request** feature.
Section 17	Payments	Lists and describes the batches used for the **Payments** feature.
Section 18	Pricing Management	Lists and describes the batches used for the **Pricing Management** feature.
Section 19	Reconciliation	Lists and describes the batches used for the **Reconciliation** feature.
Section 20	Repricing	Lists and describes the batches used for the **Repricing** feature.
Section 21	Request	Lists and describes the batches used for the **Request** feature.
Section 22	Security	Lists and describes the batches used for the **Security** feature.
Section 23	Statement	Lists and describes the batches used for the **Statement** feature.
Section 24	Stop-Loss Insurance	Lists and describes the batches used for the **Stop-Loss Insurance** feature.
Section 25	Upload Request	Lists and describes the batches used for the **Upload Request** feature.
Section 26	Data Extractor	Lists and describes the batches used for the **Data Extractor** feature.
Section 27	Generic Extract from Datasets	Lists and describes the batches used for the **Generic Extract from Datasets** feature.

Conventions

The following conventions are used across the document:

Convention	Meaning
boldface	Boldface indicates graphical user interface elements associated with an action, or terms defined in the text.
italic	Italic indicates a document or book title.
monospace	Monospace indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or information that an end-user needs to enter in the application.

Related Documents

You can refer to the following documents for more information:

Document Name	Description
Oracle Revenue Management and Billing Banking User Guide	Describes various features which are available for the financial services business. For example, customer registration, customer 360° view, invoicing group, pricing management, multi-currency accounts, currency conversion, construct based billing and settlement, trial billing, product lifecycle management, subscription billing, mass pricing update, accrual, foreign exchange gain loss, transaction feed management, upload validated payment and adjustment data, freeze payments on notification, payment request, offset request, funding request, hold request, refund/write off request, dispute request, upload request, earnings credit rate, payment agreement request, invoice request, deal management, etc. It describes all screens related to these features and explains how to perform various tasks related to the feature in the application.

Document Name	Description
Oracle Revenue Management and Billing Insurance User Guide	Describes various features which are available for the following three lines of health insurance business – fully-insured group, self-funded, and individual. For example, customer registration, customer 360° view, pricing management, trial billing, transaction feed management, upload validated payment and adjustment data, freeze payments on notification, payment request, offset request, funding request, hold request, refund/write off request, upload request, payment agreement request, invoice request, inbound message processing, reconciliation, entity audit, premium repricing, etc.

It describes all screens related to these features and explains how to perform various tasks related to the feature in the application. |

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Contents

1. Accruals

This section provides information about the following batches:

- [Accrual Calculation (C1_ACCAL)](#)
- [Accrual Creation (C1_ACCRE)](#)
- [Accrual Reversal (C1_ACREV)](#)
- [Create To Do for Account without Accrual Cycle (C1_ACELG)](#)

1.1 Accrual Calculation (C1_ACCAL)

The **Accrual Calculation (C1_ACCAL)** batch is used to calculate the accrual amount for accounts. It considers the batch business date as the accrual date and finds the accrual cycle within which the accrual date is present. Once the accrual cycle is identified, the batch finds all accounts where the accrual cycle is defined. It also identifies the accrual type which is effective for the division (to which the account belongs) on the batch business date. Based on the options defined in the accrual type, the system accordingly calculates the accrual amount for the accounts.

This batch executes the algorithms attached at the following spots in the accrual type in the specified order:

1. Accrual Eligibility
2. Accrual Pre Processing
3. Accrual Calculation
4. Accrual Post Calculation

If an algorithm created using the **C1_ACCR_ELG** algorithm type is attached to the **Accrual Eligibility** algorithm spot in the accrual type, the system checks whether the account is a member of an invoicing group or whether the account is included in an active construct. If the account is a member of an invoicing group or included in an active construct, the account is not considered for accrual calculation.

> **Note:** The system checks whether the account is a member of an invoicing group only when the **includeIGAEligibility** parameter in an algorithm created using the **C1_ACCR_ELG** algorithm type is set to **Y**.

If an algorithm created using the **C1_ACCR_POCA** algorithm type is attached to the **Accrual Post Calculation** algorithm spot in the accrual type, the status of the accrual is changed to **Approval in Progress** once the accrual is calculated for the account. However, if there is no algorithm attached to the **Accrual Post Calculation** algorithm spot in the accrual type, the status of the accrual is changed to **Ready For Accrual** once the accrual is calculated for the account.

> **Note:** This batch caches division related information, such as work calendar, currency conversion algorithms, rounding type, and accrual types, which is required for accrual calculation. If any of these attributes change for a division, you must execute the **Flush All Caches (F1-Flush)** batch before executing the **Accrual Calculation (C1_ACCAL)** batch. The **Flush All Caches (F1-Flush)** batch is used to clean the application cache.

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Division	Yes	Used when you want to calculate accruals for accounts belonging to a particular division. **Note:** You must specify a division which is already defined in the system.
Accrual Cycle	No	Used when you want to calculate accruals for accounts having a particular accrual cycle. **Note:** You must specify an accrual cycle which is already defined in the system.
Delete Manual Accrual	No	Used when you want to delete the accrual which is manually created for the account on the accrual date. The valid values are: • Y • N **Points to Note:** If you do not specify any value, by default, the parameter value is set to **N**. If you set the parameter value to **N**, the batch will throw an error when the accrual in the **Ready For Accrual**, **Processed**, or **Approval in Progress** status already exists for the account on the accrual date.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

> **Note:** If the **Accrual Calculation (C1_ACCAL)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the accrual amount is calculated for the accounts that meet the search criteria. The status of the accrual is changed to either **Approval in Progress** or **Ready For Accrual** depending on whether the accrual post calculation algorithm is attached on the accrual type.

1.2   Accrual Creation (C1_ACCRE)

The **Accrual Creation (C1_ACCRE)** batch is used to check whether there are any accruals in the **Ready for Accrual** or **Processed** status. It considers only those accruals whose accrual date is earlier than or equal to the batch business date. If there is an accrual in the **Ready for Accrual** or **Processed** status, the accrual object (i.e. adjustment) is created for the account. Based on the adjustment type defined in the accrual type, the system accordingly creates the accrual adjustment for the account. Once the accrual adjustment is created for the account, the status of the accrual is changed to **Accrued**.

This batch executes the algorithms attached at the following spots in the accrual type in the specified order:

1. Accrual Creation
2. Accrual Post Processing

This batch is a multi-threaded batch. The multi-threading is based on accrual ID and chunks for multi-threading are created based on numerical distribution of accrual ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Division	Yes	Used when you want to create accrual adjustment for accounts belonging to a particular division. **Note:** You must specify a division which is already defined in the system.
Accrual Cycle	No	Used when you want to create accrual adjustment for accounts having a particular accrual cycle. **Note:** You must specify an accrual cycle which is already defined in the system.
Validation Required	No	Used to indicate whether the system should check if there are any accruals (whose accrual date is earlier than or equal to the batch business date) in the **Approval in Progress** status. The valid values are: • Y • N

Parameter Name	Mandatory (Yes or No)	Description
		Note: If you set the parameter value to **Y**, the batch will end in error when there are accruals (whose accrual date are earlier than or equal to the batch business date) in the **Approval in Progress** status. However, if you set the parameter value to **N**, the batch will skip the accruals (if any) in the **Approval in Progress** status and creates accrual adjustment for accruals in the **Ready For Accrual** or **Processed** status.
Characteristics Type for Distribution Code Level Adjustments	No	Used to specify the characteristic type that stores the adjustment type. If you specify the characteristic type, the system will check whether the specified characteristic type is defined for the distribution code. If so, the accrual adjustment for the distribution code is created using the respective adjustment type. If the characteristic type is not specified in this parameter or if the specified characteristic type is not defined for the distribution code, the system creates the accrual adjustment for the distribution code using the accrual adjustment type that is specified in the accrual type.
		Note: You must specify a characteristic type where the characteristic entity is set to **Distribution Code**.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Accrual Creation (C1_ACCRE)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the accrual adjustment is created for accounts and the status of the accrual is changed to **Accrued**.

1.3   Accrual Reversal (C1_ACREV)

The **Accrual Reversal (C1_ACREV)** batch is used to reverse the accrual for the accounts. It only considers the accruals where the batch business date is between the reversal start and end dates. On reversing the accrual, the accrual adjustments are cancelled and the status of the accrual is changed to **Reversed**.

This batch executes the algorithms attached at the following spots in the accrual type in the specified order:

1. Accrual Reversal Pre Processing
2. Accrual Reversal
3. Accrual Reversal Post Processing

This batch is a multi-threaded batch. The multi-threading is based on accrual ID and chunks for multi-threading are created based on numerical distribution of accrual ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Division	Yes	Used when you want to reverse accrual for accounts belonging to a particular division. **Note:** You must specify a division which is already defined in the system.
Accrual Cycle	No	Used when you want to reverse accrual for accounts having a particular accrual cycle. **Note:** You must specify an accrual cycle which is already defined in the system.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Accrual Reversal (C1_ACREV)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the accrual is reversed for the accounts and the accrual adjustments are cancelled. In addition, the status of the accrual is changed to **Reversed**.

1.4 Create To Do for Account without Accrual Cycle (C1_ACELG)

The **Create To Do for Account without Accrual Cycle (C1_ACELG)** batch is used to create a To Do for all such accounts which are eligible for accrual, but the accrual cycle is not defined either at the account or division level.

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Division	Yes	Used when you want to create To Do for accounts belonging to a particular division.
		Note: You must specify a division which is already defined in the system.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Create To Do for Account without Accrual Cycle (C1_ACELG)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, a To Do is created for an account which is eligible for accrual, but the accrual cycle is not defined either at the account or division level.

2.   Billing

Oracle Revenue Management and Billing enables you to carry out billing based on the bill cycle (defined for the account). In other words, you are able to generate a bill for a bill cycle using the billable charges whose start or end date is earlier than or equal to the cut-off date. You can generate bills online or through a batch process. If you generate bills through a batch process, the batch business date is treated as the cut-off date.

Oracle Revenue Management and Billing provides you with a batch named **BILLING**, which generates the bills based on the bill cycle defined for the account. In the **BILLING** batch, multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. This chunking logic might not be efficient while handling voluminous data. To handle high volume with better performance, you must generate bills by executing the following three batches of either of sets in the specified order:

Set A

1.   Pending Bill Generation (C1-PNDBL)
2.   Bill Segment Generation (C1-BLGEN)
3.   Bill Completion (C1-BLPPR)

Set B

1.   Pending Bill Generation (BILLOPEN)
2.   Bill Segment Generation (BSGENREG)
3.   Bill Completion (POSTPROC)

> **Points to Note:**
>
> These three batches are primarily designed and developed for the Banking and Payments industries.
>
> You should execute all three batches in the above mentioned sequence to complete the billing process for a set of accounts. Ideally, you should not change any setup data (for example, bill cycle of the account) within the billing process (i.e. in between the above three batch execution). Also, you must specify the same parameters (for example, division, bill cycle, etc.) including the batch business date in the **C1-PNDBL**, **C1-BLGEN** and **C1-BLPPR** batches during the billing process. Otherwise, erroneous results might occur.

Before you execute these billing batches (i.e. **BILLING**, **C1-PNDBL**, **C1-BLGEN**, **C1-BLPPR, BILLOPEN**, **BSGENREG**, and **POSTPROC**), you must do the following:

- Set the **Bill Segment Freeze Option** field in the **Billing** tab of the **Installation Options** screen to either **Freeze At Will** or **Freeze At Bill Completion**.

- Set the **Billable Charges Date** option type in the **C1_FINTRANOP** feature configuration to either **S** or **E** depending on whether you want to consider the billable charges whose start or end date is earlier than or equal to the cut-off date.

> **Note:**
>
> If the **Billable Charges Date** option type of the **C1_FINTRANOP** feature configuration is set to **E**, the system considers the following:
>
> \>> Recurring billable charges whose start date is earlier than the cut-off date and end date is later than the cut-off date
>
> \>> Non-recurring billable charges whose end date is earlier than or equal to the cut-off date

This section provides information about the following batches:

- [Billing (BILLING)](#)
- [Pending Bill Generation (C1-PNDBL)](#)
- [Bill Segment Generation (C1-BLGEN)](#)
- [Bill Completion (C1-BLPPR)](#)
- [Pending Bill Generation (BILLOPEN)](#)
- [Bill Segment Generation (BSGENREG)](#)
- [Bill Completion (POSTPROC)](#)
- [Refresh Pricing (C1-TXNRP)](#)
- [Upload and Validate Usage Data File (C1-ODFU)](#)
- [Billable Charge Creation (C1-ODBCH)](#)
- [Adhoc Billing (C1-FABL)](#)
- [Freeze and Complete Adhoc Bills (C1-FCADH)](#)
- [Pending Bill Segments Deletion (C1-BSEGD)](#)
- [Pending Bill Deletion (C1-PNBD)](#)
- [Assign Sequential Bill Numbers (ASSGNSBN)](#)
- [Generate Credit or Correction Note for Bill (C1-BLCRN)](#)
- [Bill Cancellation (C1-BILCN)](#)
- [Bill Deletion (C1BILDEL)](#)

2.1 Billing (BILLING)

The **Billing (BILLING)** batch is used to generate bills based on the bill cycle defined for the account. Based on the batch business date, the system identifies the bill cycles for which the bills must be generated. Once the bill cycles are identified, the system identifies a set of accounts on which these bill cycles are defined. Then, based on a set of parameters specified while executing this batch, the bills are generated for accounts that meet the criteria and the status of the bills is set to **Pending**.

Then, based on the batch business date, the system identifies the recurring and non-recurring billable charges for which bill segments must be generated. Once the billable charges are identified, the bill segments are generated in the pending bills. In addition, the post-processing bill segments, if any, are generated in the pending bills.

> **Note:**
>
> If the **Billable Charges Date** option type of the **C1_FINTRANOP** feature configuration is set to **E**, the system considers the following:
>
> >> Recurring billable charges whose start date is earlier than the cut-off date and end date is later than the cut-off date
>
> >> Non-recurring billable charges whose end date is earlier than or equal to the cut-off date

If the **Freeze and Complete** check box is selected for the bill cycle for which the bill is generated, the status of the bill segments is set to **Frozen** and the status of the bill is set to **Complete**. But, if the **Freeze and Complete** check box is not selected for the bill cycle for which the bill is generated, the status of the bill segments is set to **Freezable** and the status of the bill remains as **Pending**.

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Batch Business Date	No	Used to identify the following: • Bill cycles within which the specified date falls • Billable charges for which bill segments must be generated **Points to Note:** All billable charges whose start or end date is earlier than or equal to the specified date are considered during bill segment generation. If you do not specify any date, the batch business date is set to the current date.
Thread Count	No	Used to specify the number of threads you want to spawn in parallel. **Note:** By default, the thread count is set to 0.
Division	No	Used when you want to generate bills for accounts belonging to a particular division. **Note:** You must specify a division which is already defined in the system.

Parameter Name	Mandatory (Yes or No)	Description
Bill Cycle	No	Used when you want to generate bills for accounts having a particular bill cycle. **Points to Note:** You must specify a bill cycle which is already defined in the system. You can specify maximum twenty comma-separated values for this parameter.
Account ID	No	Used when you want to generate bills for a particular account.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Billing (BILLING)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the bills are generated for accounts that meet the criteria.

2.2 Pending Bill Generation (C1-PNDBL)

The **Pending Bill Generation (C1-PNDBL)** batch is used to generate pending trial or actual bills for accounts that meet the criteria.

Based on the batch business date, the system identifies the bill cycles for which the pending trial or actual bills must be generated. Once the bill cycles are identified, the system identifies a set of accounts on which these bill cycles are defined. Then, based on a set of parameters specified while executing this batch, the pending trial or actual bills are generated for accounts that meet the criteria.

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Batch Business Date	No	Used to identify the bill cycles within which the specified date falls. **Note:** If you do not specify any date, the batch business date is set to the current date.
Thread Count	No	Used to specify the number of threads you want to spawn in parallel. **Note:** By default, the thread count is set to 0.
Division	No	Used when you want to generate pending trial or actual bills for accounts belonging to a particular division. **Note:** You must specify a division which is already defined in the system.
Bill Cycle	No	Used when you want to generate pending trial or actual bills for accounts having a particular bill cycle. **Points to Note:** You must specify a bill cycle which is already defined in the system. You can specify maximum twenty comma-separated values for this parameter.

Parameter Name	Mandatory (Yes or No)	Description
Delete Existing Bill	No	Used to indicate whether you want to delete existing actual bills which are in the **Pending** status. The valid value is: • Y **Points to Note:** If you do not want to delete the existing pending bill, you must leave this parameter blank. In such case, new pending bill is not generated, and the existing pending bill (available for the account and bill cycle combination) is used for further processing. The system does not delete the existing bills which are in the **Complete** status. If the existing pending bills have any bill segments in the **Freezable** or **Error** status, the bills are deleted. But, if the existing pending bills have any bill segments in the **Frozen** or **Cancelled** status, the bills are not deleted. This parameter is not supported when you are generating trial bills.
Bill Generation Type	No	Used to indicate whether you want to generate pending trial or actual bills. If you want to generate pending trial bills, the valid value is **Trial**. And, if you want to generate pending actual bills, you must leave this parameter blank.
Description for Trial Billing Batch Run	No	Used to specify the description for the batch run. This description is stamped on each trial bill which is generated in the batch run. This helps to identify the trial bills which are generated in a particular batch run. **Points to Note:** This parameter is used only when you are generating pending trial bills. If you do not specify the description, by default, the description is set to "Trial Invoicing Run On YYYY-MM-DD-HH.MM.SS".

Parameter Name	Mandatory (Yes or No)	Description
Process All or Selected Accounts	No	Used to indicate whether you want to generate pending trial bills for all accounts (that meet the criteria) or only for those accounts (that meet the criteria and) where the **Trial Bill** check box is selected. If you want to generate pending trial bills for only those accounts where the **Trial Bill** check box is selected, set the value of this parameter to **Y**. If you want to generate pending trial bills for all accounts irrespective of whether the **Trial Bill** check box is selected or not, you must leave this parameter blank. **Note:** This parameter is used either when you are creating trial bills or when you are creating actual bills using trial bills.
Account ID	No	Used when you want to generate pending trial or actual bills for a particular account.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

If the **Pending Bill Generation (C1-PNDBL)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

If the status of the **Pending Bill Generation (C1-PNDBL)** batch run is set to **Error**, you need to first resolve the error and then execute the batch once again.

Post Execution Check/Clean Up:

On successful completion of this batch, the pending trial or actual bills are generated for the accounts that meet the criteria.

2.3   Bill Segment Generation (C1-BLGEN)

The **Bill Segment Generation (C1-BLGEN)** batch is used to generate bill segments in the pending trial or actual bills. It is also used to copy trial bills' bill segments to the actual bills while creating actual bills using trial bills.

Based on the batch business date, the system identifies the recurring and non-recurring billable charges for which bill segments must be generated. Once the billable charges are identified, the bill segments are generated in the pending trial or actual bills.

Note:

If the **Billable Charges Date** option type of the **C1_FINTRANOP** feature configuration is set to **E**, the system considers the following:

>> Recurring billable charges whose start date is earlier than the cut-off date and end date is later than the cut-off date

>> Non-recurring billable charges whose end date is earlier than or equal to the cut-off date

---

If the **Freeze and Complete** check box is selected for the bill cycle for which the pending actual bill is generated, the status of the bill segments is set to **Frozen**. But, if the **Freeze and Complete** check box is not selected for the bill cycle for which the pending actual bill is generated, the status of the bill segments is set to **Freezable**. However, in case of pending trial bills, irrespective of whether the **Freeze and Complete** check box is selected or not for the bill cycle, the status of the bill segments is set to **Freezable**.

While creating actual bills using trial bills, the system copies trial bills' bill segments which are in the **Freezable** status to the actual bills. If you want to freeze the bill segments in such actual bills, you need to execute the **Bill Segment Generation (C1-BLGEN)** batch once again without specifying the trial billing batch run number.

Points to Note:

There might be situations when you have received certain billing information after the bill and bill segments are generated. In such scenarios, you must first execute the **Pending Bill Generation (C1-PNDBL)** batch to fetch the newly created billable charges and then execute the **Bill Segment Generation (C1-BLGEN)** batch. While executing the **Pending Bill Generation (C1-PNDBL)** batch, you must leave the **Delete Existing Bill** parameter blank indicating that you want to use the existing pending bill for further processing.

In addition, there might be situations when you want to regenerate the bill segments before freezing them. In such case, you need to set the **Bill Segment Regeneration on Freeze** option type of the **C1_FINTRANOP** feature configuration to **Y**, and then execute the **Pending Bill Generation (C1-PNDBL)** and **Bill Segment Generation (C1-BLGEN)** batches once again. While executing the **Pending Bill Generation (C1-PNDBL)** batch, you must leave the **Delete Existing Bill** parameter blank indicating that you want to use the existing pending bill for further processing. If the **Freeze and Complete** check box is selected for the bill cycle and the **Bill Segment Regeneration on Freeze** option type is set to **Y**, then the bill segments are regenerated and the status of the bill segments is set to **Frozen**. Note that if the **Freeze and Complete** check box is not selected for the bill cycle, then the bill segments are always regenerated irrespective of whether the **Bill Segment Regeneration on Freeze** option type is set to **Y** or not.

This batch is a multi-threaded batch. The multi-threading is based on billable charge ID and chunks for multi-threading are created based on numerical distribution of billable charge ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Batch Business Date	No	Used to identify the billable charges for which bill segments must be generated.
		Points to Note:
		All billable charges whose start or end date is earlier than or equal to the specified date are considered during bill segment generation.
		If you do not specify any date, the batch business date is set to the current date.
Thread Count	No	Used to specify the number of threads you want to spawn in parallel.
		Note: By default, the thread count is set to 0.
Division	No	Used when you want to generate bill segments for accounts belonging to a particular division.
		Note: You must specify a division which is already defined in the system.
Bill Cycle	No	Used when you want to generate bill segments for accounts having a particular bill cycle.
		Points to Note:
		You must specify a bill cycle which is already defined in the system.
		You can specify maximum twenty comma-separated values for this parameter.

Parameter Name	Mandatory (Yes or No)	Description
Billing Processing Sequence	No	Used to specify the billing processing sequence of the contract's contract type whose bill segments you want to generate. This parameter is useful when you want to generate bill segments in a particular order of importance.
		Note: You can specify comma-separated values for this parameter. For example, 10, 20, 30. In such case, the bill segments of contracts whose contract types' billing processing sequence is 10, 20, or 30 will be generated simultaneously.
Bill Generation Type	No	Used to indicate whether you want to generate bill segments for pending trial or actual bills. If you want to generate bill segments for pending trial bills, the valid value is **Trial**. And, if you want to generate bill segments for pending actual bills or if you want to copy trial bills' bill segments while creating actual bills using trial bills, you must leave this parameter blank.
Trial Billing Batch Run Number	No	Used to indicate the trial billing batch run number whose trial bills' bill segments you want to copy to the actual bills.
		Note: This parameter is used only when you are creating actual bills using trial bills.

Parameter Name	Mandatory (Yes or No)	Description
Process All or Selected Accounts	No	Used to indicate whether you want to generate bill segments for all accounts (that meet the criteria) or only for those accounts (that meet the criteria and) where the **Trial Bill** check box is selected. If you want to generate bill segments for only those accounts where the **Trial Bill** check box is selected, set the value of this parameter to **Y**. If you want to generate bill segments for all accounts irrespective of whether the **Trial Bill** check box is selected or not, you must leave this parameter blank.
		Points to Note: This parameter is used either when you are creating trial bills or when you are creating actual bills using trial bills. If you set the value of this parameter to **Y** while creating actual bills using trial bills, the system will copy trial bills' bill segments of only those accounts (that meet the criteria and) where the **Trial Bill** check box is selected. And, if you leave this parameter blank, the system will copy trial bills' bill segments of all accounts (that meet the criteria) irrespective of whether the **Trial Bill** check box is selected or not. If the bill segments do not exist for an account while creating actual bills using trial bills, the system generates the bill segments for the account.
Account ID	No	Used when you want to generate bill segments for a particular account.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated.
		Note: You must specify a value greater than or equal to zero.

Parameter Name	Mandatory (Yes or No)	Description
Freeze and Complete	No	Used to override the **Freeze and Complete** option defined for the bill cycle. The valid values are: • **Y** – Used when you want to freeze and complete the bill segments irrespective of whether the **Freeze and Complete** option is selected or not for the bill cycle. • **N** – Used when you do not want to freeze and complete the bill segments irrespective of whether the **Freeze and Complete** option is selected or not for the bill cycle. **Note:** If you do not want to override the **Freeze and Complete** option defined for the bill cycle, you must leave this field blank.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

You must specify the same parameters (for example, division, bill cycle, etc.) including the batch business date in the **C1-PNDBL**, **C1-BLGEN** and **C1-BLPPR** batches during the billing process. Otherwise, erroneous results will occur.

Before executing the **Bill Segment Generation (C1-BLGEN)** batch, you need to ensure that the **Pending Bill Generation (C1-PNDBL)** batch is successfully executed.

If the **Bill Segment Generation (C1-BLGEN)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

If the status of the **Bill Segment Generation (C1-BLGEN)** batch run is set to **Error**, you need to first resolve the error and then execute the batch once again.

Post Execution Check/Clean Up:

On successful completion of this batch, the bill segments are generated in the pending trial or actual bills.

2.4 Bill Completion (C1-BLPPR)

The **Bill Completion (C1-BLPPR)** batch is used to generate post-processing bill segments in the pending trial or actual bills during the bill completion process. The adjustments, if any, created on the accounts are swept onto the pending trial or actual bills. In addition, the bill routing information and bill messages are stamped on the pending trial or actual bills. If the adjustment currency is different from the account's invoice currency, the transfer adjustment is created on the pending actual bill. However, in case of pending trial bill, the currency conversion for adjustments is done, but the transfer adjustments are not created. Finally, the status of the actual bill is set to **Complete**, whereas the status of the trial bill remains as **Pending**.

Based on the batch business date, the system identifies the bill cycles whose pending bills must be completed. Once the bill cycles are identified, the system identifies a set of accounts on which these bill cycles are defined. Then, based on a set of parameters specified while executing this batch, the pending bills are completed for accounts that meet the criteria.

If the **Freeze and Complete** check box is selected for the bill cycle for which the pending actual bill is generated, the status of the post-processing bill segments is set to **Frozen** and the status of the actual bill is changed to **Complete**. But, if the **Freeze and Complete** check box is not selected for the bill cycle for which the pending actual bill is generated, the status of the post-processing bill segments is set to **Freezable** and the status of the actual bill remains as **Pending**. However, in case of pending trial bills, irrespective of whether the **Freeze and Complete** check box is selected or not for the bill cycle, the status of the post-processing bill segments is set to **Freezable** and the status of the trial bill remains as **Pending**.

While creating actual bills using trial bills, the system copies trial bills' post-processing bill segments which are in the **Freezable** status to the actual bills. If you want to freeze the post-processing bill segments and complete such actual bills, you need to execute the **Bill Completion (C1-BLPPR)** batch once again without specifying the trial billing batch run number.

Points to Note:

There might be situations when you have selected the **Freeze and Complete** check box for the bill cycle after all three batches are executed. In such scenarios, the system allows you to re-execute these batches. However, to enhance performance, you must first execute the **Bill Segment Generation (C1-BLGEN)** batch and then execute the **Bill Completion (C1-BLPPR)** batch. The status of the bill segments and bills is changed accordingly.

In addition, there might be situations when you want to regenerate the post-processing bill segments before freezing them. In such case, you need to set the **Bill Segment Regeneration on Freeze** option type of the **C1_FINTRANOP** feature configuration to **Y**, and then execute the **Bill Completion (C1-BLPPR)** batch once again. If the **Freeze and Complete** check box is selected for the bill cycle and the **Bill Segment Regeneration on Freeze** option type is set to **Y**, then the post-processing bill segments are regenerated, and the status of the bill segments is set to **Frozen**. Note that if the **Freeze and Complete** check box is not selected for the bill cycle, then the post-processing bill segments are always regenerated irrespective of whether the **Bill Segment Regeneration on Freeze** option type is set to **Y** or not.

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Batch Business Date	No	Used to identify the bill cycles within which the specified date falls. **Note:** If you do not specify any date, the batch business date is set to the current date.
Thread Count	No	Used to specify the number of threads you want to spawn in parallel. **Note:** By default, the thread count is set to 0.
Division	No	Used when you want to initiate the bill completion process for accounts belonging to a particular division. **Note:** You must specify a division which is already defined in the system.
Bill Cycle	No	Used when you want to initiate the bill completion process for accounts having a particular bill cycle. **Points to Note:** You must specify a bill cycle which is already defined in the system. You can specify maximum twenty comma-separated values for this parameter.
Bill Generation Type	No	Used to indicate whether you want to initiate the bill completion process for pending trial or actual bills. If you want to initiate the bill completion process for pending trial bills, the valid value is **Trial**. And, if you want to initiate the bill completion process for pending actual bills or if you want to copy trial bills' post-processing bill segments while creating actual bills using trial bills, you must leave this parameter blank.
Trial Billing Batch Run Number	No	Used to indicate the trial billing batch run number whose trial bills' post-processing bill segments you want to copy to the actual bills. **Note:** This parameter is used only when you are creating actual bills using trial bills.

Parameter Name	Mandatory (Yes or No)	Description
Process All or Selected Accounts	No	Used to indicate whether you want to initiate the bill completion process for all accounts (that meet the criteria) or only for those accounts (that meet the criteria and) where the **Trial Bill** check box is selected. If you want to initiate the bill completion process for those accounts where the **Trial Bill** check box is selected, set the value of this parameter to **Y**. If you want to initiate the bill completion process for all accounts irrespective of whether the **Trial Bill** check box is selected or not, you must leave this parameter blank.
		Points to Note: This parameter is used either when you are creating trial bills or when you are creating actual bills using trial bills. If you set the value of this parameter to **Y** while creating actual bills using trial bills, the system will copy trial bills' post-processing bill segments of only those accounts (that meet the criteria and) where the **Trial Bill** check box is selected. And, if you leave this parameter blank, the system will copy trial bills' post-processing bill segments of all accounts (that meet the criteria) irrespective of whether the **Trial Bill** check box is selected or not. If the post-processing bill segments do not exist for an account while creating actual bills using trial bills, the system generates the post-processing bill segments (if any) for the account.
Account ID	No	Used when you want to initiate the bill completion process for a particular account.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated.
		Note: You must specify a value greater than or equal to zero.

Parameter Name	Mandatory (Yes or No)	Description
Freeze and Complete	No	Used to override the **Freeze and Complete** option defined for the bill cycle. The valid values are: • **Y** – Used when you want to freeze and complete the bill segments irrespective of whether the **Freeze and Complete** option is selected or not for the bill cycle. • **N** – Used when you do not want to freeze and complete the bill segments irrespective of whether the **Freeze and Complete** option is selected or not for the bill cycle. **Note:** If you do not want to override the **Freeze and Complete** option defined for the bill cycle, you must leave this field blank.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

You must specify the same parameters (for example, division, bill cycle, etc.) including the batch business date in the **C1-PNDBL**, **C1-BLGEN** and **C1-BLPPR** batches during the billing process. Otherwise, erroneous results will occur.

Before executing the **Bill Completion (C1-BLPPR)** batch, you need to ensure that the **Bill Segment Generation (C1-BLGEN)** batch is successfully executed.

If the **Bill Completion (C1-BLPPR)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

If the status of the **Bill Completion (C1-BLPPR)** batch run is set to **Error**, you need to first resolve the error and then execute the batch once again.

To improve the performance of the **Bill Completion (C1-BLPPR)** batch, you can store the post-processing price item pricing information. For more information, refer to the [Refresh Pricing (C1-TXNRP)](#) section.

Post Execution Check/Clean Up:

On successful completion of this batch, the post-processing bill segments (if any) are generated in the pending trial or actual bills. The status of the actual bill is changed to **Complete** and the status of the trial bill remains as **Pending**.

2.5   Pending Bill Generation (BILLOPEN)

The **Pending Bill Generation (BILLOPEN)** batch is used to generate pending trial or actual bills for accounts that meet the criteria. It is used for both construct as well as non-construct based billing.

Based on the batch business date, the system identifies the bill cycles for which the pending trial or actual bills must be generated. Once the bill cycles are identified, the system identifies a set of accounts on which these bill cycles are defined. These accounts may be usage or invoice accounts. Then, based on a set of parameters specified while executing this batch, the pending trial or actual bills are generated for accounts that meet the criteria.

In addition, this batch fetches all recurring and non-recurring billable charges of the accounts with the **Adhoc** flag set to **No** and whose start or end date is earlier than or equal to the batch business date. If a billable charge satisfies the rule defined on any active construct, the construct and invoice account details are stamped in the temporary area. And, if a billable charge does not satisfy the rule defined on any active construct, the invoice account details are stamped in the temporary area. Note that in such case, the account will be billed individually and not through any invoice account.

> **Note:**
>
> If the **Billable Charges Date** option type of the **C1_FINTRANOP** feature configuration is set to **E**, the system considers the following:
>
> >> Recurring billable charges whose start date is earlier than the cut-off date and end date is later than the cut-off date
>
> >> Non-recurring billable charges whose end date is earlier than or equal to the cut-off date

On executing the **BILLOPEN** batch with the **Off Cycle Switch** parameter set to **Y** and the **Off Cycle Mode** parameter set to **I**, the system checks whether the account that meets the search criteria is included in any invoice request with the **Defer Processing Batch** status. If the account is included in an invoice request with the **Defer Processing Batch** status, the system checks whether the processing date of the invoice request is earlier than or equal to the batch business date. If the processing date is earlier than or equal to the batch business date, the system checks whether there is any record in the **PROC** (i.e., **Processing)** status for the account in the invoice request. If so, the system checks whether the pending bill already exists for the account. If the pending bill does not exist for the account, the system checks whether the cutoff date is later than the bill after date. If cutoff date is later than the bill after date, the system checks whether the accounting calendar is open for the accounting date. If the accounting calendar is open for the accounting date, the system creates an adhoc trial or actual bill for the account depending on the bill generation type and then sets the status of the bill to **Pending**.

However, if the pending bill exists for the account, or if the cutoff date is earlier than or equal to the bill after date, or if the accounting calendar is not open for the accounting date, or if an error occurs while creating an adhoc trial or actual bill for the account, the status of the respective record in the invoice request is changed to **EROR** (i.e., **Error**).

> **Note:** If all the records in the invoice request are in the **EROR** (i.e., **Error**) status, then the status of the invoice request is also changed to **Error**. The system then enables you to move such erroneous invoice request back to the **Draft** status.

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Batch Business Date	No	Used to identify the bill cycles within which the specified date falls. **Note:** If you do not specify any date, the batch business date is set to the current date.
Thread Count	No	Used to specify the number of threads you want to spawn in parallel. **Note:** By default, the thread count is set to 0.
Division or Division Group	No	Used when you want to generate pending trial or actual bills for accounts belonging to a particular division or division group. **Points to Note:** You must specify a division or division group which is already defined in the system. If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division. If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.
Bill Cycle	No	Used when you want to generate pending trial or actual bills for accounts having a particular bill cycle. **Points to Note:** You must specify a bill cycle which is already defined in the system. You can specify maximum twenty comma-separated values for this parameter.

Parameter Name	Mandatory (Yes or No)	Description
Delete Existing Bill	No	Used to indicate whether you want to delete bill segments from the existing actual bills that are in the **Pending** status. The valid value is **Y**. **Points to Note:** If you do not want to delete the bill segments from an existing pending bill, you must leave this parameter blank. In such case, the existing pending bill is used for further processing. The system does not delete the bill segments from the existing bills that are in the **Complete** status. If the existing pending bills have any bill segments in the **Freezable** or **Error** status, the batch deletes such bill segments. However, if the existing pending bills have any bill segments in the **Frozen** or **Cancelled** status, the batch does not delete such bill segments. If an existing pending bill has a To Do associated with it, the system will complete the To Do and detach it from the bill. The batch will then use the same pending bill of the account for further processing and will not create a new pending bill for the account. This parameter is not supported when you are generating trial bills.
Bill Generation Type	No	Used to indicate whether you want to generate pending trial or actual bills. If you want to generate pending trial bills, the valid value is **Trial**. And, if you want to generate pending actual bills, you must leave this parameter blank.

Parameter Name	Mandatory (Yes or No)	Description
Description for Trial Billing Batch Run	No	Used to specify the description for the batch run. This description is stamped on each trial bill which is generated in the batch run. This helps to identify the trial bills which are generated in a particular batch run.
		Points to Note: This parameter is used only when you are generating pending trial bills. If you do not specify the description, by default, the description is set to "Trial Invoicing Run On YYYY-MM-DD-HH.MM.SS".
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated.
		Note: You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.
Off Cycle Switch	No	Used to indicate whether to consider the **Off Cycle Mode** parameter. The valid value is **Y**. If you set the parameter value to **Y**, the batch considers the **Off Cycle Mode** parameter and accordingly creates the adhoc bills. However, if you specify any value other than **Y** or if you leave this parameter blank, the batch creates regular trial or actual bills for the accounts as per the bill cycle of the account.
Skip Charge When Pending To Do Exists	No	Used when you want to skip the accounts for which a pending bill exists with a To Do. The valid value is **Y**. If you set the parameter value to **Y**, the batch will not extract the details for the skipped accounts' billable charges in the temporary area. However, if you specify any value other than **Y** or if you leave this parameter blank, the batch does not skip the accounts for which a pending bill exists with a To Do while billing.

Parameter Name	Mandatory (Yes or No)	Description
Off Cycle Mode	No	Used to determine whether to create adhoc bills for an account through an invoice request or through a bill correction request. The valid values are: • **I** – Used when you want to process the invoice requests in the **Deferred Processing Batch** status. Note that you can create an adhoc trial or actual bill for an account through an invoice request. • **C** – Used when you want to process the bill correction requests in the **Deferred Regeneration** status. Note that you can only create an adhoc actual bill for an account through a bill correction request. **Note:** If you do not specify any value for this parameter when the **Off Cycle Switch** parameter is set to **Y**, by default, it is set to **I**.
Billing for CMS Account	No	Used to indicate whether you want to create actual or trail bills for the sponsor or membership (i.e., non-sponsor) accounts. The valid values are: • **Y** - Used when you want to skip the non-sponsor accounts (i.e., the accounts where the **IS_CMS_ACCT_SW** field is set to **N**) and create bills for the sponsor accounts (i.e., the accounts where the **IS_CMS_ACCT_SW** field is set to **Y**). • **N** – Used when you want to skip the sponsor accounts (i.e., the accounts where the **IS_CMS_ACCT_SW** field is set to **Y**) and create bills for the non-sponsor accounts (i.e., the accounts where the **IS_CMS_ACCT_SW** field is set to **N**). **Points to Note:** If you do not specify any value for this parameter, the system will create bills for both the sponsor and non-sponsor accounts. The system does not store the sponsor account details in the temporary area and skips those sponsor accounts for which a pending bill already exists.

Parameter Name	Mandatory (Yes or No)	Description
Account ID	No	Used when you want to generate trial or actual bill for a particular account.
Process All or Selected Accounts	No	Used to indicate whether you want to generate trial bills for all the accounts (that meet the criteria) or only for those accounts (that meet the criteria and where the **Trial Bill** option is selected). The valid value is **Y**.
		If you set the parameter value to **Y**, the batch generates trial bills for only those accounts where the **Trial Bill** option is selected. However, if you want to generate trial bills for all the accounts irrespective of whether the **Trial Bill** option is selected or not, you must leave this parameter blank.
		Note: This parameter is used either when you are creating trial bills or when you are creating actual bills using trial bills.

Points to Note:

If the **Pending Bill Generation (BILLOPEN)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

If the status of the **Pending Bill Generation (BILLOPEN)** batch run is set to **Error**, you need to first resolve the error and then execute the batch once again.

Post Execution Check/Clean Up:

On successful completion of this batch, the pending trial or actual bills are generated for the accounts that meet the criteria.

2.6   Bill Segment Generation (BSGENREG)

The **Bill Segment Generation (BSGENREG)** batch is used to generate bill segments in the pending trial or actual bills. It is also used to copy trial bills' bill segments to the actual bills while creating actual bills using trial bills. It is used for both construct as well as non-construct based billing.

If the construct and invoice account details are stamped corresponding to a billable charge in the temporary area, the bill segment is generated on the pending trial or actual bill of the invoice account. However, if only the invoice account details are stamped corresponding to a billable charge in the temporary area, the bill segment is generated on the pending trial or actual bill of the usage account.

If the **Freeze and Complete** check box is selected for the bill cycle for which the pending actual bill is generated, the status of the bill segments is set to **Frozen**. But, if the **Freeze and Complete** check box is not selected for the bill cycle for which the pending actual bill is generated, the status of the bill segments is set to **Freezable**. However, in case of pending trial bills, irrespective of whether the **Freeze and Complete** check box is selected or not for the bill cycle, the status of the bill segments is set to **Freezable**.

While creating actual bills using trial bills, the system copies trial bills' bill segments which are in the **Freezable** status to the actual bills. If you want to freeze the bill segments in such actual bills, you need to execute the **Bill Segment Generation (BSGENREG)** batch once again without specifying the trial billing batch run number.

> **Points to Note:**
>
> There might be situations when you have received certain billing information after the bill and bill segments are generated. In such scenarios, you must first execute the **Pending Bill Generation (BILLOPEN)** batch to fetch the newly created billable charges and then execute the **Bill Segment Generation (BSGENREG)** batch. While executing the **Pending Bill Generation (BILLOPEN)** batch, you must leave the **Delete Existing Bill** parameter blank indicating that you want to use the existing pending bill for further processing.
>
> In addition, there might be situations when you want to regenerate the bill segments before freezing them. In such case, you need to set the **Bill Segment Regeneration on Freeze** option type of the **C1_FINTRANOP** feature configuration to **Y**, and then execute the **Pending Bill Generation (BILLOPEN)** and **Bill Segment Generation (BSGENREG)** batches once again. While executing the **Pending Bill Generation (BILLOPEN)** batch, you must leave the **Delete Existing Bill** parameter blank indicating that you want to use the existing pending bill for further processing. If the **Freeze and Complete** check box is selected for the bill cycle and the **Bill Segment Regeneration on Freeze** option type is set to **Y**, then the bill segments are regenerated and the status of the bill segments is set to **Frozen**. Note that if the **Freeze and Complete** check box is not selected for the bill cycle, then the bill segments are always regenerated irrespective of whether the **Bill Segment Regeneration on Freeze** option type is set to **Y** or not.

On executing the **BSGENREQ** batch with the **Off Cycle Switch** parameter set to **Y** and the **Off Cycle Mode** parameter set to **I**, the system creates bill segments for an adhoc trial or actual bill that is created for an account through an invoice request. However, if an error occurs while creating bill segments for the adhoc trial or actual bills, the status of the respective record in the invoice request is changed to **EROR** (i.e., **Error**).

> **Note:** If all the records in the invoice request are in the **EROR** (i.e., **Error**) status, then the status of the invoice request is also changed to **Error**. The system then enables you to move such erroneous invoice request back to the **Draft** status.

This batch is a multi-threaded batch. The multi-threading is based on billable charge ID and chunks for multi-threading are created based on numerical distribution of billable charge ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Batch Business Date	No	Used to identify the billable charges for which bill segments must be generated. **Points to Note:** All billable charges whose start or end date is earlier than or equal to the specified date are considered during bill segment generation.

Parameter Name	Mandatory (Yes or No)	Description
		If you do not specify any date, the batch business date is set to the current date.
Thread Count	No	Used to specify the number of threads you want to spawn in parallel.
		Note: By default, the thread count is set to 0.
Division or Division Group	No	Used when you want to generate bill segments for accounts belonging to a particular division or division group.
		Points to Note:
		You must specify a division or division group which is already defined in the system.
		If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division.
		If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.
Bill Cycle	No	Used when you want to generate bill segments for accounts having a particular bill cycle.
		Points to Note:
		You must specify a bill cycle which is already defined in the system.
		You can specify maximum twenty comma-separated values for this parameter.
Bill Generation Type	No	Used to indicate whether you want to generate bill segments for pending trial or actual bills. If you want to generate bill segments for pending trial bills, the valid value is **Trial**. And, if you want to generate bill segments for pending actual bills or if you want to copy trial bills' bill segments while creating actual bills using trial bills, you must leave this parameter blank.

Parameter Name	Mandatory (Yes or No)	Description
Trial Billing Batch Run Number	No	Used to indicate the trial billing batch run number whose trial bills' bill segments you want to copy to the actual bills. **Note:** This parameter is used only when you are creating actual bills using trial bills.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Freeze and Complete	No	Used to override the **Freeze and Complete** option defined for the bill cycle. The valid values are: • **Y** – Used when you want to freeze and complete the bill segments irrespective of whether the **Freeze and Complete** option is selected or not for the bill cycle. • **N** – Used when you do not want to freeze and complete the bill segments irrespective of whether the **Freeze and Complete** option is selected or not for the bill cycle. **Note:** If you do not want to override the **Freeze and Complete** option defined for the bill cycle, you must leave this field blank.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Off Cycle Switch	No	Used to indicate whether to consider the **Off Cycle Mode** parameter. The valid value is **Y**. If you set the parameter value to **Y**, the batch considers the **Off Cycle Mode** parameter and accordingly creates bill segments for the adhoc bills. However, if you specify any value other than **Y** or if you leave this parameter blank, the batch creates bill segments for the accounts as per the bill cycle of the account.

Parameter Name	Mandatory (Yes or No)	Description
Skip Charge When Pending To Do Exists	No	Used when you want to skip the billable charges for which bill segments are already generated on a pending bill with a To Do. The valid value is **Y**. If you set the parameter value to **Y**, the batch will not regenerate bill segments for the skipped billable charges. However, if you specify any value other than **Y** or if you leave this parameter blank, the batch does not skip the billable charges, for which bill segments are already generated on a pending bill with a To Do, while processing.
Regenerate Bill Segment	No	Used when you do not want to delete and regenerate existing bill segments on the bill. The valid value is **N**. If you specify any value other than **N** or if you leave this parameter blank, the batch deletes and regenerates existing bill segments on the bill.
Off Cycle Mode	No	Used to indicate whether you want to create bill segments for the accounts whose adhoc bill is generated through an invoice request or through a bill correction request. The valid values are: • **I** – Used when you want to create bill segments for the accounts whose adhoc trial or actual bill is generated through an invoice request. • **C** – Used when you want to create bill segments for the accounts whose adhoc actual bill is generated through a bill correction request. **Note:** If you do not specify any value for this parameter when the **Off Cycle Switch** parameter is set to **Y**, by default, it is set to **I**.
Account ID	No	Used when you want to generate bill segments for a particular account.
Billing Processing Sequence	No	Used to specify the billing processing sequence of the contract's contract type whose bill segments you want to generate. This parameter is useful when you want to generate bill segments in a particular order of importance.

Parameter Name	Mandatory (Yes or No)	Description
		Note: You can specify comma separated values for this parameter. For example, 10, 20, 30. In such case, the bill segments of contracts whose contract types' billing processing sequence is 10, 20, or 30 will be generated simultaneously.
Process All or Selected Accounts	No	Used to indicate whether you want to generate bill segments for all the accounts (that meet the criteria) or only for those accounts (that meet the criteria and where the **Trial Bill** option is selected). The valid value is **Y**.
		If you set the parameter value to **Y**, the batch generates bill segments for only those accounts where the **Trial Bill** option is selected. However, if you want to generate bill segments for all the accounts irrespective of whether the **Trial Bill** option is selected or not, you must leave this parameter blank.
		Note: This parameter is used either when you are creating trial bills or when you are creating actual bills using trial bills.

Points to Note:

You must specify the same parameters (for example, division, bill cycle, etc.) including the batch business date in the **BILLOPEN**, **BSGENREG**, and **POSTPROC** batches during the billing process. Otherwise, erroneous results will occur.

Before executing the **Bill Segment Generation (BSGENREG)** batch, you need to ensure that the **Pending Bill Generation (BILLOPEN)** batch is successfully executed.

If the **Bill Segment Generation (BSGENREG)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

If the status of the **Bill Segment Generation (BSGENREG)** batch run is set to **Error**, you need to first resolve the error and then execute the batch once again.

Post Execution Check/Clean Up:

On successful completion of this batch, the bill segments are generated in the pending trial or actual bills.

2.7   Bill Completion (POSTPROC)

The **Bill Completion (POSTPROC)** batch is used to generate post-processing bill segments in the pending trial or actual bills during the bill completion process. It is used for both construct as well as non-construct based billing.

The post-processing bill segments, if any, are generated on the invoice or usage account depending on the type of price assignment. If an adjustment created on the account satisfies the rule defined on any active construct, the adjustment is swept onto the pending trial or actual bill of the invoice account. However, if an adjustment created on the account does not satisfy the rule defined on any active construct, the adjustment is swept onto the pending trial or actual bill of the usage account.

In addition, the bill routing information and bill messages are stamped on the pending trial or actual bills. If the adjustment currency is different from the account's invoice currency, the transfer adjustment is created on the pending actual bill. However, in case of pending trial bill, the currency conversion for adjustments is done, but the transfer adjustments are not created. Finally, the status of the actual bill is set to **Complete**, whereas the status of the trial bill remains as **Pending**.

Based on the batch business date, the system identifies the bill cycles whose pending bills must be completed. Once the bill cycles are identified, the system identifies a set of accounts on which these bill cycles are defined. Then, based on a set of parameters specified while executing this batch, the pending bills are completed for accounts that meet the criteria.

If the **Freeze and Complete** check box is selected for the bill cycle for which the pending actual bill is generated, the status of the post-processing bill segments is set to **Frozen** and the status of the actual bill is changed to **Complete**. But, if the **Freeze and Complete** check box is not selected for the bill cycle for which the pending actual bill is generated, the status of the post-processing bill segments is set to **Freezable** and the status of the actual bill remains as **Pending**. However, in case of pending trial bills, irrespective of whether the **Freeze and Complete** check box is selected or not for the bill cycle, the status of the post-processing bill segments is set to **Freezable** and the status of the trial bill remains as **Pending**.

While creating actual bills using trial bills, the system copies trial bills' post-processing bill segments which are in the **Freezable** status to the actual bills. If you want to freeze the post-processing bill segments and complete such actual bills, you need to execute the **Bill Completion (POSTPROC)** batch once again without specifying the trial billing batch run number.

Points to Note:

There might be situations when you have selected the **Freeze and Complete** check box for the bill cycle after all three batches are executed. In such scenarios, the system allows you to re-execute these batches. However, to enhance performance, you must first execute the **Bill Segment Generation (BSGENREG)** batch and then execute the **Bill Completion (POSTPROC)** batch. The status of the bill segments and bills is changed accordingly.

In addition, there might be situations when you want to regenerate the post-processing bill segments before freezing them. In such case, you need to set the **Bill Segment Regeneration on Freeze** option type of the **C1_FINTRANOP** feature configuration to **Y**, and then execute the **Bill Completion (POSTPROC)** batch once again. If the **Freeze and Complete** check box is selected for the bill cycle and the **Bill Segment Regeneration on Freeze** option type is set to **Y**, then the post-processing bill segments are regenerated, and the status of the bill segments is set to **Frozen**. Note that if the **Freeze and Complete** check box is not selected for the bill cycle, then the post-processing bill segments are always regenerated irrespective of whether the **Bill Segment Regeneration on Freeze** option type is set to **Y** or not.

On executing the **POSTPROC** batch with the **Off Cycle Switch** parameter set to **Y** and the **Off Cycle Mode** parameter set to **I**, the system creates the post-processing bill segments (if any) for an adhoc trial or actual bill that is created for an account through an invoice request. The system will then freeze and complete the adhoc actual bills created through an invoice request. However, the system will not freeze and complete the adhoc trial bills created through an invoice request. Finally, the status of the invoice request is changed to **Processed**.

However, if an error occurs while creating post-processing bill segments for the adhoc trial or actual bills, or if a To Do is generated for the adhoc actual bill review, the status of the respective record in the invoice request is changed to **EROR** (i.e., **Error**).

Note: If all the records in the invoice request are in the **EROR** (i.e., **Error**) status, then the status of the invoice request is also changed to **Error**. The system then enables you to move such erroneous invoice request back to the **Draft** status.

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Batch Business Date	No	Used to identify the bill cycles within which the specified date falls.
		Note: If you do not specify any date, the batch business date is set to the current date.
Thread Count	No	Used to specify the number of threads you want to spawn in parallel.
		Note: By default, the thread count is set to 0.

Parameter Name	Mandatory (Yes or No)	Description
Division or Division Group	No	Used when you want to initiate the bill completion process for accounts belonging to a particular division or division group.
		Points to Note:
		You must specify a division or division group which is already defined in the system.
		If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division.
		If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.
Bill Cycle	No	Used when you want to initiate the bill completion process for accounts having a particular bill cycle.
		Points to Note:
		You must specify a bill cycle which is already defined in the system.
		You can specify maximum twenty comma-separated values for this parameter.
Bill Generation Type	No	Used to indicate whether you want to initiate the bill completion process for pending trial or actual bills. If you want to initiate the bill completion process for pending trial bills, the valid value is **Trial**. And, if you want to initiate the bill completion process for pending actual bills or if you want to copy trial bills' post-processing bill segments while creating actual bills using trial bills, you must leave this parameter blank.

Parameter Name	Mandatory (Yes or No)	Description
Trial Billing Batch Run Number	No	Used to indicate the trial billing batch run number whose trial bills' post-processing bill segments you want to copy to the actual bills. **Note:** This parameter is used only when you are creating actual bills using trial bills.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Freeze and Complete	No	Used to override the **Freeze and Complete** option defined for the bill cycle. The valid values are: • **Y** – Used when you want to freeze and complete the bill segments irrespective of whether the **Freeze and Complete** option is selected or not for the bill cycle. • **N** – Used when you do not want to freeze and complete the bill segments irrespective of whether the **Freeze and Complete** option is selected or not for the bill cycle. **Note:** If you do not want to override the **Freeze and Complete** option defined for the bill cycle, you must leave this field blank.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Off Cycle Switch	No	Used to indicate whether to consider the **Off Cycle Mode** parameter. The valid value is **Y**. If you set the parameter value to **Y**, the batch considers the **Off Cycle Mode** parameter and accordingly creates post-processing bill segments (if any) for the adhoc bills. It also completes the adhoc actual bills.

Parameter Name	Mandatory (Yes or No)	Description
		However, if you specify any value other than **Y** or if you leave this parameter blank, the batch create post-processing bill segments (if any) for the accounts as per the bill cycle of the account. It also completes actual bills of the accounts.
Skip Account When Pending To Do Exists	No	Used when you want to skip the accounts for which a pending bill exists with a To Do. The valid value is **Y**.
		If you set the parameter value to **Y**, the batch will not extract the details for the skipped accounts' billable charges in the temporary area. However, if you specify any value other than **Y** or if you leave this parameter blank, the batch does not skip the accounts for which a pending bill exists with a To Do while billing. In addition, it does the following:
		• If the bill tolerance To Do is created, the batch will not clear the respective records in the temporary area and the bill will not be completed.
		• If the bill tolerance To Do is not created, the batch will clear the respective records in the temporary area and the bill will be completed.
Off Cycle Mode	No	Used to indicate whether you want to create post-processing bill segments (if any) for the accounts whose adhoc bill is generated through an invoice request or through a bill correction request.
		It also completes adhoc actual bills of the accounts. The valid values are:
		• **I** – Used when you want to create post-processing bill segments (if any) for the accounts whose adhoc trial or actual bill is generated through an invoice request.
		• **C** – Used when you want to create post-processing bill segments (if any) for the accounts whose adhoc actual bill is generated through a bill correction request.

Parameter Name	Mandatory (Yes or No)	Description
		Points to Note: If you do not specify any value for this parameter when the **Off Cycle Switch** parameter is set to **Y**, by default, it is set to **I**. While creating a batch job using the **POSTPROC** batch control with the **Off Cycle Mode** parameter set to **C**, you need to specify your email address. The system will then internally update the status of the impacted bill, respective correction entity, and bill correction request once the batch is successfully executed.
Billing for CMS Account	No	Used to indicate whether the system should freeze and complete bills of the sponsor or membership (i.e., non-sponsor) accounts. The valid values are: • **Y** - Used when you want to skip the non-sponsor accounts (i.e., the accounts where the **IS_CMS_ACCT_SW** field is set to **N**) and freeze and complete the bills of the sponsor accounts (i.e., the accounts where the **IS_CMS_ACCT_SW** field is set to **Y**). • **N** – Used when you want to skip the sponsor accounts (i.e., the accounts where the **IS_CMS_ACCT_SW** field is set to **Y**) and freeze and complete the bills of the non-sponsor accounts (i.e., the accounts where the **IS_CMS_ACCT_SW** field is set to **N**). **Points to Note:** If you do not specify any value for this parameter, the system will skip the sponsor accounts and freeze and complete the bills of the non-sponsor accounts. The system will check whether the bill segment on the sponsor account's bill is canceled. If so, the system will stamp the bill ID of the sponsor account on the canceled bill segment's financial transaction.

Parameter Name	Mandatory (Yes or No)	Description
Account ID	No	Used when you want to initiate the bill completion process for a particular account.
Process All or Selected Accounts	No	Used to indicate whether you want to generate post-processing bill segments (if any) for all the accounts (that meet the criteria) or only for those accounts (that meet the criteria and where the **Trial Bill** option is selected). The valid value is **Y**.
		If you set the parameter value to **Y**, the batch generates post-processing bill segments (if any) for only those accounts where the **Trial Bill** option is selected. However, if you want to generate post-processing bill segments (if any) for all the accounts irrespective of whether the **Trial Bill** option is selected or not, you must leave this parameter blank.
		Note: This parameter is used either when you are creating trial bills or when you are creating actual bills using trial bills.

Points to Note:

You must specify the same parameters (for example, division, bill cycle, etc.) including the batch business date in the **BILLOPEN**, **BSGENREG**, and **POSTPROC** batches during the billing process. Otherwise, erroneous results will occur.

Before executing the **Bill Completion (POSTPROC)** batch, you need to ensure that the **Bill Segment Generation (BSGENREG)** batch is successfully executed.

If the **Bill Completion (POSTPROC)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

If the status of the **Bill Completion (POSTPROC)** batch run is set to **Error**, you need to first resolve the error and then execute the batch once again.

Post Execution Check/Clean Up:

On successful completion of this batch, the post-processing bill segments (if any) are generated in the pending trial or actual bills. The status of the actual bill is changed to **Complete** and the status of the trial bill remains as **Pending**.

2.8   Refresh Pricing (C1-TXNRP)

You can store the regular and post-processing price item pricing information in the following tables and thereby improve the performance of the **Price Item Pricing Verification (C1-TXNVP)** and **Bill Completion (C1-BLPPR)** batches:

- CI_PRC_AGRD
- CI_PRC_PL
- CI_PRC_INH_PL

If you set the **Materialized View Used** option type of the **C1_FM** feature configuration to **true**, the system will store the regular and post-processing price item pricing information in the above mentioned tables. But, if you set the **Materialized View Used** option type of the **C1_FM** feature configuration to **false**, the system will not store the regular and post-processing price item pricing information in the above mentioned tables.

This mechanism to store pricing information in the above mentioned tables helps to quickly search for regular or post-processing price item pricing information while executing the **Price Item Pricing Verification (C1-TXNVP)** or **Bill Completion (C1-BLPPR)** batch, respectively. The system stores all price items pricing information irrespective of whether it is effective or not. If the price assignment ID is not stamped on any billable charge, the system refers to the price assignment search algorithm which searches for the pricing in the original tables and not in the above mentioned tables.

If there are any pricing changes, you will have to update these tables before you execute the **Price Item Pricing Verification (C1-TXNVP)** or **Bill Completion (C1-BLPPR)** batch. The **Refresh Pricing (C1-TXNRP)** batch is used to update the regular and post-processing price item pricing information in the above mentioned tables.

This batch is a multi-threaded batch. The multi-threading is based on person ID and chunks for multi-threading are created based on numerical distribution of person ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Count	No	Used to specify the number of threads you want to spawn in parallel.
		Note: By default, the thread count is set to 0.
Chunk Size	Yes	Used to specify the number of persons whose regular and post-processing price item pricing information you want to update in each work unit.
Division	No	Used when you want to update the price item pricing information of accounts belonging to a particular division.
		Note: You must specify a division which is already defined in the system.

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Refresh Pricing (C1-TXNRP)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the regular and post-processing price item pricing information is updated in the **CI_PRC_AGRD**, **CI_PRC_PL**, and **CI_PRC_INH_PL** tables. During this process, the existing data is first removed from these tables, and then the latest information is added into these tables.

2.9   Upload and Validate Usage Data File (C1-ODFU)

The **Upload and Validate Usage Data File (C1-ODFU)** batch is used to upload and validate a usage data file. It is also used for revalidating a usage data file which is already uploaded in the system.

While uploading or revalidating a usage data file, a batch job is created using this batch automatically when the usage data file size exceeds the file size limit defined in the **On Demand Billing (C1-ODBFC)** feature configuration.

This batch is a multi-threaded batch. The multi-threading is based on staging ID and chunks for multi-threading are created based on numerical distribution of staging ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
File Group	Yes	Used to specify the group in which you want to upload the usage data file.
File Name	Yes	Used to specify the name of the usage data file that you want to upload or revalidate.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Points to Note:
		If you do not specify any value, by default, it is set to **DEFAULT**.
		If a batch job is created automatically, the value defined in the **Thread Pool Name** option type of the **On Demand Billing (C1-ODBFC)** feature configuration is used.
File Type	Yes	Used to specify the type of usage data file.

Parameter Name	Mandatory (Yes or No)	Description
Billing Method	Yes	Used to specify the type of billing you want to use for the usage data file. The valid values are: • ADB • RLB **Note:** ADB stands for adhoc billing and RLB stands for regular billing.
Access Group	No	Used to specify the access group of the user who is uploading or revalidating the usage data file.
Bill After Date	Yes (Conditional)	Used to specify the date after which the bills must be generated for the usage data file. **Note:** This parameter is required only when the billing method is **ADB**.
Default Usage Start Date	Yes	Used to specify the date from when the price item was used. The default usage start date is used when the start date is not specified for a record in the usage data file.
Default Usage End Date	Yes	Used to specify the date till when the price item was used. The default usage end date is used when the end date is not specified for a record in the usage data file.
File Path	Yes (Conditional)	Used to specify the path of the usage data file that you want to upload. **Note:** This parameter is not required when you are revalidating a usage data file.
Revalidate	Yes	Used to indicate whether you want to revalidate the usage data file. The valid values are: • **Y** – Used when you want to revalidate a usage data file. • **N** – Used when you want to upload and validate a usage data file.

Parameter Name	Mandatory (Yes or No)	Description
Chunk Size	Yes	Used to specify the number of usage records you want to validate or revalidate in each work unit.
		Note: If a batch job is created automatically, the value defined in the **Validation Chunk Size** option type of the **On Demand Billing (C1-ODBFC)** feature configuration is used.

Points to Note:

If a batch job is created automatically, the values for these parameters are automatically specified in the batch job depending on whether you are uploading or revalidating a usage data file.

If the **Upload and Validate Usage Data File (C1-ODFU)** batch fails or aborts due to some reason, you cannot restart the batch. Instead, you need to upload the usage data file once again in the system.

Post Execution Check/Clean Up:

On successful completion of this batch, records are either uploaded and validated or revalidated in the system. If a usage record is successfully validated, the status of the usage record is changed to **Valid**. And, if a usage record could not pass through the validation process, the status of the usage record is changed to **Invalid**.

In addition, a To Do entry is created using the **C1-ODFU** To Do type. You can view this To Do entry only when you are assigned a To Do role which is mapped to the **C1-ODFU** To Do type. This To Do indicates whether the batch was successfully executed or not. If the batch was successfully executed, it indicates the following:

- Number of records that were uploaded in the system (appears only on uploading a usage data file)
- Number of records which were successfully validated
- Number of records which could not pass through the validation process
- Number of records which failed to validate due to some technical reasons

2.10   Billable Charge Creation (C1-ODBCH)

The **Billable Charge Creation (C1-ODBCH)** batch is used to create a billable charge for a valid usage record. While submitting or approving usage records, a batch job is created using this batch automatically when the total number of valid records in a usage data file exceeds the transaction limit defined in the **On Demand Billing (C1-ODBFC)** feature configuration.

This batch is a multi-threaded batch. The multi-threading is based on staging ID and chunks for multi-threading are created based on numerical distribution of staging ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
File Name	Yes	Used to specify the name of the usage data file for which you want to create the billable charges.
Thread Pool Name	Yes	Used to specify the thread pool on which you want to execute the batch.
		Points to Note: If you do not specify any value, by default, it is set to **DEFAULT**. If a batch job is created automatically, the value defined in the **Thread Pool Name** option type of the **On Demand Billing (C1-ODBFC)** feature configuration is used.
Chunk Size	Yes	Used to specify the number of valid records for which you want to create billable charges in each work unit.
		Note: If a batch job is created automatically, the value defined in the **Billable Charge Chunk Size** option type of the **On Demand Billing (C1-ODBFC)** feature configuration is used.
File Group Header ID	No	Used to specify the header ID of the file group for which you want to create the billable charges.

Points to Note:

If a batch job is created automatically, the values for these parameters are automatically specified in the batch job.

If the **Billable Charge Creation (C1-ODBCH)** batch fails or aborts due to some reason, you cannot restart the batch. Instead, you need to submit or approve the usage records once again in the system.

Post Execution Check/Clean Up:

On successful completion of this batch, a billable charge is created for each valid record. The status of the usage record is changed to **Processed**.

In addition, a To Do entry is created using the **C1-ODBCH** To Do type. You can view this To Do entry only when you are assigned a To Do role which is mapped to the **C1-ODBCH** To Do type. This To Do indicates whether the batch was successfully executed or not. If the batch was successfully executed, it indicates the following:

- Number of records for which billable charges were created
- Number of records for which billable charges could not be created successfully

2.11   Adhoc Billing (C1-FABL)

The **Adhoc Billing (C1-FABL)** batch is used to generate an adhoc bill for a file group and account combination or for an account when the bill after date is earlier than or equal to the batch business date. Both, automatically generated and manually created billable charges (with the **Adhoc Bill** flag set to **Y**) are considered during adhoc billing.

While generating adhoc bills through the **On Demand Billing** screen, a batch job is created using this batch automatically when the total number of billable charges for a file group exceeds the online bill creation limit defined in the **On Demand Billing (C1-ODBFC)** feature configuration.

Once the billable charges are identified, bill segments are generated in the adhoc bill. The status of bill segments is set to **Freezable** and the status of adhoc bill is set to **Pending**. In the following scenarios, the system behaves in the following manner:

- If an adhoc bill with the **Pending** status already exists for an account or for a file group and account combination, then the existing bill is deleted and regenerated.
- If an adhoc bill with the **Completed** status already exists for an account or for a file group and account combination, then a new bill is generated (but the existing bill is not deleted).
- If a completed bill of an account or of a file group and account combination is cancelled (in other words, bill segments in an adhoc bill are cancelled), a new bill is generated (but the existing bill is not deleted).

This batch is a multi-threaded batch. The chunks for multi-threading are created randomly and there is no specific logic defined for chunking in the system. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Batch Business Date	No	Used to identify the billable charges for which bill segments must be generated. If the file group is specified, the billable charges that meet the following criteria are considered during adhoc billing: • Billable charges are created for accounts whose usage data is uploaded in the specified file group. • The **Adhoc Bill** flag of the billable charge is set to **Y**. • Billable charges whose bill after date is earlier than or equal to the specified date. **Points to Note:** If you do not specify any date, the batch business date is set to the current date. This parameter is used to identify billable charges only when the cut-off date is not specified.
File Group	No	Used to indicate the file group for which you want to generate the adhoc bills. **Note:** If a batch job is created automatically, the value for this parameter is automatically specified in the batch job.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Points to Note:** If you do not specify any value, by default, it is set to **DEFAULT**. If a batch job is created automatically, the value defined in the **Thread Pool Name** option type of the **On Demand Billing (C1-ODBFC)** feature configuration is used.
Division	No	Used when you want to generate adhoc bills for accounts belonging to a particular division. **Note:** You must specify a division which is already defined in the system.

Parameter Name	Mandatory (Yes or No)	Description
Account ID	No	Used when you want to generate adhoc bills for a particular account.
Lead Days	No	Used to specify the number of lead days that you want to consider while identifying the billable charges. If you specify the lead days, the system considers all adhoc billable charges whose bill after date is earlier than or equal to batch business date + lead days.
Cutoff Date	No	Used to identify the billable charges for which bill segments must be generated. If the file group is specified, the billable charges that meet the following criteria are considered during adhoc billing: • Billable charges are created for accounts whose usage data is uploaded in the specified file group. • The **Adhoc Bill** flag of the billable charge is set to **Y**. • Billable charges whose bill after date is earlier than or equal to the specified date.
Accounting Date	No	Used to specify the accounting date. **Note:** The accounting date cannot be earlier than the cutoff date.

Note: If the **Adhoc Billing (C1-FABL)** batch fails or aborts due to some reason, you cannot restart the batch. Instead, you need to generate adhoc bills for the closed file group once again in the system.

Post Execution Check/Clean Up:

On successful completion of this batch, an adhoc bill is generated for an account or for a file group and account combination when the bill after date is earlier than or equal to the batch business date or batch business date + lead days.

In addition, a To Do entry is created using the **C1-FABL** To Do type. You can view this To Do entry only when you are assigned a To Do role which is mapped to the **C1-FABL** To Do type. This To Do indicates whether the batch was successfully executed or not. If the batch was successfully executed, it indicates the following:

• Number of billable charges which were processed

• Number of billable charges for which the bill after date is a future date

• Number of adhoc bills generated for the usage data file

• Number of billable charges which could not be processed successfully

2.12   Freeze and Complete Adhoc Bills (C1-FCADH)

The **Freeze and Complete Adhoc Bills (C1-FCADH)** batch is used to freeze and complete all adhoc bills of a file group which are in the **Pending** status. While executing this batch, the system does not freeze and complete pending adhoc bills which have bill segments in the **Error** status.

While freezing and completing adhoc bills through the **On Demand Billing** screen, a batch job is created using this batch automatically.

This batch is a multi-threaded batch. The chunks for multi-threading are created randomly and there is no specific logic defined for chunking in the system. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
File Group	No	Used to indicate the file group whose adhoc bills you want to freeze and complete. **Note:** If a batch job is created automatically, the value for this parameter is automatically specified in the batch job.
Accounting Date	No	Used when you want to override the accounting date specified while creating adhoc bills.
Bill Date	No	Used to specify the bill date that you want to display on the adhoc bills. **Note:** If you do not specify any date, the bill date is set to the current date.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Points to Note:** If you do not specify any value, by default, it is set to **DEFAULT**. If a batch job is created automatically, the value defined in the **Thread Pool Name** option type of the **On Demand Billing (C1-ODBFC)** feature configuration is used.
Division	No	Used when you want to freeze and complete adhoc bills of accounts belonging to a particular division. **Note:** You must specify a division which is already defined in the system.

Note: If the **Freeze and Complete Adhoc Bills (C1-FCADH)** batch fails or aborts due to some reason, you cannot restart the batch. Instead, you need to freeze and complete adhoc bills of the closed file group once again in the system.

Post Execution Check/Clean Up:

On successful completion of this batch, adhoc bills of the file group are completed. The status of the adhoc bills is changed to **Complete**.

In addition, a To Do entry is created using the **C1-FCADH** To Do type. You can view this To Do entry only when you are assigned a To Do role which is mapped to the **C1-FCADH** To Do type. This To Do indicates whether the batch was successfully executed or not. If the batch was successfully executed, it indicates the following:

- Number of bills which were picked up for completion
- Number of bills which were completed
- Number of bills which could not be completed successfully

2.13   Pending Bill Segments Deletion (C1-BSEGD)

The **Pending Bill Segments Deletion (C1-BSEGD)** batch is used to delete bill segments which are in the **Freezable** or **Error** status. It does not delete bill segments which are in the **Pending Cancel**, **Frozen**, or **Cancel** status.

This batch is a multi-threaded batch. The multi-threading is based on bill segment ID and chunks for multi-threading are created based on numerical distribution of bill segment ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Count	No	Used to specify the number of threads you want to spawn in parallel. **Note:** By default, the thread count is set to 0.
Account ID	No	Used to indicate the account whose bill segments you want to delete.

Parameter Name	Mandatory (Yes or No)	Description
Division or Division Group	No	Used when you want to delete bill segments of accounts belonging to a particular division or division group. **Points to Note:** You must specify a division or division group which is already defined in the system. If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division. If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.
Bill Cycle	No	Used when you want to delete bill segments of accounts having a particular bill cycle.
Request Type	No	Used to indicate whether you want to delete bill segments during the disaggregation process. The valid values are: • **BILLING** – Used when you want to delete bill segments of pending bills of accounts that meet the criteria. • **DISAGG** – Used when you want to delete bill segments of pending bills of accounts that meet the criteria and for which the disaggregation request is created. **Note:** If you do not specify any value, by default, the parameter value is set to **BILLING**.
Chunk Size	Yes	Used to specify the number of transactions you want to execute in each work unit.
Maximum Batch Count	Yes	Used to specify the maximum number of transactions after which the data must be committed in the database.

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. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

At present, the batch business date is not used (or considered) while executing the **Pending Bill Segments Deletion (C1-BSEGD)** batch.

If the **Pending Bill Segments Deletion (C1-BSEGD)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the bill segments in the **Freezable** or **Error** status and their corresponding financial transactions are deleted. The corresponding bill ID is added in the **CI_DELETE_BILL_DETAIL** table for further processing.

2.14   Pending Bill Deletion (C1-PNBD)

The **Pending Bill Deletion (C1-PNBD)** batch checks whether the pending bills listed in the **CI_DELETE_BILL_DETAIL** table have any bill segments in the **Pending Cancel**, **Frozen**, or **Cancel** status. If there are bill segments in the **Pending Cancel**, **Frozen**, or **Cancel** status, the pending bill is not deleted. However, if there are no bill segments in the **Pending Cancel**, **Frozen**, or **Cancel** status, the pending bill is deleted. This batch deletes regular pending bills and not adhoc pending bills.

Note: If the pending bills have bill segments, you must first execute the **Pending Bill Segments Deletion (C1-BSEGD)** batch and then execute the **Pending Bill Deletion (C1-PNBD)** batch. While executing these batches in the specified order, ensure that you specify the same parameters in both these batches.

This batch is a multi-threaded batch. The multi-threading is based on bill ID and chunks for multi-threading are created based on numerical distribution of bill ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Count	No	Used to specify the number of threads you want to spawn in parallel. **Note:** By default, the thread count is set to 0.
Account ID	No	Used to indicate the account whose pending bills you want to delete.
Division or Division Group	No	Used when you want to delete pending bills of accounts belonging to a particular division or division group.

Parameter Name	Mandatory (Yes or No)	Description
		Points to Note: You must specify a division or division group which is already defined in the system. If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division. If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.
Bill Cycle	No	Used when you want to delete pending bills of accounts having a particular bill cycle. **Note:** You must specify a bill cycle which is already defined in the system.
Request Type	No	Used to indicate whether you want to delete pending bills during the disaggregation process. The valid values are: • **BILLING** – Used when you want to delete pending bills of accounts that meet the criteria. • **DISAGG** – Used when you want to delete pending bills of accounts that meet the criteria and for which the disaggregation request is created. **Note:** If you do not specify any value, by default, the parameter value is set to **BILLING**.

Parameter Name	Mandatory (Yes or No)	Description
Chunk Size	Yes	Used to specify the number of transactions you want to execute in each work unit.
Maximum Batch Count	Yes	Used to specify the maximum number of transactions after which the data must be committed in the database.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

At present, the batch business date is not used (or considered) while executing the Pending Bill Deletion (C1-PNBD) batch.

If the **Pending Bill Deletion (C1-PNBD)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the **BILL_ACT_CD** column corresponding to a pending bill in the **CI_DELETE_BILL_DETAIL** table is set to either **10** or **20**. **10** indicate that the pending bill contains no bill segments in the **Pending Cancel**, **Frozen**, or **Cancel** status, and therefore can be deleted. However, **20** indicate that the pending bill contains bill segments in the **Pending Cancel**, **Frozen**, or **Cancel** status, and therefore cannot be deleted. If the **BILL_ACT_CD** column corresponding to a pending bill is set to 10, the system deletes the pending bill.

2.15   Assign Sequential Bill Numbers (ASSGNSBN)

The **Assign Sequential Bill Numbers (ASSGNSBN)** batch is used to generate alternate sequential bill numbers for the completed and pending bills. It considers only those bills for which the alternate sequential bill number is not yet generated.

You need to configure this batch such that it is executed at regular intervals. Note that you need to schedule this batch only when the **Use Sequential Bill Numbers** option is selected in the **Billing** tab of the **Installation Options** screen.

This batch invokes the following algorithm:

If the alternate sequential bill numbers must be….	Then the batch…
Unique throughout the system	Invokes an algorithm created using the **C1_ALTBLLSYS** or **C1_ALTBLSYS1** algorithm type which is attached to the **Sequence Generation Algorithm** field in the **Billing** tab of the **Installation Options** screen.

If the alternate sequential bill numbers must be….	Then the batch…
Unique within the division	Invokes an algorithm created using the **C1_ALTBLLDIV** or **C1_ALTBLDIV1** algorithm type which is attached to the **Sequential Bill Number Generation** algorithm spot of the division.

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Division	No	Used when you want to generate alternate sequential bill numbers for the pending or completed bills of the accounts which belong to a particular division. **Note:** You must specify a division which is already defined in the system.
Process Pending Bills	No	Used to indicate whether you want to generate alternate sequential bill numbers for the pending or completed bills. The valid values are: • **Y** – Used when you want to generate alternate sequential bill numbers for the pending bills. • **N** – Used when you want to generate alternate sequential bill numbers for the completed bills. **Note:** If you do not specify the value for this parameter, by default, it is set to **N**.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Assign Sequential Bill Numbers (ASSGNSBN)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, an alternate sequential bill number is generated for the completed or pending bill and stamped corresponding to the bill in the **CI_BILL** table. A To Do is generated for a bill when error occurs while generating alternate sequential bill number for the bill.

2.16 Generate Credit or Correction Note for Bill (C1-BLCRN)

The **Generate Credit or Correction Note for Bill (C1-BLCRN)** batch is used to generate a credit or correction note for a bill. It is also used to generate a new bill for an account. This batch is a single-threaded batch.

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Bill ID	Yes (Conditional)	Used to indicate the bill for which you want to generate a credit or correction note. **Note:** This parameter is required when the bill generation type is set to **CNON** or **CNBL**.
Account ID	Yes (Conditional)	Used to indicate the account for which you want to generate a new bill. **Note:** This parameter is required when the bill generation type is set to **NBAC**.
Bill Cancel Reason	Yes (Conditional)	Used to indicate the reason why you want to cancel a bill and generate a credit or correction note for the bill. **Points to Note:** You must specify a bill cancel reason which is already defined in the system. This parameter is required when the bill generation type is set to **CNON** or **CNBL**.
Accounting Date	No	Used to specify the date when the new or canceled bill segments' financial transactions should be booked. **Points to Note:** You must specify the date in the YYYY-MM-DD format. If you do not specify any date, the accounting date is set to the batch business date.

Parameter Name	Mandatory (Yes or No)	Description
Credit Note Date	No	Used to specify the date when a bill should be canceled and a credit note should be generated for the bill. It is also used as a bill date when a new bill is generated for an account.
		Points to Note: You must specify the date in the YYYY-MM-DD format. If you do not specify any date, the credit note date is set to the batch business date. This parameter is required when the bill generation type is set to **CNON** or **NBAC**.
Bill Generation Type	No	Used to indicate whether you want to generate a credit or correction note for a bill or a new bill for an account. The valid values are: • **CNON** - Used when you want to generate a credit note for a bill. • **CNBL** - Used when you want to generate a correction note for a bill. • **NBAC** - Used when you want to generate a new bill for an account.
		Note: If you do not specify the value, by default, it is set to **CNON**.
Cutoff Date	No	Used to specify the date till when the billable charges should be considered while creating a correction note for a bill or a new bill for an account.
		Points to Note: You must specify the date in the YYYY-MM-DD format. If you do not specify any date, the cutoff date is set to the batch business date.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Generate Credit or Correction Note for Bill (C1-BLCRN)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, a credit or correction note is generated for the bill or a new bill is generated for the account.

2.17 Bill Cancellation (C1-BILCN)

The **Bill Cancellation (C1-BILCN)** batch is used to cancel all frozen bill segments of a bill at once. It checks whether the specified bill is in the **Complete** status. If the specified bill is in the **Complete** status, it cancels all frozen bill segments of the bill and their corresponding financial transactions. However, if the specified bill is in the **Pending** status, it throws an appropriate error message. To cancel the frozen bill segments of a bill, you need to specify the bill cancel reason which indicates why you want to cancel the frozen bill segments of the bill.

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Bill ID	Yes	Used to indicate the bills whose frozen bill segments you want to cancel. **Note:** If you want to cancel frozen bill segments of multiple bills at once, you can specify comma-separated values for this parameter.
Bill Cycle	No	Used when you want to consider bills of accounts having a particular bill cycle. **Note:** You must specify a bill cycle which is already defined in the system.
Window Start Date	No	Used when you want to consider bills having a particular window start date.
Bill Cancel Reason	Yes	Used to indicate the reason why you want to cancel the frozen bill segments of the bills. **Note:** You must specify a bill cancel reason which is already defined in the system.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Bill Cancellation (C1-BILCN)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of all frozen bill segments of the bills and their corresponding financial transactions is changed to **Canceled**. In addition, the system clears the data from the **BILL_CYC_CD** and **WIN_START_DT** columns of the corresponding records in the **CI_BILL** and **CI_BSEG** tables.

2.18 Bill Deletion (C1BILDEL)

The **Bill Deletion (C1BILDEL)** batch is used to delete a bill and all its bill segments at once. It checks whether the specified bill is in the **Pending** status. If the specified bill is in the **Pending** status, it deletes the bill and all its bill segments. However, if the specified bill is not in the **Pending** status, it throws an appropriate error message.

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Bill ID	Yes	Used to indicate the bills that you want to delete.
		Note: If you want to delete multiple bills at once, you can specify comma-separated values for this parameter.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Bill Deletion (C1BILDEL)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the bill and its corresponding bill segments are deleted from the system.

3.   Contracts

This section provides information about the **Activate or Stop Contracts (SAACT)** batch.

3.1   Activate or Stop Contracts (SAACT)

The **Activate or Stop Contracts (SAACT)** batch is used to do the following:

- Activate the contracts which are in the **Pending Start** status and whose start date is earlier than or equal to the system date
- Stop the contracts which are in the **Pending Stop** status and whose end date is earlier than or equal to the system date

This batch is a multi-threaded batch. The multi-threading is based on contract ID and chunks for multi-threading are created based on numerical distribution of contract ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Activate or Stop Contracts (SAACT)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the contract is:

- Changed from **Pending Start** to **Active** when the contract start date is earlier than or equal to the system date
- Changed from **Pending Stop** to **Stopped** when the contract end date is earlier than or equal to the system date

4.   Deal Pricing Management

This section provides information about the following batches:

- Aggregate Account's Volume and Count (PREAGGR)
- Aggregate Customer's Volume and Count (CSTPREAG)
- Create Hierarchies (C1_HRC)
- Deal Simulation (C1_SMLD)
- Deal Monitoring (C1_DLMTR)

4.1   Aggregate   Account's   Volume   and   Count (PREAGGR)

The **Aggregate Account's Volume and Count (PREAGGR)** batch is used to:

- Aggregate the recurring and non-recurring pass through charges of an account for all deal frequencies (i.e. Monthly, Quarterly, Half Annually, and Annually) and store the information in the **C1_CHARGE_AGGR** table.
- Aggregate the service quantities from the SQI based recurring and non-recurring billable charges of an account for all deal frequencies (i.e. Monthly, Quarterly, Half Annually, and Annually) and store the information in the **C1_VOLUME_AGGR** table.

If you set the **Refer Volume Aggregation** option type of the **C1-DEAL** feature configuration to **Y**, the system extracts the information from the **C1_CHARGE_AGGR** and **C1_VOLUME_AGGR** tables while creating a deal. If you want to extract the information from the **C1_CHARGE_AGGR** and **C1_VOLUME_AGGR** tables for a deal, you must execute the **Aggregate Account's Volume and Count (PREAGGR)** batch before creating the deal for the account.

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. You can specify the following parameters while executing this batch:

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. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Account ID	No	Used when you want to extract the pass through charges and billed service quantities of a particular account.
Division	No	Used when you want to extract the pass through charges and billed service quantities of all accounts which belong to a particular division. **Note:** You must specify a division which is already defined in the system.

Note: If the **Aggregate Account's Volume and Count (PREAGGR)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the pass through charges and billed service quantities of the account are aggregated and stored in the **C1_CHARGE_AGGR** and **C1_VOLUME_AGGR** tables, respectively.

4.2 Aggregate Customer's Volume and Count (CSTPREAG)

The **Aggregate Customer's Volume and Count (CSTPREAG)** batch is used to:

- Aggregate the pass through charges of all accounts of a customer for all deal frequencies (i.e. Monthly, Quarterly, Half Annually, and Annually) and store the information in the **C1_CHARGE_AGGR** table.

- Aggregate the billed service quantities of all accounts of a customer for all deal frequencies (i.e. Monthly, Quarterly, Half Annually, and Annually) and store the information in the **C1_VOLUME_AGGR** table.

If you set the **Refer Volume Aggregation** option type of the **C1-DEAL** feature configuration to **Y**, the system extracts the information from the **C1_CHARGE_AGGR** and **C1_VOLUME_AGGR** tables while creating a deal.

If you want to extract the information from the **C1_CHARGE_AGGR** and **C1_VOLUME_AGGR** tables for a deal, you must execute the following batches in the specified sequence before creating the deal for the customer:

1. **Aggregate Account's Volume and Count (PREAGGR)**
2. **Aggregate Customer's Volume and Count (CSTPREAG)**

This batch is a multi-threaded batch. The multi-threading is based on person ID and chunks for multi-threading are created based on numerical distribution of person ID. You can specify the following parameters while executing this batch:

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. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Person ID	No	Used when you want to aggregate the pass through charges and billed service quantities of all accounts of a particular person.
Division	No	Used when you want to aggregate the pass through charges and billed service quantities of the respective accounts of the persons which belong to a particular division. **Note:** You must specify a division which is already defined in the system.

Note: If the **Aggregate Customer's Volume and Count (CSTPREAG)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the pass through charges and billed service quantities of the respective accounts of a person are aggregated and stored in the **C1_CHARGE_AGGR** and **C1_VOLUME_AGGR** tables, respectively.

4.3   Create Hierarchies (C1_HRC)

The **Create Hierarchies (C1_HRC)** batch is used to monitor or check whether there is any deal in the **Hierarchy Creation Deferred** status. If there is a deal in the **Hierarchy Creation Deferred** status, the system considers its deal version which is in the **Hierarchy Creation Deferred** status. This batch then does the following for the respective deal version:

- Copies the customer, all its accounts, and the customer hierarchy (i.e. its child persons and child persons' accounts), and creates the corresponding prospect persons and accounts in the system.

- Copies the price list assignments and price assignments of the customer, accounts, child persons and child persons' accounts, and assigns the price lists and pricing to the respective prospect persons and accounts in the **Proposed** status.

- Copies the product enrollments of the customer's and child persons' accounts, and enrolls the respective prospect accounts for the products in the **Proposed** status.

- Copies the commitments from the SQI based billable charges of the customer's and child persons' accounts (based on the deal frequency and usage period), and accordingly adds proposed commitments for the respective prospect accounts.

In addition, if you have referenced another customer while creating a deal for an existing customer, the system does the following:

- Copies the price list assignments and price assignments of the referenced customer and assigns the price lists and pricing to the prospect person in the **Proposed** status

- Copies the commitments from the SQI based billable charges of the referenced customer's accounts (based on the deal frequency and usage period), and accordingly adds rolled up proposed commitments for the prospect person

Finally, the system then changes the status of the deal and its deal version to **Pending Simulation**.

This batch is a multi-threaded batch. The multi-threading is based on deal version ID and chunks for multi-threading are created based on numerical distribution of deal version ID. You can specify the following parameters while executing this batch:

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. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Deal ID	No	Used when you want to create the customer hierarchy for a particular deal.
Division	No	Used when you want to create the customer hierarchy for deals which belong to a particular division. **Note:** You must specify a division which is already defined in the system.

Note: If the **Create Hierarchies (C1_HRC)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the customer hierarchy is created for the deal version and the status of the deal and its deal version is changed from **Hierarchy Creation Deferred** to **Pending Simulation**.

4.4   Deal Simulation (C1_SMLD)

The **Deal Simulation (C1_SMLD)** batch is used to monitor or check whether there is any deal in the **Simulation Deferred** status. If there is a deal in the **Simulation Deferred** status, the system considers its deal version which is in the **Simulation Deferred** status. This batch then calculates the following for the respective deal version:

- Original revenue, cost, profit, and profitability (%) at the deal level
- Proposed revenue, cost, profit, and profitability (%) at the deal level
- Profit variation between original and proposed profit at the deal level
- Original revenue, cost, profit, and profitability (%) at the division level when the division level approval is required
- Proposed revenue, cost, profit, and profitability (%) at the division level when the division level approval is required
- Profit variation between original and proposed profit at the division level when the division level approval is required
- Original revenue, cost, profit, and profitability (%) at the product level when the product level approval is required

- Proposed revenue, cost, profit, and profitability (%) at the product level when the product level approval is required
- Profit variation between original and proposed profit at the product level when the product level approval is required
- Original and proposed revenue and cost of each entity in the customer hierarchy
- Revenue variation between original and proposed revenue at the entity level
- Original and proposed cost and revenue from each price item in the price item hierarchy
- Revenue variation between original and proposed revenue at the price item level
- Average price for each price item in the price item hierarchy

Finally, the system then changes the status of the deal and its deal version to **Simulated**.

This batch is a multi-threaded batch. The multi-threading is based on deal version ID and chunks for multi-threading are created based on numerical distribution of deal version ID. You can specify the following parameters while executing this batch:

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. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Deal ID	No	Used when you want to calculate the cost, revenue, and profitability for a particular deal.
Deal Version ID	No	Used when you want to calculate the cost, revenue, and profitability for a particular deal version of a deal.

Note: If the **Deal Simulation (C1_SMLD)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the above listed calculations are done for the deal version and the status of the deal and its deal version is changed from **Simulation Deferred** to **Simulated**.

4.5   Deal Monitoring (C1_DLMTR)

The **Deal Monitoring (C1_DLMTR)** batch is used to monitor or check whether there is any deal in the following status:

- Draft
- Error
- Hierarchy Creation Deferred
- Pending For Approval
- Pending Simulation
- Ready For Approval
- Simulated
- Simulation Deferred
- Fully Orchestrated

The following table describes how the batch behaves when a deal is in a particular status:

If there is a deal in the following status…	Then…
Fully Orchestrated	The system considers its deal version which is in the **Fully Orchestrated** status. This batch then does the following: • Calculates the review date and To Do reminder date for the deal • Create a To Do for deal review and changes the status of the deal to **Pending For Review** when the To Do reminder date is equal to the system date If the deal is not yet reviewed even once, the batch calculates the first review date using the deal effective start date and deal review frequency. For example, if there is a deal which is effective from 07-01-2020 to 12-31-2021 and where the deal review frequency is set to **Biannually**, the system sets the first review date to 01-01-2021 (i.e. 07-01-2020 + 6 months). However, if the deal is already reviewed in the past, the batch calculates the subsequent review date using the last review date and deal review frequency. For example, if there is a deal which is effective from 07-01-2020 to 12-31-2021 and where the last review date is set to 01-01-2021 and the deal review frequency is set to **Biannually**, the system sets the subsequent review date to 07-01-2021 (i.e. 01-01-2021 + 6 months). Note that if the calculated review date is later than the deal effective end date, then the batch sets the review date to the deal effective end date.

If there is a deal in the following status…	Then…
	For example, if there is a deal which is effective from 07-01-2020 to 12-31-2021 and where the last review date is set to 07-01-2021 and the deal review frequency is set to **Biannually**, the system sets the subsequent review date to 12-31-2021. This is because the calculated review date (i.e. 01-01-2023) is later than the deal effective end date (i.e. 12-31-2021). In addition, the batch calculates the To Do reminder date which is review date minus the To Do reminder days. For example, if the review date is 12-31-2021 and To Do reminder days is 60, the system sets the To Do reminder date to 11-01-2021.

Draft, Error, Hierarchy Creation Deferred, Pending For Approval, Pending Simulation, Ready For Approval, Simulated, or Simulation Deferred	The system considers its deal version which is in the **Draft**, **Error**, **Hierarchy Creation Deferred**, **Pending For Approval**, **Pending Simulation**, **Ready For Approval**, **Simulated**, or **Simulation Deferred** status. This batch then does the following:
	
	• Calculates the following dates for the deal:
	
	    o **Deal Inactivation Date** = Last Updated Date + Days to Inactivate Deal
	
	    o **First To Do Reminder Date** = Deal Inactivation Date – To Do Remainder Days
	
	    o **Second To Do Remainder Date** = Deal Inactivation Date – To Do High Priority Days
	
	• Creates two reminder notifications (i.e. To Dos) for the relationship manager indicating that the system will inactivate the deal on a particular date if no actions are performed on it before the deal inactivation date. The reminder notifications are sent when the First To Do Reminder Date or Second To Do Remainder Date is equal to the system date.
	
	• Changes the status of the deal to **Inactive** when the deal inactivation date is equal to the system date.

This batch is a multi-threaded batch. The multi-threading is based on deal version ID and chunks for multi-threading are created based on numerical distribution of deal version ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Division	No	Used when you want to monitor deals which are created for a particular division.

Parameter Name	Mandatory (Yes or No)	Description
		Note: You must specify a division which is already defined in the system.
Customer Segment	No	Used when you want to monitor deals which are created for a particular customer segment.
Customer Tier	No	Used when you want to monitor deals which are created for a particular customer tier.
Deal Type Code	No	Used when you want to monitor deals which are created using a particular deal type. **Note:** You must specify a deal type which is already defined in the system.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Deal Monitoring (C1_DLMTR)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the system does the following:

- If a deal version is in the **Fully Orchestrated** status, the Review Date and To Do Reminder Date is calculated for the deal version, a To Do is created for the deal review (when the To Do reminder date is equal to the system date), and the status of the deal and its deal version is changed from **Fully Orchestrated** to **Pending For Review**.

- If a deal version is in the **Draft**, **Error**, **Hierarchy Creation Deferred**, **Pending For Approval**, **Pending Simulation**, **Ready For Approval**, **Simulated**, or **Simulation Deferred** status, the Deal Inactivation Date, First To Do Reminder Date, and Second To Do Remainder Date is calculated for the deal version, a To Do is created for the deal inactivation reminder (when the First To Do Reminder Date or Second To Do Remainder Date is equal to the system date), and the status of the deal and its deal version is changed to **Inactive** (when the deal inactivation date is equal to the system date).

5.   Deferred Revenue Recognition

This section provides information about the following batches:

- Deferred Revenue Recognition Periodic Monitor (C1-RRSMO)
- Deferred Revenue Recognition Adjustments (C1-RRADJ)

5.1   Deferred Revenue Recognition Periodic Monitor (C1-RRSMO)

The **Deferred Revenue Recognition Periodic Monitor (C1-RRSMO)** batch is used to monitor or check whether there are any deferred revenue recognitions in the **Draft** status. If there is a deferred revenue recognition in the **Draft** status, the system checks whether the recognition schedule generation type in the respective deferred revenue recognition template is set to **Automatic** or **Manual**. If the recognition schedule generation type is set to **Automatic**, the status of the deferred revenue recognition is changed to **In Progress** and the algorithms attached to the **In Progress** status are executed. Once the recognition schedule is successfully created, the status of the deferred revenue recognition is changed to **Active**.

However, if the recognition schedule generation type is set to **Manual**, the status of the deferred revenue recognition remains as **Draft**.

This batch is a multi-threaded batch. The multi-threading is based on deferred revenue recognition ID and chunks for multi-threading are created based on numerical distribution of deferred revenue recognition ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor deferred revenue recognitions which are created using the business objects of a particular maintenance object. **Note:** By default, the parameter value is set to **C1-REVRECSCH**.
Batch Control	No	Used when you want to monitor deferred revenue recognitions whose current status is linked to the **Deferred Revenue Recognition Periodic Monitor (C1-RRSMO)** batch. The valid value is **true**.

Parameter Name	Mandatory (Yes or No)	Description
Business Object	No	Used when you want to monitor deferred revenue recognitions which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Status	No	Used when you want to monitor deferred revenue recognitions which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Deferred Revenue Recognition Periodic Monitor (C1-RRSMO)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the schedule is created for the deferred revenue recognition and all lines of the recognition schedule are added in the **CI_REV_REC_ITEM** table. The status of the deferred revenue recognition is changed to **Active** in the **CI_REV_REC_SCH** table.

A log entry generated for the deferred revenue recognition during the status transition is added in the **CI_REV_REC_SCH_LOG** table. In addition, if an error occurs while creating the recognition schedule, it is recorded in the **CI_REV_REC_SCH_LOG** table.

5.2   Deferred Revenue Recognition Adjustments (C1-RRADJ)

The **Deferred Revenue Recognition Adjustments (C1-RRADJ)** batch is used to monitor or check whether there are any deferred revenue recognitions in the **Active** status. If there is a deferred revenue recognition in the **Active** status, it then checks whether there is any line within the recognition schedule whose recognition date is earlier than or equal to the batch business date. If so, it considers those active deferred revenue recognitions for adjustment generation. If an algorithm of the **C1-REVITMREC** algorithm type is attached to the respective deferred revenue recognition template and the **Create Adjustments (Y or N)** parameter in the algorithm is set to **N**, the system does not consider deferred revenue recognitions for adjustment generation. However, if an algorithm of the **C1-REVITMREC** algorithm type is not attached or if the **Create Adjustments (Y or N)** parameter in the algorithm is set to **Y**, the system creates deferred revenue recognition adjustment using the adjustment type defined in the respective deferred revenue recognition template. Before creating the deferred revenue recognition adjustment, it checks whether the contract for which the deferred revenue recognition is created is cancelled.

If so, the status of the deferred revenue recognition is set to **Canceled** and no more adjustments are created for such deferred revenue recognition. If all lines in the recognition schedule are recognized, the status of the deferred revenue recognition is set to **Closed**.

This batch is a multi-threaded batch. The multi-threading is based on deferred revenue recognition ID and chunks for multi-threading are created based on numerical distribution of deferred revenue recognition ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Deferred Revenue Recognition Template	No	Used when you want to create adjustments for deferred revenue recognitions which are created using a particular deferred revenue recognition template.
		Points to Note:
		You must specify a deferred revenue recognition template which is already defined in the system.
		You can specify comma-separated values for this parameter.
Division	No	Used when you want to create adjustments for deferred revenue recognitions which are created against contracts belonging to a particular division.
		Points to Note:
		You must specify a division which is already defined in the system.
		You can specify comma-separated values for this parameter.
Chunk Size	Yes	Used to specify the number of deferred revenue recognitions you want to execute in each work unit.
		Note: By default, the parameter value is set to **100**.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

> **Note:** If the **Deferred Revenue Recognition Adjustments (C1-RRADJ)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the deferred revenue recognition adjustment is created and the adjustment ID is stamped corresponding to the recognition line in the **CI_REV_REC_ITEM** table. If the contract for which the deferred revenue recognition is created is cancelled, the status of the deferred revenue recognition is changed to **Canceled** in the **CI_REV_REC_SCH** table. If all lines in the recognition schedule are recognized, the status of the deferred revenue recognition is set to **Closed** in the **CI_REV_REC_SCH** table. However, if all lines in the recognition schedule are not yet recognized, the status of the deferred revenue recognition remains as **Active** in the **CI_REV_REC_SCH** table.

A log entry generated for the deferred revenue recognition during the status transition is added in the **CI_REV_REC_SCH_LOG** table. In addition, if an error occurs while creating adjustments, it is recorded in the **CI_REV_REC_SCH_LOG** table.

6. Earnings Credit Rate

This section provides information about the following batches:

- Financial Transaction Distribution Monitor (C1-FTDRD)
- Financial Transaction Distribution (C1-FTDTS)

6.1 Financial Transaction Distribution Monitor (C1-FTDRD)

The **Financial Transaction Distribution Monitor (C1-FTDRD)** batch is used to monitor or check whether there are any financial transaction distribution objects in the **Pending** status. If there is a financial transaction distribution object in the **Pending** status, the system validates the financial transaction distribution object. If the validation is successful, the status of the financial transaction distribution object is changed to **Validated**. Then, depending on the following condition which is met, the status of the financial transaction distribution object is changed from **Validated** to **Expire**, **Write Off**, or **Refund**.

If...	Then...
The credit or debit amount on the financial transaction distribution object is expired	The status of the financial transaction distribution object is changed to **Expire** and the expiration adjustment is created using the expiration adjustment type specified in the respective financial transaction distribution type. Once the expiration adjustment is created, the status of the financial transaction distribution object is changed to **Complete**.
The credit or debit amount on the financial transaction distribution object is less than or equal to the write off threshold amount	The status of the financial transaction distribution object is changed to **Write Off** and the write off adjustment is created using the write off adjustment type specified in the respective financial transaction distribution type. Once the write off adjustment is created, the status of the financial transaction distribution object is changed to **Complete**.
There is excess credit amount on the financial transaction distribution object at the end of the year and the source account belongs to a customer class where is refund is allowed	The status of the financial transaction distribution object is changed to **Refund** and the refund adjustment is created using the refund adjustment type specified in the respective financial transaction distribution type. Once the refund adjustment is created, the status of the financial transaction distribution object is changed to **Complete**.

If none of the above conditions are met, the status of the financial transaction distribution object remains as **Validated**. However, if the validation fails for a financial transaction distribution object, the status of the financial transaction distribution object remains as **Pending**.

This batch is a multi-threaded batch. The multi-threading is based on financial transaction distribution ID and chunks for multi-threading are created based on numerical distribution of financial transaction distribution ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor financial transaction distribution objects which are created using the business objects of a particular maintenance object. **Note:** By default, the parameter value is set to **C1-FTDIST**.
Batch Control	No	Used when you want to monitor financial transaction distribution objects whose current status is linked to the **Financial Transaction Distribution Monitor (C1-FTDRD)** batch. The valid value is **true**.
Financial Transaction Distribution Type	No	Used when you want to monitor financial transaction distribution objects which are controlled through a particular financial transaction distribution type. **Note:** You must specify a financial transaction distribution type which is already defined in the system.
Business Object	No	Used when you want to monitor financial transaction distribution objects which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.

Parameter Name	Mandatory (Yes or No)	Description
Status	No	Used when you want to monitor financial transaction distribution objects which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Financial Transaction Distribution Monitor (C1-FTDRD)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the financial transaction distribution object is changed to **Validated** when its validation is successful. However, if the validation fails, it remains in the **Pending** status.

Depending on the above mentioned condition that is met, the status of the financial transaction distribution object is changed from **Validated** to **Expire**, **Write Off**, or **Refund**. Accordingly, the expiration, write off, or refund adjustment is created. However, if none of the above mentioned conditions are met, the status of the financial transaction distribution object remains as **Validated**.

6.2 Financial Transaction Distribution (C1-FTDTS)

The **Financial Transaction Distribution (C1-FTDTS)** batch is used to monitor or check whether there are any financial transaction distribution objects in the **Validated** status. If there is a financial transaction distribution object in the **Validated** status, the system changes the status of the financial transaction distribution object to **Apply Rule** and executes the algorithm attached to the **Apply Rule** system event on the financial transaction distribution rule version which is effective on the batch business date. Once the credit or debit distribution array of accounts is created, the status of the financial transaction distribution object is changed to **Distributed** and then the algorithm attached to the **Distribute** system event on the financial transaction distribution rule version which is effective on the batch business date is executed.

On the credit or debit amount distribution, two adjustments are created – one against the source contract and another against the bill segment or adjustment's contract to which the amount is distributed. Once the credit or debit amount on the financial transaction distribution object is fully distributed, the status of the financial transaction distribution object is changed to **Complete**. However, if the credit or debit

amount on the financial transaction distribution object is not yet fully distributed, the status of the financial transaction distribution object is changed to **Validated**.

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Account ID	No	Used when you want to distribute the credit or debit amount on the financial transaction distribution objects which are created for a particular account. **Note:** You can specify comma-separated values for this parameter.
Division	No	Used when you want to distribute the credit or debit amount on the financial transaction distribution objects which are created for accounts belonging to a particular division. **Note:** You must specify a division which is already defined in the system.
Business Object	Yes	Used when you want to distribute the credit or debit amount on the financial transaction distribution objects which are created using a particular business object. **Note:** By default, the parameter value is set to **C1-FTDIST**.
Status	Yes	Used when you want to distribute the credit or debit amount on the financial transaction distribution objects which are in a particular status. **Note:** By default, the parameter value is set to **VALIDATED**.
Next Status	Yes	Used to indicate the next status to which the financial transaction distribution objects must be transitioned. **Note:** By default, the parameter value is set to **APPLYRULE**.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Parameter Name	Mandatory (Yes or No)	Description
Credit Objects First (Y/N)	No	Used to determine whether the credit or debit financial transaction distribution objects should be distributed first. The valid values are: • **Y** – Used when you want to distribute the credit financial transaction distribution objects first. • **N** – Used when you want to distribute the debit financial transaction distribution objects first. **Note:** By default, the parameter value is set to **N**.
Order processing based on Expiration Date (Y/N)	No	Used to determine the order in which the financial transaction distribution objects should be processed. The valid values are: • **Y** – Used when you want to process the financial transaction distribution objects based on the expiration date. • **N** – Used when you want to process the financial transaction distribution objects based on the creation date. **Note:** By default, the parameter value is set to **N**.

Note: If the **Financial Transaction Distribution (C1-FTDTS)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, two adjustments are created when the credit or debit amount on the financial transaction distribution object is distributed against a bill segment or an adjustment. One adjustment is created against the source contract and another adjustment is created against the bill segment or adjustment's contract to which the amount is distributed.

If the credit or debit amount on the financial transaction distribution object is fully distributed, the status of the financial transaction distribution object is changed to **Complete**. However, if the credit or debit amount on the financial transaction distribution object is not yet fully distributed, the status of the financial transaction distribution object is changed to **Validated**.

7.   Entity Audit

This section provides information about the **Audit Event Processing (C1-AUDEV)** batch.

7.1   Audit Event Processing (C1-AUDEV)

The **Audit Event Processing (C1-AUDEV)** batch is used to check whether there are any audit events in the **Pending** status. If there is an audit event in the **Pending** status, the system checks whether an algorithm is attached to the **Audit Event Processing** system event of the respective audit event type. If an algorithm is not attached to the **Audit Event Processing** system event of the respective audit event type, the status of the audit event is set to **Complete**.

However, if an algorithm is attached to the **Audit Event Processing** system event of the respective audit event type, the system executes the algorithm and accordingly changes the status of the audit event to **Complete**. For example, while processing the audit events of the **C1-PricingRuleTierBased** and **C1-PricingRuleAgeBased** business objects, the system checks whether an algorithm is attached to the **Audit Event Processing** system event of the respective audit event type. If the **C1-AUDEVMPR** algorithm is attached to the audit event type, the system identifies the policy plan where the pricing rule is defined. Once the policy plan is identified, it extracts a list of membership defined on the policy plan and the pricing rule type using which the pricing rule is created. Then, the system creates an entry for each membership, pricing rule type, and effective date combination in the **CI_REPRC_ENTITY_DTL** table. The status of these entries is set to **P**. Finally, the status of the audit event is set to **Complete**.

If any error occurs, the status of the audit event is set to **Error**. This batch is a multi-threaded batch. The multi-threading is based on audit event ID and chunks for multi-threading are created based on numerical distribution of audit event ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	No	Used when you want to monitor and process audit events which are created for business objects of a particular maintenance object. **Note:** You must specify a maintenance object which is already defined in the system.
Audit Event Type	No	Used when you want to monitor and process audit events which are created using a particular audit event type. **Note:** You must specify an audit event type which is already defined in the system.

Parameter Name	Mandatory (Yes or No)	Description
Audit Event Business Object	No	Used when you want to monitor and process audit events which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Audit Event Error Status	No	Used when you want monitor and process audit events in the **Error** status.
Audit Event Usage	No	Used when you want to monitor and process audit events which are created for a particular usage. **Note:** You must specify a value which is already defined in the **AUDIT_USAGE_FLG** lookup field.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Audit Event Processing (C1-AUDEV)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the audit event is changed to either **Complete** or **Error** in the **C1_AUDIT_EVENT** table.

8.   Financial Transactions

This section provides information about the following batches:

- [Foreign Exchange Loss Gain (C1-FXLG)](#)
- [Assign GL Account to Financial Transaction (C1-GLASN)](#)
- [Insert Records in CI_FTTEMP (GLASSGN1)](#)
- [Assign GL Account to Financial Transaction (GLASSGN2)](#)
- [GL Download Staging (GLS)](#)
- [GL Download Extract (GLDL)](#)
- [General Ledger Data Extract (C1-GLS)](#)
- [General Ledger Data Download (C1-GLDL)](#)

8.1   Foreign Exchange Loss Gain (C1-FXLG)

The **Foreign Exchange Loss Gain (C1-FXLG)** batch is used to calculate foreign exchange gain or loss on financial transactions, such as payments and adjustments. This gain or loss occurs due to fluctuations in the exchange rate at different point in time. During the batch execution, the system considers the following types of financial transactions:

- Payments (i.e. Pay Segments)
- Credit adjustments which are created against a bill after the bill completion
- Write-off adjustments which are created against a bill
- Write Up adjustments which are automatically created when you match a payment against a bill using the Bill Weighted match type
- Write Down adjustments which are automatically created when you match a payment against a bill using the Bill Weighted match type

However, the system considers payment and adjustment FTs when:

- Division's Base Currency is different from the Payment or Adjustment Currency
- Payment or Adjustment Freeze Date is different from the Bill Completion Date
- Match Event ID exists in the financial transaction
- **FXLG_CALC_STATUS** column corresponding to the financial transaction in the **CI_FT** table is set to **N** or **NULL**

This batch is a multi-threaded batch. The multi-threading is based on financial transaction ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Division	Yes	Used to calculate foreign exchange gain or loss for financial transactions, which are created for accounts, belonging to a particular division.

Parameter Name	Mandatory (Yes or No)	Description
		Note: You must specify a division which is already defined in the system.
Distribution Code for FXLG	Yes	Used to indicate the distribution code against which you want to book the foreign exchange gain or loss.
		Note: You must specify a distribution code which is already defined in the system.
Distribution Code for Contract Balancing	No	Used to indicate the distribution code against which you want to balance the foreign exchange gain or loss.
		Points to Note: You must specify a distribution code which is already defined in the system. If you do not specify any distribution code, by default, the distribution code associated with the FT's contract is used to balance the foreign exchange gain or loss.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

You must execute the **Foreign Exchange Loss Gain (C1-FXLG)** batch prior to the **Assign GL Account to Financial Transaction (C1-GLASN)** or **Insert Records in CI_FTTEMP (GLASSGN1)** batch.

If the **Foreign Exchange Loss Gain (C1-FXLG)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, two entries are added corresponding to each financial transaction (for which foreign exchange gain or loss is calculated) in the **CI_FT_GL** and **CI_FT_GL_EXT** tables. One entry for foreign exchange gain or loss is booked against the distribution code for FXLG and another entry for foreign exchange gain or loss is booked against the distribution code which is used for contract balancing. The foreign exchange loss is recorded as positive entry and the foreign exchange gain is recorded as negative entry.

In the **CI_FT_GL** table, the foreign exchange gain or loss is recorded in the financial transaction currency. And, in the **CI_FT_GL_EXT** table, the foreign exchange gain or loss is recorded in the division's base currency. The **FXLG_CALC_STATUS** column corresponding to the financial transaction in the **CI_FT** table is set to **I**.

8.2 Assign GL Account to Financial Transaction (C1-GLASN)

The **Assign GL Account to Financial Transaction (C1-GLASN)** batch is used to assign GL account to the trial or actual financial transactions. It allows you to validate the GL account before it is assigned to the financial transaction. If you opt to validate GL account, the batch assigns the GL account to the financial transaction only when it is valid. The date when the GL account is validated is then stored corresponding to the record in the **GLA_VAL_DT** column of the **CI_TRL_FT_GL** or **CI_FT_GL** table, respectively, depending on whether the GL account is assigned to a trial or actual financial transaction. However, if the GL account is invalid or if the GL account could not be validated successfully, the batch does not assign the GL account to the financial transaction. Also, the **GLA_VAL_DT** column is not updated in the respective table.

This batch facilitates you to assign GL account only to frozen or to both frozen and freezable financial transactions depending on the business requirement.

Points to Note:

If the GL account is not successfully assigned any financial transaction, an error is logged and To Do is created for the corresponding financial transaction.

For more information about GL account validation, see the **Dynamic GL Account Validation** section in the *Oracle Revenue Management and Billing Banking User Guide*.

This batch is a multi-threaded batch. The multi-threading is based on financial transaction ID and chunks for multi-threading are created based on numerical distribution of financial transaction ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Process Only Frozen FTs	No	Used to indicate whether you want to assign GL account only to frozen financial transactions. The valid value is **Y**. If you set the parameter value to **Y**, the batch will assign GL account only to frozen financial transactions. However, if you specify any value other than **Y** or if you leave this parameter blank, the batch will assign GL account to both frozen and freezable financial transactions.
Chunk Size	No	Used to specify the number of financial transactions to which you want to assign GL account in each work unit. **Note:** If you do not specify any value, by default, it is set to 100.

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.
Division or Division Group	No	Used when you want to assign GL account to financial transactions, which are created for accounts, belonging to a particular division or division group.
		Points to Note:
		You must specify a division or division group which is already defined in the system.
		If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division.
		If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.
Bill Generation Type	No	Used to indicate whether you want to assign GL account to the trial or actual financial transactions. The valid values are:
		• Trial
		• Regular
		Note: If you do not specify any value, by default, the bill generation type is set to **Regular**.
Trial Billing Batch Run Number	No	Used to indicate the trial billing batch run number whose trial bills' financial transactions you want to consider for assigning GL account.
		Note: This parameter is used only when you are assigning GL account to the trial financial transactions.

Parameter Name	Mandatory (Yes or No)	Description
Validate GL Account	No	Used to indicate whether you want to validate the GL account before it is assigned to the financial transaction. The valid value is **Y**. If you set the parameter value to **Y**, the batch will validate the GL account. However, if you specify any value other than **Y** or if you leave this parameter blank, the batch will not validate the GL account.
Create Additional FT GL Entries	No	Used to indicate whether you want to create an FT GL entry for a financial transaction using the GL accounting template. The valid value is **Y**. If you set the parameter value to **Y**, the batch will invoke the **C1-GLCE** algorithm to create an FT GL entry for a financial transaction using the GL accounting template. In addition, if the financial transaction's currency is different from the division's base currency and/or division's local currency, the algorithm will create additional FT GL entries (i.e., FT GL extension records) for a financial transaction in the **CI_FT_GL_EXT** table. However, if you specify any value other than **Y** or if you leave this parameter blank, the batch will create an FT GL entry for a financial transaction in the traditional method (i.e., using the distribution code from the adjustment type, rate component, billable charge pass through line, or contract type depending on the type of financial transaction). For more information, see the **Additional FT GL Entry Creation on the C1-GLASN or GLASSGN2 Batch Execution** section in the *Oracle Revenue Management and Billing Business Process Guide*.

Parameter Name	Mandatory (Yes or No)	Description
Financial Transaction Type	No	Used to indicate whether you want to assign GL account to a particular type of financial transactions. The valid values are: • AD • AX • BS • BX • PS • PX **Points to Note:** You can specify comma-separated values for this parameter. If you do not specify any value, by default, the batch assigns GL account to all the financial transactions irrespective of their type.

Note: If the **Assign GL Account to Financial Transaction (C1-GLASN)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the following columns are updated in the **CI_TRL_FT_GL** or **CI_FT_GL** table depending on whether the GL account is assigned to a trial or actual financial transaction:

• GL_ACCT
• GLA_VAL_DT
• VALIDATE_SW

Note: The **GLA_VAL_DT** column is updated only when the GL account is valid and not when the GL account is invalid.

8.3   Insert Records in CI_FTTEMP (GLASSGN1)

The **Insert Records in CI_FTTEMP (GLASSGN1)** batch is used to search and add the frozen and/or freezable financial transactions (FTs), where the GL account is not yet assigned, in the **CI_FTTEMP** table. The data in this table is then used for creating chunks for multi-threading while executing the **Assign GL Account to Financial Transaction (GLASSGN2)** batch.

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Process Only Frozen FTs	No	Used to indicate whether you want to add only frozen financial transactions in the **CI_FTTEMP** table. The valid value is **Y**. If you set the parameter value to **Y**, the batch will add only frozen financial transactions in the **CI_FTTEMP** table. However, if you specify any value other than **Y** or if you leave this parameter blank, the batch will add both frozen and freezable financial transactions in the **CI_FTTEMP** table.
Division	No	Used when you want to add financial transactions, which are created for accounts, belonging to a particular division. **Note:** You must specify a division which is already defined in the system.
Bill Generation Type	No	Used to indicate whether you want to add trial or actual financial transactions in the **CI_FTTEMP** table. The valid values are: • Trial • Regular **Note:** If you do not specify any value, by default, the bill generation type is set to **Regular**.

Parameter Name	Mandatory (Yes or No)	Description
Trial Billing Batch Run Number	No	Used to indicate the trial billing batch run number whose trial bills' financial transactions you want to add in the **CI_FTTEMP** table. **Note:** This parameter is used only when you are adding the trial financial transactions in the **CI_FTTEMP** table.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

You must specify the same set of parameters in the **GLASSGN1** and **GLASSGN2** batches. Otherwise, erroneous results might occur.

If the **Insert Records in CI_FTTEMP (GLASSGN1)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the financial transactions from the **CI_TRL_FT_GL** or **CI_FT_GL** table are added in the **CI_FTTEMP** table.

8.4   Assign GL Account to Financial Transaction (GLASSGN2)

The **Assign GL Account to Financial Transaction (GLASSGN2)** batch is used to assign GL account to frozen and/or freezable financial transactions which are stored in the **CI_FTTEMP** table. It allows you to validate the GL account before it is assigned to the financial transactions. If you opt to validate GL account, the batch assigns the GL account to the financial transaction only when it is valid. The date when the GL account is validated is then stored corresponding to the record in the **GLA_VAL_DT** column of the **CI_TRL_FT_GL** or **CI_FT_GL** table, respectively, depending on whether the GL account is assigned to a trial or actual financial transaction. However, if the GL account is invalid or if the GL account could not be validated successfully, the batch does not assign the GL account to the financial transaction. Also, the **GLA_VAL_DT** column is not updated in the respective table.

Points to Note:

If the GL account is not successfully assigned any financial transaction, an error is logged and To Do is created for the corresponding financial transaction.

For more information about GL account validation, see the **Dynamic GL Account Validation** section in the *Oracle Revenue Management and Billing Banking User Guide*.

This batch is a multi-threaded batch. The multi-threading is based on financial transaction ID and chunks for multi-threading are created based on numerical distribution of financial transaction ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Process Only Frozen FTs	No	Used to indicate whether you want to assign GL account only to frozen financial transactions. The valid value is **Y**. If you set the parameter value to **Y**, the batch will assign GL account only to frozen financial transactions. However, if you specify any value other than **Y** or if you leave this parameter blank, the batch will assign GL account to both frozen and freezable financial transactions.
Division	No	Used when you want to assign GL account to financial transactions, which are created for accounts, belonging to a particular division. **Note:** You must specify a division which is already defined in the system.
Bill Generation Type	No	Used to indicate whether you want to assign GL account to the trial or actual financial transactions. The valid values are: • Trial • Regular **Note:** If you do not specify any value, by default, the bill generation type is set to **Regular**.
Trial Billing Batch Run Number	No	Used to indicate the trial billing batch run number whose trial bills' financial transactions you want to consider for assigning GL account.

Parameter Name	Mandatory (Yes or No)	Description
		Note: This parameter is used only when you are assigning GL account to the trial financial transactions.
Validate GL Account	No	Used to indicate whether you want to validate the GL account before it is assigned to the financial transaction. The valid value is **Y**.
		If you set the parameter value to **Y**, the batch will validate the GL account. However, if you specify any value other than **Y** or if you leave this parameter blank, the batch will not validate the GL account.
Create Additional FT GL Entries	No	Used to indicate whether you want to create an FT GL entry for a financial transaction using the GL accounting template. The valid value is **Y**.
		If you set the parameter value to **Y**, the batch will invoke the **C1-GLCE** algorithm to create an FT GL entry for a financial transaction using the GL accounting template. In addition, if the financial transaction's currency is different from the division's base currency and/or division's local currency, the algorithm will create additional FT GL entries (i.e., FT GL extension records) for a financial transaction in the **CI_FT_GL_EXT** table.
		However, if you specify any value other than **Y** or if you leave this parameter blank, the batch will create an FT GL entry for a financial transaction in the traditional method (i.e., using the distribution code from the adjustment type, rate component, billable charge pass through line, or contract type depending on the type of financial transaction).
		For more information, see the **Additional FT GL Entry Creation on the C1-GLASN or GLASSGN2 Batch Execution** section in the *Oracle Revenue Management and Billing Business Process Guide*.

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. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

You must specify the same set of parameters in the **GLASSGN1** and **GLASSGN2** batches. Otherwise, erroneous results might occur.

If the **Assign GL Account to Financial Transaction (GLASSGN2)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the following columns are updated in the **CI_TRL_FT_GL** or **CI_FT_GL** table depending on whether the GL account is assigned to a trial or actual financial transaction:

- GL_ACCT
- GLA_VAL_DT
- VALIDATE_SW

Note: The **GLA_VAL_DT** column is updated only when the GL account is valid and not when the GL account is invalid.

8.5   GL Download Staging (GLS)

The **GL Download Staging (GLS)** batch is used to mark the financial transaction (FT) records which are eligible for extraction so that they can be posted to the General Ledger application for final accounting and reporting. Note that the batch considers only those FT records against which the GL account is assigned.

Here, marking for extraction means stamping the batch control and the batch control's current batch run number on the FT record. This batch stamps the batch control that is specified in the **G/L Batch Code** field of the installation options. Once the FT records are marked, you can then extract the eligible FT records in a flat file using the **GLDL** batch. For more information, see GL Download Extract (GLDL).

This batch is a multi-threaded batch. The multi-threading is based on financial transaction ID and chunks for multi-threading are created based on numerical distribution of financial transaction ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.

Parameter Name	Mandatory (Yes or No)	Description
Division	No	Used when you want to mark the FT records of accounts belonging to a particular division for extraction.
		Points to Note: You must specify a division which is already defined in the system. You can specify maximum twenty comma-separated values for this parameter.
Bill Generation Type	No	Used to indicate whether you want to mark the trial or actual FT records for extraction. The valid values are: • Trial • Regular
		Note: If you do not specify any value, by default, the bill generation type is set to **Regular**.
Trial Billing Batch Run Number	No	Used to indicate the trial billing batch run number whose trial bills' FT records you want to mark for extraction.
		Note: This parameter is used only when you want to mark the trial FT records for extraction.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

You must specify the same set of parameters (for example, division, bill generation type, and trial billing batch run number) including the batch business date in the **C1-GLASN**, **GLS**, and **GLDL** batches during the batch execution cycle. Otherwise, erroneous results might occur.

If the **GL Download Staging (GLS)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

8.6   GL Download Extract (GLDL)

The **GL Download Extract (GLDL)** batch is used to extract the trial or actual FT records depending on the specified bill generation type in a flat file. It extracts the data for the FT records whose entry exists in the **CI_TRL_FT_PROC** or **CI_FT_PROC** table with the given batch run number. If the batch run number is not specified during the batch execution, the batch extracts the data for the FT records whose entry exists in the **CI_TRL_FT_PROC** or **CI_FT_PROC** table with the current batch run number. The flat file is then used for uploading data in the General Ledger application.

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Output Directory Path	Yes	Used to specify the path where you want to save the flat file.
Output File Name	Yes	Used to specify the name that you want to use for the flat file.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Division	No	Used when you want to extract FT records of accounts belonging to a particular division. **Points to Note:** You must specify a division which is already defined in the system. You can specify maximum twenty comma-separated values for this parameter.
Bill Generation Type	No	Used to indicate whether you want to extract the trial or actual FT records. The valid values are: • Trial • Regular **Note:** If you do not specify any value, by default, the bill generation type is set to **Regular**.

Parameter Name	Mandatory (Yes or No)	Description
Trial Billing Batch Run Number	No	Used to indicate the trial billing batch run number whose trial bills' FT records you want to extract.
		Note: This parameter is used only when you want to extract the trial FT records.
GLDL Insert Switch	No	Used to indicate whether you want to insert the extracted records in the **C1_GL_DL_HDR** and **C1_GL_DL_DTL** tables. The valid values are:
		• Y
		• N
		If you set the parameter value to **Y**, the batch will insert the extracted records in the above tables along with the flat file. However, if you set the parameter value to **N** or if you leave this parameter blank, the batch will not insert the extracted records in the **C1_GL_DL_HDR** and **C1_GL_DL_DTL** tables. The system will then extract the records only in the flat file.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

You must specify the same set of parameters (for example, division, bill generation type, and trial billing batch run number) including the batch business date in the **C1-GLASN**, **GLS**, and **GLDL** batches during the batch execution cycle. Otherwise, erroneous results might occur.

If the **GL Download Extract (GLDL)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the FT records are extracted in a flat file at the specified location. If the **GLDL Insert Switch** parameter is set to **Y**, the FT records are also extracted in the **C1_GL_DL_HDR** and **C1_GL_DL_DTL** tables.

8.7 General Ledger Data Extract (C1-GLS)

The **General Ledger Data Extract (C1-GLS)** batch is used to mark the financial transaction (FT) records which are eligible for extraction so that they can be posted to the General Ledger application for final accounting and reporting. Note that the batch considers only those FT records against which the GL account is assigned. You can opt to mark the FT records for extraction based on the scheduled distribution date. The system will then consider and mark only those FT records for extraction whose scheduled distribution date is on or before the specified date.

Here, marking for extraction means stamping the batch control and the batch control's current batch run number on the FT record. This batch stamps the batch control that is specified in the **G/L Batch Code** field of the installation options. Once the FT records are marked, you can then extract the eligible FT records in the required format using the **C1-GLDL** batch. For more information, see General Ledger Data Download (C1-GLDL).

This batch is a multi-threaded batch. The multi-threading is based on financial transaction ID and chunks for multi-threading are created based on numerical distribution of financial transaction ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Division or Division Group	No	Used when you want to mark the FT records of accounts belonging to a particular division or division group for extraction. **Points to Note:** You must specify a division or division group which is already defined in the system. If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division.

Parameter Name	Mandatory (Yes or No)	Description
		If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.
Bill Generation Type	No	Used to indicate whether you want to mark the trial or actual FT records for extraction. The valid values are: • Trial • Regular **Note:** If you do not specify any value, by default, the bill generation type is set to **Regular**.
Trial Billing Batch Run Number	No	Used to indicate the trial billing batch run number whose trial bills' FT records you want to mark for extraction. **Note:** This parameter is used only when you want to mark the trial FT records for extraction.
Filter By Bill Due Date	No	Used to indicate whether to mark FT records for extraction based on the scheduled distribution date. The valid values are: • **Y** – Used when you want to mark FT records for extraction based on the scheduled distribution date. • **N** - Used when you want to mark all the FT records that meet the criteria for extraction irrespective of the scheduled distribution date. **Note:** If you do not specify any value, by default, it is set to **N**.
Bill Due Date	No	Used to specify the date in the **YYYY-MM-DD** format. The batch will then mark only those FT records for extraction whose scheduled distribution date is on or before the specified date. **Points to Note:** This parameter is used only when you set the **Filter By Bill Due Date** parameter to **Y**.

Parameter Name	Mandatory (Yes or No)	Description
		If you do not specify any value, by default, it is set to the batch business date.
FT Type Flag	No	Used to indicate whether you want to mark FT records of a particular type for extraction. The valid values are: • BS • BX • AD • AX • PS • PX **Points to Note:** You can specify comma-separated values for this parameter. If you do not specify any value, by default, the batch will mark the FT records for extraction irrespective of their type.
Chunk Size	No	Used to specify the number of FT records you want to process in each work unit. **Note:** If the specified value is less than 2 or if you do not specify any value, by default, the chunk size is set to 2.
Parallel Query Degree	No	Used to specify the degree of parallelism that you want to use while creating chunks (i.e., work units). The valid values are: • 1 • 2 • 3 • 4 • 5 • 6 • 7 **Points to Note:** With the parallel query feature, multiple threads on the database can work together simultaneously to process a single SQL statement which is used to

Parameter Name	Mandatory (Yes or No)	Description
		create chunks. This helps to improve the batch performance.
		The system calculates the number of threads on the database that can be used to execute the SQL statement by setting 2 to the power of the specified parallel query degree. For example, if you set the parallel query degree to 4, the system uses 16 (i.e., 2^4) threads on the database to create work units.
		You must set the parallel query degree depending on the infrastructure available on the site. If you do not specify the value for this parameter, by default, the parallel query degree is set to 4.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.
Email Mode	No	Used to indicate whether you want to send the email notification on the batch job completion depending on the batch job run status. The valid values are: • **ERROR** – Used when you want to send the email notification on the batch job completion only when the batch job run is in the **Error** status. • **SUCCESS** - Used when you want to send the email notification on the batch job completion only when the batch job run is in the **Complete** status. • **ALL** - Used when you want to send the email notification on the batch job completion irrespective of the batch job run status.
		Points to Note: If you do not specify any value, by default, it is set to **ALL**.

Parameter Name	Mandatory (Yes or No)	Description
		This parameter is applicable only when the email address is specified in the batch job.

Points to Note:

You must specify the same set of parameters in the **C1-GLASN**, **C1-GLS**, and **C1-GLDL** batches during the batch execution cycle. Otherwise, erroneous results might occur.

If the **General Ledger Data Extract (C1-GLS)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the **GL_DISTRIB_STATUS** column corresponding to the FT record in the **CI_TRL_FT** or **CI_FT** table is set to **D** (i.e., Distributed). In addition, the **XFER_TO_GL_DT** column corresponding to the FT record in the **CI_TRL_FT** or **CI_FT** table is set to the batch business date. A new record for the FT is added in the **CI_TRL_FT_PROC** or **CI_FT_PROC** table, and the batch control and the batch control's current batch run number is stamped corresponding to the record.

8.8   General Ledger Data Download (C1-GLDL)

The **General Ledger Data Download (C1-GLDL)** batch is used to extract the trial or actual FT records depending on the specified bill generation type in the required file format. It extracts the data for the FT records whose entry exists in the **CI_TRL_FT_PROC** or **CI_FT_PROC** table with the given batch run number. If the batch run number is not specified during the batch execution, the batch extracts the data for the FT records whose entry exists in the **CI_TRL_FT_PROC** or **CI_FT_PROC** table with the current batch run number. The extracted file is then used for uploading data in the General Ledger application.

Note: The batch invokes an algorithm created using the **C1-GLPROCEX** algorithm type attached to the **Process Record** system event. This algorithm extracts the required information of the financial transaction using the **C1-FTGLPROCRECEX** data area schema. You can attach a custom algorithm to the **C1-GLDL** batch if you want to extract the data in the custom format.

This batch is a multi-threaded batch. The multi-threading is based on financial transaction ID and chunks for multi-threading are created based on numerical distribution of financial transaction ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Output Directory Path	Yes	Used to specify the path where you want to save the file.
		Points to Note: Here, you must specify a sub-directory which already exists in the $SPLOUTPUT location. If you do not specify any value, by default, it is set to $SPLOUTPUT.

Parameter Name	Mandatory (Yes or No)	Description
Output File Name	Yes	Used to specify the name that you want to use for the file.
		Note: If the specified file already exists in the given location, the batch will replace the existing file with the new file. However, if the specified file does not exist in the given location, the batch will create the new file with the given name.
File Format	No	Used to indicate the format in which you want to extract the data. The valid values are:
		• **CSV** – Used when you want to extract the data in the **Comma-Separated Values (CSV)** format.
		• **FIXD** – Used when you want to extract the data in the **Fixed** format (often a plain text file, potentially with a **.txt** extension).
		• **XML** – Used when you want to extract the data in the **eXtensible Markup Language (XML)** format.
		Note: If you do not specify any value, the batch will determine the file format from the given file extension.
XML Root Name	No	Used to specify the name of the root element for an XML file.
		Note: If you do not specify any value, by default, it is set to **root**.
CSV File Delimiter	No	Used to specify the delimiter for a CSV file when you want to use something else other than the **Comma** (,) symbol.
		Points to Note:
		You can specify any single special character except double quotes.
		You can specify the character itself (for example, |) or the Unicode notation of the character in the \uhhhh format (for example, \u0020 for a space).

Parameter Name	Mandatory (Yes or No)	Description
Character Encoding	No	Used to indicate the character encoding format that you want to use for the file. **Note:** If you do not specify any value, by default, it is set to **UTF-8**.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Division or Division Group	No	Used when you want to extract the FT records of accounts belonging to a particular division or division group. **Points to Note:** You must specify a division or division group which is already defined in the system. If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division. If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.
Bill Generation Type	No	Used to indicate whether you want to extract the trial or actual FT records. The valid values are: • Trial • Regular **Note:** If you do not specify any value, by default, the bill generation type is set to **Regular**.

Parameter Name	Mandatory (Yes or No)	Description
Trial Billing Batch Run Number	No	Used to indicate the trial billing batch run number whose trial bills' FT records you want to extract.
		Note: This parameter is used only when you want to extract the trial FT records.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

You must specify the same set of parameters in the **C1-GLASN**, **C1-GLS**, and **C1-GLDL** batches during the batch execution cycle. Otherwise, erroneous results might occur.

If the **General Ledger Data Download (C1-GLDL)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the FT records are extracted in the required file format at the specified location.

9.   FOP Reports

This section provides information about the **Report Generation (C1-RPTGN)** batch.

9.1   Report Generation (C1-RPTGN)

The **Report Generation (C1-RPTGN)** batch is used to generate an FOP report which is already defined in the system. It is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Report Code	Yes	Used to specify the FOP report that you want to generate. **Note:** Here, you must specify an FOP report where the reporting mode is set to **Batch**.
Batch Report Zone	Yes	Used to specify the query zone designed for the respective report. The batch then generates separate output files based on the query defined in the zone. **Points to Note:** In the query zone, you can define a SELECT statement to retrieve data from a table or view based on a condition. You can use the bind variables in the SELECT statement. The bind variables must be alphanumeric and prefixed with the **Colon** (:) symbol. The values for these bind variables can be either defaulted in the query or specified while executing the batch. A sample query named **C1_FOP_DIV** is shipped with the product for your reference. You can design and use the query zone specific to your business requirement for each FOP report.
Binding Parameter Names used in the Zone	No	Used to specify the bind variables that are used in the specified query zone. **Note:** If you want to specify multiple bind variables, you need to separate the bind variables using the **Tilde (~)** symbol. For example, CIS_DIVISION~START_DATE.

Parameter Name	Mandatory (Yes or No)	Description
Binding Parameter Values for the above Names	No	Used to specify value for the bind variables used in the specified query zone.
		Points to Note: If you want to specify multiple bind variable values, you need to separate the bind variable values using the **Tilde (~)** symbol. For example, CALIFORNIA~01-JAN-2000. You must specify the date value in the **DD-MON-YEAR** format.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Report Generation (C1-RPTGN)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the report is generated in the predefined format and placed at the location that is specified in the **Report Generation (FOP) File Path** option type of the **C1-REPORTVW** feature configuration.

Note: If the sender email is configured in the **Email Sender Configuration** master configuration, the generated reports are also emailed to the email address specified in the respective FOP report.

10. Fully-Insured Billing

This section provides information about the **Fully-Insured Billable Charge Creation (C1-FIBCR)** batch.

10.1 Fully-Insured Billable Charge Creation (C1-FIBCR)

The **Fully-Insured Billable Charge Creation (C1-FIBCR)** batch is used to create SQI based billable charge for calculated premium amount. It considers the timelines in the **Pending (P)** and **Error (E)** status from the **CI_PRCE_CALC** table. Note that it considers only those timelines where the entity type is set to **Membership**. For each timeline, membership and price item combination, the system does the following:

- Derives the bill period for the account using the **Invoice Day (C1OFFRST)** characteristic defined on the account.

- Creates an SQI based billable charge. Here, the start and end dates on the billable charge are set to the timeline start and end dates.

If the premium amount of the two consecutive timelines of a membership and price item combination is same, this algorithm does not create a new billable charge. Instead, it extends the end date of the previous timeline's billable charge. If the membership end date is set to a date which is earlier than the previous date (for example, Old Membership End Date: 31-12-2019 and New Membership End Date: 31-10-2019) and a billable charge exists for the price item and membership combination with the same start date and amount, then the system behaves in the following manner:

If…	Then…
Bill segments for the 01-01-2019 to 31-12-2019 period are not yet generated in the system	The end date in the existing billable charge is updated to 31-10-2019.
Bill segments for the 01-01-2019 to 31-12-2019 period are in the **Freezable** status	The bill segments for the 01-11-2019 to 31-12-2019 period are deleted and the end date in the existing billable charge is updated to 31-10-2019.
Bill segments for the 01-01-2019 to 31-12-2019 period are in the **Frozen** or **Pending Cancel** status	The bill segments for the 01-11-2019 to 31-12-2019 period are canceled and the end date in the existing billable charge is updated to 31-10-2019.

However, the membership end date is set to a date which is earlier than the previous date (for example, Old Membership End Date: 31-12-2019 and New Membership End Date: 31-10-2019) and a billable charge exists for the price item and membership combination with the same start date, but different amount, then the system behaves in the following manner:

If…	Then…
Bill segments for the 01-01-2019 to 31-12-2019 period are not yet generated in the system	The existing billable charge is canceled and a new billable charge is created for the 01-01-2019 to 31-10-2019 period.

If…	Then…
Bill segments for the 01-01-2019 to 31-12-2019 period are in the **Freezable** status	The bill segments for the 01-01-2019 to 31-12-2019 period are deleted. The existing billable charge is canceled and a new billable charge is created for the 01-01-2019 to 31-10-2019 period.
Bill segments for the 01-01-2019 to 31-12-2019 period are in the **Frozen** or **Pending Cancel** status	The bill segments for the 01-01-2019 to 31-12-2019 period are canceled. The existing billable charge is canceled and a new billable charge is created for the 01-01-2019 to 31-10-2019 period.

And, if the membership end date is set to a date which is earlier than the previous date (for example, Old Membership End Date: 31-12-2019 and New Membership End Date: 31-10-2019) and a billable charge does not exist for the price item and membership combination with the same start date, then a billable charge is created for the 01-01-2019 to 31-10-2019 period. Note that if there are billable charges for the price item and membership combination whose date range is within the timeline, such billable charges are canceled in the system and their corresponding bill segments (if any) are either deleted or canceled.

The pre-calculated or calculated premium amount is stored as a pass through line in the SQI based billable charge. Note that these pass through lines are memo only and no general ledger entries will be created for these pass through lines. This algorithm fetches SQI from the **Proration SQI** option type of the **C1-ASOBLLNG** feature configuration. The SQI value is set to the total amount of the pass through lines in the billable charge. The system prorates the amount only when the SQ rule specified in the rate schedule attached to the active contract contains the SQI which is specified in the **Proration SQI** option type of the **C1-ASOBLLNG** feature configuration.

Once the billable charge is created, the membership ID is stamped as characteristic on the billable charge. This indicates that the billable charge is created for the respective membership. This algorithm fetches the characteristic type (which must be defined on the billable charge) from the **Characteristic Type For Membership Id** option type of the **C1-ASOBLLNG** feature configuration.

This batch is a multi-threaded batch. The multi-threading is based on membership ID and chunks for multi-threading are created based on numerical distribution of membership ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Bill Group Account ID	No	Used when you want to create billable charges for a particular account.
Membership ID	No	Used when you want to create billable charges for a particular membership.
Policy ID	No	Used when you want to create billable charges for a particular policy.
Chunk Size	No	Used to specify the number of billable charges you want to create in each work unit.

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Fully-Insured Billable Charge Creation (C1-FIBCR)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the SQI based billable charges are created for the memberships. The status of the corresponding timeline is set to **Complete I** in the **CI_PRCE_CALC** table. If any error occurs while creating a billable charge, the status of the corresponding timeline is set to **ErrI(E)** in the **CI_PRCE_CALC** table.

11. Funding Request

This section provides information about the **Funding Request Periodic Monitor (C1-FNDRQ)** batch.

11.1   Funding Request Periodic Monitor (C1-FNDRQ)

The **Funding Request Periodic Monitor (C1-FNDRQ)** batch is used to monitor or check whether there are any funding requests in the **Deferred Processing** status. If there is a funding request in the **Deferred Processing** status, the batch changes the status of the funding request to **Create Funding**. Then, the algorithms attached to the **Create Funding** status are triggered in the specified sequence.

This batch is a multi-threaded batch. The multi-threading is based on funding request ID and chunks for multi-threading are created based on numerical distribution of funding request ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor funding requests which are created using the business objects of a particular maintenance object. **Note:** By default, the parameter value is set to **C1-FUNDINREQ**.
Restrict By Batch Code	No	Used when you want to monitor funding requests whose current status is linked to the **Funding Request Periodic Monitor (C1-FNDRQ)** batch. The valid value is **true**.
Restrict By Business Object	No	Used when you want to monitor funding requests which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Restrict By Status Code	No	Used when you want to monitor funding requests which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Post Execution Check/Clean Up:

On successful completion of this batch, for debit bills, payment events are created or the records are inserted in the **CI_BILL_ACH** table and for credit bills, adjustments are created and inserted in **C1_FUNDING_REQ_DTLS_ADJ** table.

The status of the records is either set to **PROCESSED** or **ERROR** in the **C1_FUNDING_REQ_DTLS** table depending on whether the payments or adjustments are created successfully or not. Error messages corresponding to the record are inserted in **C1_FUNDING_REQ_DTLS_EXCP** table. The **BO_STATUS_CD** column corresponding to the funding request in the **C1_FUNDING_REQ** table is set to **COMPLETE**.

12. Health Insurance Policy

This section provides information about the **Policy Periodic Monitor (C1-POLPR)** batch.

12.1 Policy Periodic Monitor (C1-POLPR)

The **Policy Periodic Monitor (C1-POLPR)** batch is used to check whether there are any self-funded policies in the **Pending**, **In Force/Active**, or **Runout** status. If there is a self-funded policy in the **Pending** status, the system does the following:

- If the policy start date - X days (specified in the **C1-ASOPOLMOA** algorithm) is earlier than or equal to the system date, the system creates a To Do using the To Do type which is specified in the **To Do Type for Pending Status** parameter (of the **C1-ASOPOLMOA** algorithm).

- If the start date of the self-funded policy is earlier than or equal to the system date and the To Do created using the To Do type specified in the **To Do Type for Pending Status** parameter is completed, the system changes the status of the self-funded policy from **Pending** to **In Force/Active**.

If there is a self-funded policy in the **In Force/Active** status, the system does the following:

- If the policy end date - X days (specified in the **C1-ASOPOLMOA** algorithm) is earlier than or equal to the system date, the system creates a To Do using the To Do type which is specified in the **To Do Type for Active Status** parameter (of the **C1-ASOPOLMOA** algorithm).

- If the end date of the self-funded policy is earlier than or equal to the system date and the To Do created using the To Do type specified in the **To Do Type for Active Status** parameter is completed, the system changes the status of the self-funded policy from **In Force/Active** to **Runout**.

However, if there is a self-funded policy in the **Runout** status, the system does the following:

- If the runout end date - X days (specified in the **C1-ASOPOLMOA** algorithm) is earlier than or equal to the system date, the system creates a To Do using the To Do type which is specified in the **To Do Type for Runout Status** parameter (of the **C1-ASOPOLMOA** algorithm).

- If the runout end date of the self-funded policy is earlier than or equal to the system date and the To Do created using the To Do type specified in the **To Do Type for Runout Status** parameter is completed, the system changes the status of the self-funded policy from **Runout** to **Post Runout**.

This batch is a multi-threaded batch. The multi-threading is based on policy ID and chunks for multi-threading are created based on numerical distribution of policy ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor policies which are created using the business objects of a particular maintenance object. **Note:** By default, the parameter value is set to **C1-POLICY**.

Parameter Name	Mandatory (Yes or No)	Description
Batch Control	No	Used when you want to monitor policies whose current status is linked to the **Policy Periodic Monitor (C1-POLPR)** batch. The valid value is **true**.
Policy Type	No	Used when you want to monitor policies which are created using a particular policy type. **Note:** You must specify a policy type which is already defined in the system.
Business Object	No	Used when you want to monitor policies which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Status	No	Used when you want to monitor policies which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Policy Periodic Monitor (C1-POLPR)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the system does the following:

- Creates a To Do for a self-funded policy in the **CI_TD_ENTRY** table
- Updates the status of a self-funded policy in the **BO_STATUS_CD** column of the **CI_POLICY** table

13. Hold Request

This section provides information about the following batches:

- Hold Request Periodic Monitor (C1-HLDRQ)
- Hold Request Monitor (C1-HLMON)
- Delete Pending Bills and its Bill Segments (C1-DELBI)
- Hold Request Entity Derivation Monitor (C1-HLDET)

13.1 Hold Request Periodic Monitor (C1-HLDRQ)

The **Hold Request Periodic Monitor (C1-HLDRQ)** batch is used to monitor or check whether there are any hold requests in the **Deferred Processing** status. If there is a hold request in the **Deferred Processing** status, the batch changes the status of the hold request to **Active**. Then, the algorithm attached to the **Active** status is triggered.

This batch is a multi-threaded batch. The multi-threading is based on hold request ID and chunks for multi-threading are created based on numerical distribution of hold request ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor hold requests which are created using the business objects of a particular maintenance object. **Note:** By default, the parameter value is set to **C1-HOLDREQ**.
Batch Control	No	Used when you want to monitor hold requests whose current status is linked to the **Hold Request Periodic Monitor (C1-HLDRQ)** batch. The valid value is **true**.
Business Object	No	Used when you want to monitor hold requests which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.

Parameter Name	Mandatory (Yes or No)	Description
Status	No	Used when you want to monitor hold requests which are in a particular status.
		Note: This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Hold Request Periodic Monitor (C1-HLDRQ)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the hold request is changed from **Deferred Processing** to **Active**. The system does the following:

- If the start date of the hold request, hold entity, or hold process is earlier than the system date, it sets the respective start date to the system date. Note that the system does not change the start date of the hold request, hold entity, or hold process when it is a future date.

- If an active overdue process exists for the account whose overdue process is kept on hold through the hold request, it cancels the respective overdue process of the account. The status of the overdue process is changed to **Inactive**.

In addition, the system does the following:

- If the hold request is created for the **Account** entity, the **Bill Generation** process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date, it sets the bill after date to the hold entity end date or bill generation process end date whichever is earlier.

- If the hold request is created for the **Account** entity, the **Overdue** process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date, it sets the postpone credit review until date to the hold entity end date or overdue process end date whichever is earlier.

- If the hold request is created for the **Account** entity, the **Delinquency** process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date, it sets the postpone credit review until date to the hold entity end date or delinquency process end date whichever is earlier.

- If the hold request is created for the **Person** entity, the **Delinquency** process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date, it sets the postpone credit review until date to the hold entity end date or delinquency process end date whichever is earlier. If the **Hierarchy** option is selected for the person, it sets the postpone credit review until date for the person, its immediate accounts, its child persons, and for the child persons' accounts. Note that the system considers only the child persons and not the grandchild persons from the person's hierarchy and then derives the accounts where the child person is the main customer. However, if the **Hierarchy** option is not selected for the person, it sets the postpone credit review until date for the person and its immediate accounts (where the person is the main customer). For more information, see Hold Delinquency in the *Oracle Revenue Management and Billing Insurance User Guide*.

- If the hold request is created for the **Person** entity, the **Bill Generation** process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date, it sets the bill after date to the hold entity end date or bill generation process end date whichever is earlier. The bill after date is set either for all the accounts in the person's hierarchy or only for the persons' immediate accounts depending on whether the **Hierarchy** option is selected or not. Note that the system considers only the child persons and not the grandchild persons from the person's hierarchy and then derives the accounts where the child person is the main customer.

- If the hold request is created for the **Account** entity, the **Auto Pay** process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date, it sets the defer auto pay date to the hold entity end date or auto pay process end date whichever is earlier.

- If the hold request is created for the **Account** entity, the **Auto Pay** process is on hold, and the entity end date in the hold request is earlier than or equal to the batch business date, it updates the automatic payment details of the debit bills (if any) for the respective account in the **CI_BILL_ACH** table.

- If the hold request is created for the **Account** entity, the **Refund** process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date, it sets the hold refund until date to the hold entity end date or refund process end date whichever is earlier. In addition, the status of the non-final refund request (if any) for the respective account is set to **Hold**.

Note: If the hold entity end date is not specified, the system will consider the respective process end date while deriving the above dates. If both the hold entity end date and respective process end date are not specified, the system will consider the hold request end date while deriving the above dates.

13.2   Hold Request Monitor (C1-HLMON)

The **Hold Request Monitor (C1-HLMON)** batch is used to monitor or check whether there are any hold requests in the **Active** or **Released** status. It does the following:

- Automatically releases a hold request when the hold request end date is earlier than the batch business date

- Automatically releases a hold on the process of the entity when the derived hold end date (i.e., hold entity end date or respective process end date which is earlier) is earlier than the batch business date

- Derives the bill after date, postpone credit review until date, defer auto pay date, and hold refund until date when the derived hold start date is earlier than or equal to the batch business date

> **Note:** If the hold entity start date or process start date is later than the hold request start date, the above dates are derived through the **C1-HLMON** batch only on or after the hold entity start date or process start date.

If there is a hold request in the **Active** status, the batch does the following:

If…	Then...
The hold request is created for an account, the bill generation process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date	The entity end date or process end date whichever is earlier is set as the bill after date for the respective account. In addition, an entry is created for the hold request in the **CI_BILL_DEL_REQUEST** table.
	Note: The bill after date then appears corresponding to the account in the **Hold Entities** zone of the **Hold Request** screen. In addition, it appears in the **Bill On or After Date** field on the **Main** tab of the **Account** screen.
The hold request is created for an account, the overdue process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date	The entity end date or process end date whichever is earlier is set as the postpone credit review until date for the respective account.
	Note: The postpone credit review until date then appears corresponding to the account in the **Hold Entities** zone of the **Hold Request** screen. In addition, it appears in the **Postpone Credit Review Until** field on the **C & C** tab of the **Account** screen.
The hold request is created for an account, the delinquency process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date	The entity end date or process end date whichever is earlier is set as the postpone credit review until date for the respective account.
	Note: The postpone credit review until date then appears corresponding to the account in the **Hold Entities** zone of the **Hold Request** screen. In addition, it appears in the **Postpone Credit Review Until** field on the **C & C** tab of the **Account** screen.
The hold request is created for an account, the auto pay process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date	The entity end date or process end date whichever is earlier is set as the defer auto pay date for the respective account.
	Note: The defer auto pay date then appears corresponding to the account in the **Hold Entities** zone of the **Hold Request** screen. In addition, it appears in the **Defer Auto Pay Date** field on the **Auto Pay** tab of the **Account** screen.

If…	Then...
The hold request is created for an account, the refund process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date	The entity end date or process end date whichever is earlier is set as the hold refund until date for the respective account. In addition, the status of the non-final refund request (if any) for the respective account is set to **Hold**. **Note:** The hold refund until date then appears corresponding to the account in the **Hold Entities** zone of the **Hold Request** screen.
The hold request is created for an account, the auto pay process is on hold, and the entity end date in the hold request is earlier than the batch business date	The automatic payment details of the debit bills (if any) for the respective account are updated in the **CI_BILL_ACH** table.
The hold request is created for a person, the bill generation process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date	The entity end date or process end date whichever is earlier is set as the bill after date either for all the accounts in the person's hierarchy or only for the persons' immediate accounts depending on whether the **Hierarchy** option is selected or not. Note that the system considers only the child persons and not the grandchild persons from the person's hierarchy and then derives the accounts where the child person is the main customer. In addition, an entry is created for the hold request in the **CI_BILL_DEL_REQUEST** table. **Note:** The bill after date then appears in the **Bill On or After Date** field on the **Main** tab of the **Account** screen.
The hold request is created for a person, the delinquency process is on hold, and the entity start date in the hold request is earlier than or equal to the batch business date	The entity end date or process end date whichever is earlier is set as the postpone credit review until date for the respective person and its immediate accounts (where the person is the main customer). If the **Hierarchy** option is selected for the person, the system sets the postpone credit review until date for the person, its immediate accounts, its child persons, and for the child persons' accounts. Note that the system considers only the child persons and not the grandchild persons from the person's hierarchy and then derives the accounts where the child person is the main customer. **Note:** The postpone credit review until date then appears corresponding to the person in the **Hold Entities** zone of the **Hold Request** screen. In addition, the postpone credit review until date of the account appears in the **Postpone Credit Review Until** field on the **C & C** tab of the **Account** screen.

If…	Then...
The hold request end date is earlier than the batch business date	The status of the hold request is changed to **Released** and the release status reason is stamped corresponding to the hold request. The hold request, hold entity, and the hold process end dates are updated to the batch business date. Note that the entity and process end dates are updated only when their start date is not a future date.

Note: If the hold entity end date is not specified, the system will consider the respective process end date while deriving the above dates. If both the hold entity end date and respective process end date are not specified, the system will consider the hold request end date while deriving the above dates.

If there is a hold request in the **Released** status, the batch does the following:

If…	Then…
The hold request is created for an account, the auto pay process is on hold, and the entity end date in the hold request is earlier than or equal to the batch business date	The defer auto pay until date is set to the system date for the respective account. In addition, the automatic payment details of the debit bills (if any) for the respective account are updated in the **CI_BILL_ACH** table.
	Note: The defer auto pay date then appears corresponding to the account in the **Hold Entities** zone of the **Hold Request** screen. In addition, it appears in the **Defer Auto Pay Date** field on the **Auto Pay** tab of the **Account** screen.
The hold request is created for an account, the bill generation process is on hold, and the entity end date in the hold request is earlier than or equal to the batch business date	The bill after date is cleared for the respective account. This will enable the business to bill the account immediately on the same day when the hold is released.
	Note: The bill after date is cleared corresponding to the account in the **Hold Entities** zone of the **Hold Request** screen. In addition, it is cleared from the **Bill On or After Date** field on the **Main** tab of the **Account** screen.
The hold request is created for an account, the overdue process is on hold, and the entity end date in the hold request is earlier than or equal to the batch business date	The postpone credit review until date is set to the system date for the respective account.
	Note: The postpone credit review until date then appears corresponding to the account in the **Hold Entities** zone of the **Hold Request** screen. In addition, it appears in the **Postpone Credit Review Until** field on the **C & C** tab of the **Account** screen.

If…	Then…
The hold request is created for an account, the delinquency process is on hold, and the entity end date in the hold request is earlier than or equal to the batch business date	The postpone credit review until date is set to the system date for the respective account. **Note:** The postpone credit review until date then appears corresponding to the account in the **Hold Entities** zone of the **Hold Request** screen. In addition, it appears in the **Postpone Credit Review Until** field on the **C & C** tab of the **Account** screen.
The hold request is created for an account, the refund process is on hold, and the entity end date in the hold request is earlier than or equal to the batch business date	The hold refund until date is set to the system date for the respective account. In addition, the status of the refund request (if any) for the respective accounts is changed from **Hold** to its previous status. **Note:** The hold refund until date then appears corresponding to the account in the **Hold Entities** zone of the **Hold Request** screen.
The hold request is created for a person, the bill generation process is on hold, and the entity end date in the hold request is earlier than or equal to the batch business date	The bill after date is cleared either for all the accounts in the person's hierarchy or only for the persons' immediate accounts depending on whether the **Hierarchy** option is selected or not. Note that the system considers only the child persons and not the grandchild persons from the person's hierarchy and then derives the accounts where the child person is the main customer. **Note:** The bill after date is cleared from the **Bill On or After Date** field on the **Main** tab of the **Account** screen.
The hold request is created for a person, the delinquency process is on hold, and the entity end date in the hold request is earlier than or equal to the batch business date	The postpone credit review until date is set to the system date for the respective person and its immediate accounts (where the person is the main customer). If the **Hierarchy** option is selected for the person, the system sets the postpone credit review until date for the person, its immediate accounts, its child persons, and for the child persons' accounts. Note that the system considers only the child persons and not the grandchild persons from the person's hierarchy and then derives the accounts where the child person is the main customer. **Note:** The postpone credit review until date then appears corresponding to the person in the **Hold Entities** zone of the **Hold Request** screen. In addition, the postpone credit review until date of the account appears in the **Postpone Credit Review Until** field on the **C & C** tab of the **Account** screen.

This batch is a multi-threaded batch. The multi-threading is based on hold request ID and chunks for multi-threading are created based on numerical distribution of hold request ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.
Number of Release Days	No	Used when you do not want to monitor hold requests which were released earlier than the specified number of days. For example, if the batch business date is set to 02-20-2025 and the number of release days is set to 10, then this batch will not consider hold requests which were released earlier than 02-11-2025.
Release Status Reason	Yes	Used to indicate the reason why you want to release the hold request.
		Note: You must specify a reason which is already defined for the **Released** status of the **C1-HoldRequest** business object in the **Status Reason** screen.
Alert Type Code	No	Used to indicate the type of alert whose end date must be updated on the account when the hold request is released.
		Note: The alert end date is set to the date when the hold request is released. You need to ensure that the alert type used while activating and releasing the hold requests must be same. Therefore, always specify the same alert type in the following - **C1-HOLDACTV**, **C1-RELENTITY**, and **C1-HLMON**.
Hold Request ID	No	Used when you want to monitor a particular hold request.

Parameter Name	Mandatory (Yes or No)	Description
Status	No	Used when you want to monitor hold requests which are in a particular status.
		Note: This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.

Points to Note:

If you do not specify the batch business date, it is set to the current date.

If the **Hold Request Monitor (C1-HLMON)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

You need to execute the **C1-DPMON** batch after executing the **C1-HLMON** batch.

Post Execution Check/Clean Up:

On successful completion of this batch, the following actions take place for:

- **Active Hold Requests** - The bill after date, postpone credit review until date, defer auto pay date, and hold refund until date are updated in the **CI_ACCT** table when the respective process is kept on hold for an account through the hold request. Similarly, the postpone credit review until date is updated in the **CI_PER** and **CI_ACCT** tables, respectively, when the respective process is kept on hold for a person through the hold request. The automatic payment details are updated for the debit bills (if any) of the account in the **CI_BILL_ACH** table when its auto pay process is kept on hold and the entity end date in the hold request is earlier than or equal to the batch business date. In addition, the status of the active hold requests is changed to **Released** when the hold request end date is earlier than the batch business date.

- **Released Hold Requests** - The bill after date, postpone credit review until date, defer auto pay date, and hold refund until date are updated in the **CI_ACCT** table when the respective hold process is released for an account through the hold request. Similarly, the postpone credit review until date is updated in the **CI_PER** and **CI_ACCT** tables, respectively, when the respective hold process is released for a person through the hold request. The automatic payment details are updated for the debit bills (if any) of the account in the **CI_BILL_ACH** table when its auto pay process hold is released and the entity end date in the hold request is earlier than or equal to the batch business date.

13.3   Delete Pending Bills and its Bill Segments (C1-DELBI)

The **Delete Pending Bills and its Bill Segments (C1-DELBI)** batch considers the hold request records from the **CI_BILL_DEL_REQUEST** table where the batch run number is blank. It then deletes the following for the accounts' whose bill generation process is kept on hold through the hold request:

- First all the bill segments of pending bills that are in the **Freezable** or **Error** status
- Then the respective pending bills

This is a multi-threaded batch. The multi-threading is based on bill segment ID and chunks for multi-threading are created based on numerical distribution of bill segment ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Chunk Size	Yes	Used to specify the number of bill segments you want to delete in each work unit.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Delete Pending Bills and its Bill Segments (C1-DELBI)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the batch run number through which the pending bill and its bill segments are deleted is stored corresponding to the record in the **CI_BILL_DEL_REQUEST** table.

13.4   Hold Request Entity Derivation Monitor (C1-HLDET)

The **Hold Request Entity Derivation Monitor (C1-HLDET)** batch is used to monitor or check whether there are any hold requests in the **Entity Derivation Pending** status. Note that the system considers those hold requests whose start date is earlier than or equal to batch business date. If there is a hold request in the **Entity Derivation Pending** status, the batch invokes the monitor algorithm (i.e., **C1-MASSHOLD**) attached to the status.

The **C1-MASSHOLD** algorithm checks whether the **massHoldSw** field of the hold request is set to **Y**. If so, it derives the filter criteria specified in the hold request. It then creates a dynamic SQL query to fetch the list of accounts or persons that meets the filter criteria depending on whether the entity level is set to **Account** or **Person**. The derived accounts or persons are then stored as the hold entities for the hold request in the **C1_HOLD_REQ_DTLS** table. This algorithm sets the start date of all the hold entities to the system date and end date to null.

Once the hold request is updated, the algorithm then sets the **massHoldSw** field of the hold request to **N** and changes the status of the hold request from **Entity Derivation Pending** to **Draft**. For more information about the **C1-MASSHOLD** algorithm, see the respective section in the **Hold Request** chapter of *Oracle Revenue Management and Billing Banking User Guide* or *Oracle Revenue Management and Billing Insurance User Guide.*

This is a multi-threaded batch. The multi-threading is based on the hold request ID and the chunks for multi-threading are created based on the numerical distribution of hold request ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor hold requests which are created using the business objects of a particular maintenance object. **Note:** By default, the parameter value is set to **C1-HOLDREQ**.
Batch Control	No	Used when you want to monitor hold requests whose current status is linked to the **Hold Request Entity Derivation Monitor (C1-HLDET)** batch. The valid value is **true**.
Business Object	No	Used when you want to monitor hold requests which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Status	Yes	Used when you want to monitor hold requests which are in a particular status. **Points to Note:** By default, the parameter value is set to **ENTDERPENDG**. This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Hold Request Entity Derivation Monitor (C1-HLDET)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, a list of accounts or persons depending on the entity level of the hold request that meets the filter criteria are stored in the **C1_HOLD_REQ_DTLS** table. Once the hold entities are derived for the hold request, the status of the hold request is changed to **Draft**.

14. Inbound Message

This section provides information about the following batches:

- Health Care Inbound Message Periodic Monitor (C1-HCINB)
- Health Product and Plan Inbound Message Monitor (C1-HCPHP)

14.1 Health Care Inbound Message Periodic Monitor (C1-HCINB)

The **Health Care Inbound Message Periodic Monitor (C1-HCINB)** batch is used to monitor or check whether there are any health care inbound messages in the **Pending** status. If there is a health care inbound message in the **Pending** status, the batch changes the status of the health care inbound message to the default next status (i.e. **Validation**). Then, the algorithm attached to the **Validation** status is triggered. If the validation is successful, the system changes the status of the health care inbound message to **Processing**. Then, the algorithm attached to the **Processing** status is triggered. If the processing is successful, the system creates or updates the entities, such as person, account, policy, policy plan, membership, and/or billable charge, and then changes the status of the health care inbound message to **Processed**.

However, if an error occurs while validating or processing a health care inbound message, the system changes the status of the health care inbound message to **Rejected** and the algorithm attached to the **Rejected** status is triggered. A To Do is created using the To Do type specified in the **C1-TDCRINMSG** algorithm and assigned to the users with the To Do role specified in the **C1-TDCRINMSG** algorithm. The system checks whether the maximum retry attempts are met. If the retry attempt is less than or equal to the value specified in the **Maximum Retry** parameter of the **C1-TODORETRY** algorithm, the status of the health care inbound message is changed to **Pending**. However, if the retry attempt is greater than the value specified in the **Maximum Retry** parameter, the status of the health care inbound message remains as **Rejected**.

This batch is a multi-threaded batch. The multi-threading is based on inbound message ID and chunks for multi-threading are created based on numerical distribution of inbound message ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor health care inbound messages created using the business objects of a particular maintenance object.

Parameter Name	Mandatory (Yes or No)	Description
		Note: By default, the parameter value is set to **C1-INMSG**.
Batch Control	No	Used when you want to monitor health care inbound messages whose current status is linked to the **Health Care Inbound Message Periodic Monitor (C1-HCINB)** batch. The valid value is **true**.
Business Object	No	Used when you want to monitor health care inbound messages which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Status	No	Used when you want to monitor health care inbound messages which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

If the **Health Care Inbound Message Periodic Monitor (C1-HCINB)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

This batch description is articulated based on the lifecycle and logic defined in the **C1-HCInboundMessage** business object.

Post Execution Check/Clean Up:

If a health care inbound message is successfully validated and processed, the status of the health care inbound message is changed to **Processed**. If any error occurs while validating or processing a health care inbound message, the status of the health care inbound message is changed to **Rejected**. In addition, a message appears in the **Log** tab of the **Inbound Message** screen which indicates the error that occurred while executing the batch.

14.2 Health Product and Plan Inbound Message Monitor (C1-HCPHP)

The **Health Product and Plan Inbound Message Monitor (C1-HCPHP)** is used to monitor the health product and plan inbound messages for individual memberships. This batch checks whether there are any health product and plan inbound messages in the **Pending** status. If there is a health product and plan inbound message in the **Pending** status, the batch changes the status from **Pending** to **Validation**. The system then validates the health product and plan inbound message by triggering the algorithm attached to the **Validation** status. If a health product and plan inbound message is successfully validated, the system changes the status of the health product and plan inbound message to **Processing**. Then, the algorithm attached to the **Processing** status is triggered. If the processing is successful, the system creates or updates the entities, such as health product, health plan, pricing rules for the health plan, and then changes the status of the health product and plan inbound message to **Processed**.

This batch is a multi-threaded batch. The multi-threading is based on inbound message ID, and chunks for multi-threading are created based on the numerical distribution of inbound message ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor health product and plan inbound messages created using the business objects of a particular maintenance object. **Note:** By default, the parameter value is set to **C1-INMSG**
Batch Control	No	Used when you want to monitor health product and plan inbound messages whose current status is linked to the **Health Product and Plan Inbound Message Monitor (C1-HCPHP)** batch. The valid value is **true**.
Business Object	No	Used when you want to monitor health product and plan inbound messages which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.

Parameter Name	Mandatory (Yes or No)	Description
Status	No	Used when you want to monitor health product and plan inbound messages which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

If the **Health Product and Plan Inbound Message Monitoring (C1-HCPHP)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

This batch description is articulated based on the lifecycle and logic defined in the **C1-HCProdHlthPlnInbound** business object.

Post Execution Check/Clean Up:

If a health product and plan inbound message is successfully validated and processed, the status of the health product and plan inbound message is changed to **Processing**. If any error occurs while validating or processing a health product and plan inbound message, the status of the health product and plan inbound message is changed to **Rejected**. In addition, a message appears in the **Log** tab of the **Inbound Message** screen which indicates the error that occurred while executing the batch.

15. Offset Request

This section provides information about the **Offset Request Periodic Monitor (C1-OFSRQ)** batch.

15.1   Offset Request Periodic Monitor (C1-OFSRQ)

The **Offset Request Periodic Monitor (C1-OFSRQ)** batch is used to monitor or check whether there are any offset requests in the **Deferred Processing** or **Defer Unapplied** status. If there are any offset requests in the **Deferred Processing** status, the batch changes the status of the offset request to the default next status (i.e. **Creating Adjustment**). Then, the algorithms attached to the **Creating Adjustment** status are triggered in the specified sequence. If the transfer or offset adjustments are created successfully, the system changes the status of the offset request to **Processed**. However, if an error occurs while creating the transfer or offset adjustments, the system changes the status of the offset request to **Draft**.

Besides creating the transfer or offset adjustments for an offset request, the system does the following:

If…	Then the system…
A bill which is offset through an offset request is present in any active hold request, and the remaining outstanding amount of the bill after offsetting is not zero, but is less than the current hold amount	Updates the hold amount to the current hold amount plus remaining outstanding bill balance
A bill which is offset through an offset request is present in any active hold request and the remaining outstanding amount of the bill after offsetting is zero	Updates the hold end date of the bill in the hold request to the system date
A bill which is offset through an offset request is present in a funding request which is in the **Draft** status	Updates the funding amount of the bill in the funding request
A bill which is offset through an offset request is present in a funding request and the entire sum of the funding and garnishment amount of the bill is offset through the offset request	Excludes the bill from the funding request

However, if there are any offset requests in the **Defer Unapplied** status, the batch changes the status of the offset request to the default next status (i.e. **Unapplied Offset**). Then, the algorithms attached to the **Unapplied Offset** status are triggered in the specified sequence. The system then cancels the transfer or offset adjustments created through the offset request. Besides this, the system updates the funding amount of the bill if it is included in any funding request which is in the **Draft** status.

This batch is a multi-threaded batch. The multi-threading is based on offset request ID and chunks for multi-threading are created based on numerical distribution of offset request ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor offset requests created using the business objects of a particular maintenance object. **Note:** By default, the parameter value is set to **C1-OFFSETREQ**.
Batch Control	No	Used when you want to monitor offset requests whose current status is linked to the **Offset Request Periodic Monitor (C1-OFSRQ)** batch. The valid value is **true**.
Business Object	No	Used when you want to monitor offset requests which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Status	No	Used when you want to monitor offset requests which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Points to Note:

If the **Offset Request Periodic Monitor (C1-OFSRQ)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

This batch description is articulated based on the lifecycle and logic defined in the **C1-OffsetRequest** business object.

Post Execution Check/Clean Up:

On successful completion of this batch, the transfer or offset adjustments are created or canceled for the offset request. On creating the transfer or offset adjustments, the adjustment ID is stamped corresponding to the offset request record in the **XFER_ADJ_ID** column of the **C1_OFFSET_REQ_ADJ** table. In addition, the **BO_STATUS_CD** column corresponding to the offset request is set to **PROCESSED** or **DRAFT** in the **C1_OFFSET_REQ** table. The logs are maintained in the **C1_OFFSET_REQ_LOG** and/or **C1_HOLD_REQ_LOG** tables.

On canceling the transfer or offset adjustments, the status of the adjustment is updated in the **ADJ_STATUS_FLG** column of the **CI_ADJ** table.

16. Payment Agreement Request

This section provides information about the following batches:

- Payment Agreement Request Periodic Monitor (C1-PAREQ)
- Generate Automatic Payment for Payment Agreement Request (C1-APPAB)

16.1 Payment Agreement Request Periodic Monitor (C1-PAREQ)

The **Payment Agreement Request Periodic Monitor (C1-PAREQ)** batch is used to monitor or check whether there are any payment agreement requests in the **Active** status. If there is a payment agreement request in the **Active** status, the system checks whether the total unpaid amount of the bills (included in the payment agreement request) is equal to zero and whether each bill is fully matched. If so, the status of the payment agreement request is changed to **Kept Promise**. However, if the total unpaid amount of the bills is not equal to zero, the system checks whether the batch business date is later than the schedule date and does not fall within the grace period. If so, the system checks whether the total unpaid amount is greater than the total future schedule amount. If so, the status of the payment agreement request is changed to **Broken Promise**. However, if the batch business date is earlier than the schedule date or falls within the grace period, or the total unpaid amount is less than the total future schedule amount, the status of the payment agreement request remains in **Active**.

This batch is a multi-threaded batch. The multi-threading is based on payment agreement request ID and chunks for multi-threading are created based on numerical distribution of payment agreement request ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor payment agreement requests which are created using the business objects of a particular maintenance object.
		Note: By default, the parameter value is set to **C1-PAREQ**.
Batch Control	No	Used when you want to monitor payment agreement requests whose current status is linked to the **Payment Agreement Request Periodic Monitor (C1-PAREQ)** batch. The valid value is **true**.

Parameter Name	Mandatory (Yes or No)	Description
Business Object	No	Used when you want to monitor payment agreement requests which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Status	No	Used when you want to monitor payment agreement requests which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Payment Agreement Request Periodic Monitor (C1-PAREQ)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the payment agreement request is changed to either:

- **Kept Promise** (when the total unpaid amount of the bills (included in the payment agreement request) is equal to zero and each bill is fully matched)
- **Broken Promise** (when the total unpaid amount is greater than the total future schedule amount)

The status of the payment agreement request remains in the **Active** status when the batch business date is earlier than the schedule date or falls within the grace period, or the total unpaid amount is less than the total future schedule amount.

If any error occurs while executing the batch, a message appears in the **Log** tab of the **Payment Agreement Request** screen which indicates the error that occurred during batch execution.

16.2 Generate Automatic Payment for Payment Agreement Request (C1-APPAB)

The **Generate Automatic Payment for Payment Agreement Request (C1-APPAB)** batch is used to monitor or check whether there are any payment agreement requests in the **Active** status. If there is a payment agreement request in the **Active** status, the system checks whether the account for which the payment agreement request is created is eligible for automatic payment and the defer auto pay date (if any) defined for the account is earlier than the batch business date. If so, the system checks whether the extract date of the unpaid bill (with the earliest due date) is earlier than the schedule date. If so, the system creates the automatic payment for the unpaid bill on the schedule date. However, if the account is not eligible for automatic payment, or the defer auto pay date is equal to or later than batch business date, or the extract date is equal to or later than the schedule date, the automatic payment is not generated for the account.

This batch is a multi-threaded batch. The multi-threading is based on payment agreement request ID and chunks for multi-threading are created based on numerical distribution of payment agreement request ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Generate Automatic Payment for Payment Agreement Request (C1-APPAB)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the records are added in the following tables when the automatic payment is created for the account:

- CI_APAY_CLR_STG
- CI_PAY
- CI_PAY_EVENT
- CI_PAY_TNDR

The status of the automatic payment is set to **Incomplete**.

17. Payments

This section provides information about the following batches:

- Payment Upload (PUPL)

- Payment Upload – Step 2 - Distribute and Freeze (PUPLDF)

- Payment Upload – Step 3 - Payment Complete (PUPLCMPL)

- Create Automatic Payments and Refunds (APAYCRET)

- Activate Automatic Payments (ACTVTAPY)

- Extract Automatic Payments (APAYACH)

- Upload Response from Auto Clearing House (C1-APACK)

- Freeze or Cancel Automatic Payments (APAYRA)

- Cancel Automatic Refunds (AREFRA)

- To Do Creation for Automatic Payment Exception Records (APAYUPTD)

- Distribute and Freeze Automatic Payments (APAYDSFR)

- Create Tender Controls for Automatic Payments (BALAPY)

- Payment Request Periodic Monitor (C1-PAYRQ)

- Upload Lockbox Payment and Pay Instruction Files (C1-PUPSG)

- Payment Transfer Deferred Periodic Monitor (C1-PAYTR)

- Binder Payment Monitor Batch (C1-BPMNT)

17.1 Payment Upload (PUPL)

The **Payment Upload (PUPL)** batch is used to create the deposit control, tender controls, payment events, tenders, payments, and payment segments using payment data records in the staging area. In addition, the financial transactions are created when the payments are frozen. The system creates these objects only when the accounting date of the payment data record (i.e. date received in case of the **Payment Upload** process) is earlier than or equal to the batch business date. For the **Payment Upload** process, you need to configure this batch such that it is executed at regular intervals.

Note that this batch faces significant performance issues when the number of payment segments is huge for a single payment event or payment. To handle this performance issue, the system provides an alternative three-step approach wherein you need to execute the following batches in the specified sequence:

1. Payment Upload (PUPL)

2. Payment Upload – Step 2 - Distribute and Freeze (PUPLDF)

3. Payment Upload – Step 3 - Payment Complete (PUPLCMPL)

If you want to use three-step approach for uploading payment data records from the staging area, you must set the **Three Step Batch Processing (Y/N)** parameter to **Y**. However, if you do not specify the value for the **Three Step Batch Processing (Y/N)** parameter or set its value to **N**, you need to just execute the **Payment Upload (PUPL)** batch in the traditional manner.

In the three-step batch processing, the **Payment Upload (PUPL)** batch is used to create the deposit control, tender controls, payment events, tenders, and payments using the payment data records from the staging area. The status of the payment is set to **Step in Process**. In addition, the payment distribution algorithm is triggered and the payment distribution records are created in the **CI_PAY_DIST_TEMP** table. However, in the single-step batch processing, the **Payment Upload (PUPL)** batch behaves in the traditional manner as mentioned above (in the first paragraph).

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated.
		Note: You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.
Staging Header ID	No	Used to indicate the payment data file for which you want to create payments.
Three Step Batch Processing (Y/N)	No	Used to indicate whether you want to use three-step approach for uploading payment data records from the staging area. The valid values are: • Y • N If you set the parameter value to **Y**, you need to execute the **Payment Upload (PUPL)**, **Payment Upload – Step 2 - Distribute and Freeze (PUPLDF)**, and **Payment Upload – Step 3 - Payment Complete (PUPLCMPL)** batches in the above mentioned sequence. However, if you do not specify the value for this parameter or set the parameter value to **N**, you need to just execute the **Payment Upload (PUPL)** batch in the traditional manner.

Parameter Name	Mandatory (Yes or No)	Description
Division	No	Used when you want to create payments for accounts which belong to a particular division. **Note:** You must specify a division which is already defined in the system.
Bulk Insert Count	Yes (Conditional)	Used to indicate the number of records that you want to insert at once in the **CI_PAY_DIST_TEMP** table. **Note:** This parameter is required when the **Three Step Batch Processing (Y/N)** parameter is set to **Y**.

Note: If the **Payment Upload (PUPL)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the records are added in the following tables:

If the Three Step Batch Processing (Y/N) parameter is set to...	Then...
N	The records are added in the following tables: • CI_DEP_CTL • CI_TNDR_CTL • CI_PAY_EVENT • CI_PAY_TNDR • CI_PAY • CI_PAY_CHAR (if the characteristics exist for the payment) • CI_PAY_SEG • CI_FT • CI_FT_GL • CI_MATCH_EVT The status of the corresponding deposit control, tender control, payment tender, and payment tender staging record is changed to **Complete**. In addition, the status of the payment data record is changed to **Processed**. If an error occurs while creating a payment event and/or payment, an exception is added in the **CI_PAY_EVT_EXCP** or **CI_PAY_EXCP** table, respectively. If any error occurs while creating payment event for a payment record, the status of

If the Three Step Batch Processing (Y/N) parameter is set to…	Then…
	the deposit control, tender control, payment tender, and payment tender staging record is changed to **Error**. In addition, the status of the payment data record in the staging area is changed to **Error in Staging**.
	If a match event exists for the bill segment against which you want to apply the payment, the system uses the same match event for the payment segment. However, if the match event does not exist for the bill segment against which you want to apply the payment, the system creates a new match event for the payment segment using the match type.
Y	The records are added in the following tables: - CI_DEP_CTL - CI_TNDR_CTL - CI_PAY_EVENT - CI_PAY_TNDR - CI_PAY - CI_PAY_CHAR (if the characteristics exist for the payment) The status of the payment is set to **Step in Process**. In addition, the payment distribution records are created in the **CI_PAY_DIST_TEMP** table. If an error occurs while creating a payment event and/or payment, an exception is added in the **CI_PAY_EVT_EXCP** or **CI_PAY_EXCP** table, respectively. If any error occurs while creating payment event for a payment record, the status of the deposit control, tender control, payment tender, and payment tender staging record is changed to **Step in Error**. In addition, the status of the payment data record in the staging area is changed to **Error in Staging**.

Traditionally, the **Payment Upload (PUPL)** batch considered staging records which are in the **Error** or **Pending** status. If a staging record is in the **Error** status, the system used to reset the status to **Pending** and then further processed the staging record. This system behavior still exists for staging records which are not uploaded through the **Payment Upload** process. If the staging records are uploaded through the **Payment Upload** process, you need to manually correct the staging records and change the status of deposit control, tender control, payment tender, and payment tender staging records to **Pending**. The corrected staging records will then be processed when the **Payment Upload (PUPL)** batch is invoked at subsequent interval.

17.2   Payment Upload – Step 2 - Distribute and Freeze (PUPLDF)

The **Payment Upload – Step 2 - Distribute and Freeze (PUPLDF)** batch is used to consider the payment distribution records from the **CI_PAY_DIST_TEMP** table and accordingly create payment segments for the payments (which are created through the three-step PUPL batch processing). In addition, it creates the financial transaction for each payment segment and sets its status to **Frozen**. If any of the payment segment returns an error, the system will not hold the creation of other payment segments in the payment.

In the three-step PUPL batch processing, you must execute the **Payment Upload – Step 2 - Distribute and Freeze (PUPLDF)** batch after the **Payment Upload (PUPL)** batch. It is a multi-threaded batch. The multi-threading is based on the records present in the **CI_PAY_DIST_TEMP** table and chunks for multi-threading are created based on numerical distribution of the sequence number (i.e. **PUPLDF_SEQUENCE** column). You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Staging Header ID	No	Used to indicate the payment data file for which you want to create payment segments.
Division	No	Used when you want to create payment segments for accounts which belong to a particular division. **Note:** You must specify a division which is already defined in the system.
Chunk Size	Yes	Used to specify the number of payment segments you want to create in each work unit.
Maximum Batch Count	Yes	Used to specify the maximum number of payment segments once created the data

Parameter Name	Mandatory (Yes or No)	Description
		must be immediately committed to the database.

Points to Note:

You must specify the same parameters (for example, division, staging header ID, etc.) including the batch business date in the **PUPL, PUPLDF, and PUPLCMPL** batches during the three-step PUPL batch processing. Otherwise, erroneous results will occur.

Before executing the **Payment Upload – Step 2 - Distribute and Freeze (PUPLDF)** batch, you need to ensure that the **Payment Upload (PUPL)** batch is successfully executed.

If the **Payment Upload – Step 2 - Distribute and Freeze (PUPLDF)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

If the status of the **Payment Upload – Step 2 - Distribute and Freeze (PUPLDF)** batch run is set to **Error**, you need to first resolve the error and then execute the batch once again.

Post Execution Check/Clean Up:

On successful completion of this batch, the records are added in the following tables:

- CI_PAY_SEG
- CI_FT
- CI_FT_GL
- CI_MATCH_EVT

If a match event exists for the bill segment against which you want to apply the payment, the system uses the same match event for the payment segment. However, if the match event does not exist for the bill segment against which you want to apply the payment, the system creates a new match event for the payment segment using the match type.

17.3   Payment Upload – Step 3 - Payment Complete (PUPLCMPL)

The **Payment Upload – Step 3 - Payment Complete (PUPLCMPL)** batch is used to change the status of the payment to **Frozen** or **Error** depending on whether its payment segments and financial transactions are successfully created in the system. If all payments in a payment event are successfully created, the status of the corresponding deposit control, tender control, payment tender, and payment staging records is set to **Complete**. If any error occurs while creating payment event for a payment record, the status of the deposit control, tender control, payment tender, and payment staging records is changed to **Error**. In addition, the status of the payment record in the pre-staging area is changed to **Error in Staging**.

If the total amount distributed does not match with the payment amount, the status of the payment is set to **Step Process Error**. The **Payment Upload – Step 3 - Payment Complete (PUPLCMPL)** batch provides an option to clean up such payments. If the **Step Processing Cleanup** parameter is set to **Y**, the **PUPLCMPL** batch does the following activities during cleanup:

- Cancels the payment which is in the **Step in Error** status.

- Creates a new payment under the payment event with the same amount, match type, and match value. However, its payment amount is not distributed and the payment status is set to **Incomplete**.
- Changes the status of the payment staging record to **Incomplete**.

However, if the **Step Processing Cleanup** parameter is set to **N**, the **PUPLCMPL** batch does not initiate the cleanup process. It just changes the status of the payment staging record to **Error** when the total amount distributed does not match with the payment amount.

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Staging Header ID	No	Used to indicate the payment data file whose payment record status you want to update.
Step Processing Cleanup	Yes	Used to indicate whether you want to initiate the cleanup process for a payment which is in the **Step Process Error** status. The valid values are **Y** and **N**. **Note:** By default, the parameter value is set to **N**.
Cancel Reason Code	Yes (Conditional)	Used to indicate the reason why you want to cancel the payment. **Points to Note:** You must specify a payment cancel reason which is already defined in the system. This parameter is required when the **Step Processing Cleanup** parameter is set to **Y**.

Parameter Name	Mandatory (Yes or No)	Description
Division	No	Used when you want to update payment records of accounts which belong to a particular division.
		Note: You must specify a division which is already defined in the system.

Points to Note:

You must specify the same parameters (for example, division, staging header ID, etc.) including the batch business date in the **PUPL, PUPLDF, and PUPLCMPL** batches during the three-step PUPL batch processing. Otherwise, erroneous results will occur.

Before executing the **Payment Upload – Step 3 - Payment Complete (PUPLCMPL)** batch, you need to ensure that the **Payment Upload – Step 2 - Distribute and Freeze (PUPLDF)** batch is successfully executed.

If the **Payment Upload – Step 3 - Payment Complete (PUPLCMPL)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

If the status of the **Payment Upload – Step 3 - Payment Complete (PUPLCMPL)** batch run is set to **Error**, you need to first resolve the error and then execute the batch once again.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the payment in the **PAY_STATUS_FLG** column of the **CI_PAY** table is changed to **Frozen**, **Step in Error**, or **Incomplete**. The status of the payment tender staging record in the **PAY_TND_STG_ST_FLG** column of the **CI_PAY_TNDR_ST** table is accordingly changed to **Complete**, **Error**, or **Incomplete**. Finally, the status of the payment data record is changed to **Processed** or **Error in Staging**.

In addition, the system removes the payment distribution records from the **CI_PAY_DIST_TEMP** table when the status of the payment is changed to **Frozen** or **Incomplete**.

17.4 Create Automatic Payments and Refunds (APAYCRET)

When the automatic payment facility is used for an account, the system calculates the automatic payment amount and extract date during the bill completion and stamps these details against the bill in the **CI_BILL_ACH** table. The **Create Automatic Payments and Refunds (APAYCRET)** batch is used to create automatic payments and adjustments for automatic refunds. For each record in the **CI_BILL_ACH** table, the batch checks the date specified corresponding to the respective account in the **DEFER_AUTO_PAY_DT** field of the **CI_ACCT** table. If the defer auto pay date is later than or equal to the batch business date, the batch will skip the record of the **CI_BILL_ACH** table. However, if the defer auto pay date is earlier than the batch business date, the batch will consider the record with debit or credit amount from the **CI_BILL_ACH** table. If the automatic payment amount stamped against a bill is in debit, the batch creates automatic payment and payment event for such bill. It also creates a clearing record for each automatic payment in the **CI_APAY_CLR_STG** table. The payment freeze date in the clearing records is set to **NULL**.

Note: The automatic payment is created only when an algorithm created using the **APAY-CREATE** algorithm type is attached to the **Automatic Payment Creation** system event in the installation options.

However, if the automatic payment amount stamped against a bill is in credit, the batch creates a refund request against the bill.

Note: The refund request is created only when an algorithm created using the **C1-AREF-CRET** algorithm type is attached to the **Automatic Adjustment Creation** system event in the installation options.

The **C1-AREF-CRET** algorithm indicates the refund request type using which the refund request must be created. If the approval workflow is configured for the refund request type, the status of the refund request is set to **Pending for Approval**. However, if the approval workflow is not configured for the refund request type, the status of the refund request is set to **Processed** and the refund adjustment is created in the **Frozen** status. In addition, it creates a clearing record for each automatic refund in the **CI_ADJ_CLR_STG** table.

The extract date is calculated and stamped against the automatic payment and refund clearing records. The extract date is later used to activate the clearing record.

You must execute the **Create Automatic Payments and Refunds (APAYCRET)** batch in the specified sequence when the **Autopay Creation Option** field in the installation options is set to the following:

Autopay Creation Option	Batch Execution Sequence
Create On Extract Date	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Distribute and Freeze Automatic Payments (APAYDSFR) 5. Create Tender Controls for Automatic Payments (BALAPY)
Freeze Payment on Notification	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Upload Response from Auto Clearing House (C1-APACK) 5. Freeze or Cancel Automatic Payments (APAYRA) or Cancel Automatic Refunds (AREFRA) 6. To Do Creation for Automatic Payment Exception Records (APAYUPTD) 7. Create Tender Controls for Automatic Payments (BALAPY)

This batch is a multi-threaded batch. The multi-threading is based on bill ID and chunks for multi-threading are created based on numerical distribution of bill ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Batch Business Date	No	Used to identify the bills whose extract date is earlier than or equal to the specified date. The batch then creates automatic payments or refund adjustments for such bills.

Parameter Name	Mandatory (Yes or No)	Description
		Note: If you do not specify any date, the batch business date is set to the current date.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated.
		Note: You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.
Division or Division Group	No	Used when you want to create automatic payments or refund adjustments for accounts belonging to a particular division or division group.
		Points to Note: You must specify a division or division group which is already defined in the system. If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division. If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.
Recalculate Automatic Payment Amount (Y or N)	No	Used to indicate whether the automatic payment amount must be recalculated before creating the automatic payment or refund adjustment. The valid values are:

Parameter Name	Mandatory (Yes or No)	Description
		• Y • N **Note:** If you set the **Recalculate Automatic Payment Amount (Y or N)** parameter to **Y**, the system recalculates the automatic payment amount before creating the automatic payment or refund adjustment. The automatic payment amount is then updated against the bill in the **CI_BILL_ACH** table. If the entire amount is manually paid, then the corresponding record is removed from the **CI_BILL_ACH** table.
Re-evaluate Auto-Pay ID	Yes	Used when you want to re-evaluate auto pay details of the completed bills of an account where the **RE_EVAL_APAY_SW** flag is set to **Y**. The valid values are: • Y • N **Points to Note:** By default, the parameter value is set to **N**. The batch only considers the records with the debit amount from the **CI_BILL_ACH** table. If an account has non-rule based auto pay instructions, the batch will fetch the auto pay ID for the completed bill once again and store it corresponding to the record in the **CI_BILL_ACH** table. However, if an account has rule based auto pay instructions, the batch will fetch the auto pay ID for the completed bill once again and store it corresponding to the records in both **CI_BILL_ACH** and **C1_FT_EXT** tables. Note that if the system is not able to derive the auto pay details for a completed bill of an account, the system will create a To Do using the **C1-REAPY** To Do type.

Note: If the **Create Automatic Payments and Refunds (APAYCRET)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the records are added in the following tables when the automatic payment is created:

- CI_APAY_CLR_STG
- CI_PAY
- CI_PAY_EVENT
- CI_PAY_TNDR

However, the records are added in the following tables when the refund adjustment is created:

- CI_ADJ_CLR_STG
- C1_REF_WO_REQ
- C1_REF_WO_REQ_DTLS
- C1_REF_WO_REQ_LOG
- CI_ADJ
- CI_ADJ_APREQ

While adding a clearing record in the **CI_APAY_CLR_STG** table, the **APAY_DIST_FRZ_DT** column is set to **NULL**. In addition, the extract date is added in the **SCHED_EXTRACT_DT** column of the **CI_APAY_CLR_STG** or **CI_ADJ_CLR_STG** table, respectively.

17.5   Activate Automatic Payments (ACTVTAPY)

Unless you activate a clearing record, you cannot extract the clearing record in a flat file. The **Activate Automatic Payments (ACTVTAPY)** batch is used to activate a clearing record by stamping the **APAYACH** batch control and incremental batch run number on the clearing record. Only those clearing records whose extract date is earlier than or equal to the batch business date are activated through this batch.

You must execute the **Activate Automatic Payments (ACTVTAPY)** batch in the specified sequence when the **Autopay Creation Option** field in the installation options is set to the following:

Autopay Creation Option	Batch Execution Sequence
Create At Bill Completion	1. Activate Automatic Payments (ACTVTAPY) 2. Extract Automatic Payments (APAYACH) 3. Create Tender Controls for Automatic Payments (BALAPY)
Create On Extract Date	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Distribute and Freeze Automatic Payments (APAYDSFR) 5. Create Tender Controls for Automatic Payments (BALAPY)

Autopay Creation Option	Batch Execution Sequence
Freeze Payment on Notification	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Upload Response from Auto Clearing House (C1-APACK) 5. Freeze or Cancel Automatic Payments (APAYRA) or Cancel Automatic Refunds (AREFRA) 6. To Do Creation for Automatic Payment Exception Records (APAYUPTD) 7. Create Tender Controls for Automatic Payments (BALAPY)

This batch is a multi-threaded batch. The multi-threading is based on clearing staging ID and chunks for multi-threading are created based on numerical distribution of clearing staging ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Batch Business Date	No	Used to identify the clearing records whose extract date is earlier than or equal to the specified date. **Note:** If you do not specify any date, the batch business date is set to the current date.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Division or Division Group	No	Used when you want to activate clearing records of accounts belonging to a particular division or division group. **Points to Note:** You must specify a division or division group which is already defined in the system. If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the

Parameter Name	Mandatory (Yes or No)	Description
		system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division. If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.

Note: If the **Activate Automatic Payments (ACTVTAPY)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the **APAYACH** batch control is added in the **BATCH_CD** column and the next batch run number is added in the **BATCH_NBR** column corresponding to the clearing record in the **CI_APAY_CLR_STG** table.

17.6   Extract Automatic Payments (APAYACH)

The **Extract Automatic Payments (APAYACH)** batch is used to extract the clearing records with the latest batch run number in a flat file. This flat file is then sent to the auto clearing house for review.

You must execute the **Extract Automatic Payments (APAYACH)** batch in the specified sequence when the **Autopay Creation Option** field in the installation options is set to the following:

Autopay Creation Option	Batch Execution Sequence
Create At Bill Completion	1. Activate Automatic Payments (ACTVTAPY) 2. Extract Automatic Payments (APAYACH) 3. Create Tender Controls for Automatic Payments (BALAPY)
Create On Extract Date	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Distribute and Freeze Automatic Payments (APAYDSFR) 5. Create Tender Controls for Automatic Payments (BALAPY)
Freeze Payment on Notification	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Upload Response from Auto Clearing House (C1-APACK) 5. Freeze or Cancel Automatic Payments (APAYRA) or Cancel Automatic Refunds (AREFRA)

Autopay Creation Option	Batch Execution Sequence
	6. To Do Creation for Automatic Payment Exception Records (APAYUPTD)
	7. Create Tender Controls for Automatic Payments (BALAPY)

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Output Directory	No	Used to specify the path where you want to save the flat file. **Note:** If you do not specify the path, the flat file is, by default, saved in the sploutput directory.
Output File Name	No	Used to specify the name which you want to use for the flat file. **Note:** If you do not specify the name, the file is, by default, named as ACHMMDDYYHHSS. Here, MMDDYY indicates the date and HHSS indicates the time when the file is extracted.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Division or Division Group	No	Used when you want to extract clearing records of accounts belonging to a particular division or division group. **Points to Note:** You must specify a division or division group which is already defined in the system. If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified

Parameter Name	Mandatory (Yes or No)	Description
		value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division. If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.

Note: If the **Extract Automatic Payments (APAYACH)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the flat file is created at the specified location. If you have not specified the path and name of the output file, by default, the file is saved with the ACHMMDDYYHHSS name in the sploutput directory.

17.7 Upload Response from Auto Clearing House (C1-APACK)

The **Upload Response from Auto Clearing House (C1-APACK)** batch is used to read and upload the EDI 824 files which contain the review comments received for automatic payment clearing records from the auto clearing house.

On uploading an EDI 824 file, the system checks whether the automatic payment clearing ID specified in each record exists in the system. If so, it creates an automatic payment clearing staging record for the automatic payment clearing ID. If the acknowledgement code in the record is set to **TE** or **TR**, the system considers that the auto clearing house has rejected the automatic payment clearing record. The reason code (i.e. upload reason) specified in the **Reject Upload Reason** parameter of the **C1-APAY** algorithm is stored corresponding to the automatic payment clearing staging record in the **CI_APAY_STAGE_UP_REM** table. In addition, the error code (i.e. payment cancelation reason) is stored corresponding to the automatic payment clearing staging record in the **CI_APAY_STAGE_UP_REM** table. However, if the acknowledgement code is blank or if the acknowledgement code is set to a value other than **TE** or **TR**, the system considers that the auto clearing house has accepted the automatic payment clearing record. The reason code (i.e. upload reason) specified in the **Success Upload Reason** parameter of the **C1-APAY** algorithm is stored corresponding to the automatic payment clearing staging record in the **CI_APAY_STAGE_UP_REM** table. In addition, the error code (i.e. NOC reason) is stored corresponding to the automatic payment clearing staging record in the **CI_APAY_STAGE_UP_REM** table. In this case, the record may contain more than one error code.

The status of the automatic payment clearing staging record is set to **Pending**. If the error code (i.e. payment cancelation reason or NOC reason) is invalid, the status of the automatic payment clearing staging record is set to **Error**.

The system sets the automatic payment distribution and freeze date for each automatic payment clearing record (for which automatic payment clearing staging record is added in the system) to the payment date (which is stored on the payment event).

You must execute the **Upload Response from Auto Clearing House (C1-APACK)** batch in the specified sequence when the **Autopay Creation Option** field in the installation options is set to the following:

Autopay Creation Option	Batch Execution Sequence
Freeze Payment on Notification	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Upload Response from Auto Clearing House (C1-APACK) 5. Freeze or Cancel Automatic Payments (APAYRA) or Cancel Automatic Refunds (AREFRA) 6. To Do Creation for Automatic Payment Exception Records (APAYUPTD) 7. Create Tender Controls for Automatic Payments (BALAPY)

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
File Directory	Yes	Used to specify the path on the server from where you want to upload the EDI 824 files. Here, you must specify relative path and not absolute path. The path should begin with either of the following: • **@SHARED_DIR** – Used when you want to upload the EDI 824 files from the shared directory on the server. • **@INSTALL_DIR** – Used when you want to upload the EDI 824 files from the SPLEBASE directory which is defined in the spl.properties file.
File Name	Yes	Used to specify the name of the EDI 824 file that you want to upload. Here, you can specify either the exact file name or the file name prefix (for example, ACH*) using which you want search and upload a set of files.

Parameter Name	Mandatory (Yes or No)	Description
Archive Files (Y or N)	Yes	Used to indicate whether you want to archive the files once the automatic payment clearing staging records are created in the staging area. The valid values are: • Y • N **Note:** By default, the parameter value is set to **N**.
Archive Directory	Yes (Conditional)	Used to specify the path where you want to move the files on the server once the automatic payment clearing staging records are created in the staging area. Here, you must specify relative path and not absolute path. The path should begin with either of the following: • **@SHARED_DIR** – Used when you want to move the EDI 824 files to the shared directory on the server. • **@INSTALL_DIR** – Used when you want to move the EDI 824 files to the SPLEBASE directory which is defined in the spl.properties file. **Note:** This parameter is required when the **Archive Files (Y or N)** parameter is set to **Y**.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Upload Response from Auto Clearing House (C1-APACK)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the automatic payment clearing staging records are created in the **CI_APAY_STAGE_UP** table. The reason and error codes of the automatic payment clearing staging records are stored in the **CI_APAY_STAGE_UP_REM** table. The payment date is added in the **APAY_DIST_FRZ_DT** column corresponding to the automatic payment clearing record in the **CI_APAY_CLR_STG** table.

If the payment cancelation reason or NOC reason does not exist in the system or if the NOC reason is in the **Inactive** status, an exception is logged in the **CI_APAY_STGUP_EXC** table.

17.8   Freeze or Cancel Automatic Payments (APAYRA)

The **Freeze or Cancel Automatic Payments (APAYRA)** batch is used to freeze or cancel the automatic payments based on the reason code (i.e. upload reason). It checks whether the reason code (i.e. upload reason) and error code (i.e. payment cancelation reason or NOC reason) specified in the automatic payment clearing staging record exist in the system. An automatic payment clearing staging record which is in the **Pending** status and for which payment freeze date is specified in the automatic payment clearing record is considered during the batch execution.

If the reason code and error code exist in the system, the system executes the algorithms attached to the reason code in the specified sequence. If the type of reason code is **Cancel**, the attached algorithm cancels the automatic payment and payment event. If the type of reason code is **Success**, the attached algorithm freezes the automatic payment if the payment freeze date is specified. However, if the type of reason code is **Success** and the error code is specified along with the reason code, the attached algorithm freezes the automatic payment (if the payment freeze date is specified), stores error codes (i.e. NOC reasons) as payment tender characteristics and then creates a To Do entry to notify user about these NOC reasons. Finally, the status of the automatic payment clearing staging record is changed to **Complete**.

If the reason code and/or error code (i.e. upload reason, payment cancelation reason, or NOC reason) specified in the automatic payment clearing staging record does not exist in the system, the status of the automatic payment clearing staging record is changed to **Error**. In addition, if the status of the NOC reason is **Inactive** in the system, the status of the automatic payment clearing staging record is changed to **Error**.

You must execute the **Freeze or Cancel Automatic Payments (APAYRA)** batch in the specified sequence when the **Autopay Creation Option** field in the installation options is set to the following:

Autopay Creation Option	Batch Execution Sequence
Freeze Payment on Notification	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Upload Response from Auto Clearing House (C1-APACK) 5. Freeze or Cancel Automatic Payments (APAYRA) or Cancel Automatic Refunds (AREFRA) 6. To Do Creation for Automatic Payment Exception Records (APAYUPTD) 7. Create Tender Controls for Automatic Payments (BALAPY)

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.

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. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Division or Division Group	No	Used when you want to freeze or cancel automatic payments of accounts belonging to a particular division or division group. **Points to Note:** You must specify a division or division group which is already defined in the system. If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division. If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.
Validate Payment Freeze Date (Y or N)	No	Used to indicate whether you want to validate the payment freeze date before freezing the automatic payments. The valid values are: • Y • N **Points to Note:** If you set this parameter to **Y**, the system considers only those automatic payment clearing records whose payment freeze date is earlier than or equal to the batch business date. However, if you set this parameter to **N**, the system considers automatic payment clearing records irrespective of whether the payment freeze date is set to a past date, current date, or future date.

Parameter Name	Mandatory (Yes or No)	Description
		If you do not specify any value, by default, the parameter value is set to **N**.

Note: If the **Freeze or Cancel Automatic Payments (APAYRA)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the automatic payment clearing staging record is updated in the **APAY_UP_STATUS_FLG** column of the **CI_APAY_STAGE_UP** table. The status of automatic payment is updated in the **PAY_STATUS_FLG** column of the **CI_PAY** table. The **APAY_DIST_FRZ_DT** column of the corresponding automatic payment clearing record in the **CI_APAY_CLR_STG** table is set to **NULL**.

If the upload reason, payment cancelation reason, or NOC reason does not exist in the system, an exception is logged in the **CI_APAY_STGUP_EXC** table.

17.9   Cancel Automatic Refunds (AREFRA)

The **Cancel Automatic Refunds (AREFRA)** batch is used to void the refund request and cancel the refund adjustments based on the reason code (i.e. upload reason). It considers only those automatic refund clearing staging records which are in the **Pending** status. It checks whether the reason code (i.e. upload reason) and error code (i.e. void status reason) specified in the automatic refund clearing staging record exist in the system. If the reason code and error code exist, the system executes the algorithms attached to the reason code in the specified sequence. If the type of reason code is **Cancel**, the attached algorithm changes the status of refund request to **Voided** and the corresponding refund adjustments are canceled. If the type of reason code is **Success**, no changes are made to the refund request and refund adjustments. Finally, the status of the automatic refund clearing staging record is changed to **Complete**.

If the reason code and/or error code (i.e. upload reason, void status reason) specified in the automatic refund clearing staging record does not exist in the system, the status of the automatic refund clearing staging record is changed to **Error**.

You must execute the **Cancel Automatic Refunds (AREFRA)** batch in the specified sequence when the **Autopay Creation Option** field in the installation options is set to the following:

Autopay Creation Option	Batch Execution Sequence
Freeze Payment on Notification	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Upload Response from Auto Clearing House (C1-APACK) 5. Freeze or Cancel Automatic Payments (APAYRA) or Cancel Automatic Refunds (AREFRA) 6. To Do Creation for Automatic Payment Exception Records (APAYUPTD) 7. Create Tender Controls for Automatic Payments (BALAPY)

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated.
		Note: You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.
Division or Division Group	No	Used when you want to void refund requests of accounts belonging to a particular division or division group.
		Points to Note:
		You must specify a division or division group which is already defined in the system.
		If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division.
		If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.

Note: If the **Cancel Automatic Refunds (AREFRA)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the automatic refund clearing staging record is updated in the **APAY_UP_STATUS_FLG** column of the **CI_APAY_STAGE_UP** table. The status of refund request is updated in the **BO_STATUS_CD** column of the **C1_REF_WO_REQ** table. In addition, the status of the refund adjustment is updated in the **ADJ_STATUS_FLG** column of the **CI_ADJ** table.

If the upload reason or void status reason does not exist in the system, an exception is logged in the **CI_APAY_STGUP_EXC** table.

17.10 To Do Creation for Automatic Payment Exception Records (APAYUPTD)

The **To Do Creation for Automatic Payment Exception Records (APAYUPTD)** batch is used to create a To Do using the **APAYUPTD** To Do type when an exception is logged for an automatic payment clearing staging record in the **CI_APAY_STGUP_EXC** table. The system creates one To Do entry for all exceptions that have occurred for an automatic payment clearing staging record.

> **Note:** At present, a To Do is not created when an exception is logged for the automatic refund clearing staging record in the **CI_APAY_STGUP_EXC** table.

Only those exceptions where review date does not exist in the **CI_APAY_STGUP_EXC** table are considered during the batch execution. You must execute the **To Do Creation for Automatic Payment Exception Records (APAYUPTD)** batch in the specified sequence when the **Autopay Creation Option** field in the installation options is set to the following:

Autopay Creation Option	Batch Execution Sequence
Freeze Payment on Notification	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Upload Response from Auto Clearing House (C1-APACK) 5. Freeze or Cancel Automatic Payments (APAYRA) or Cancel Automatic Refunds (AREFRA) 6. To Do Creation for Automatic Payment Exception Records (APAYUPTD) 7. Create Tender Controls for Automatic Payments (BALAPY)

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **To Do Creation for Automatic Payment Exception Records (APAYUPTD)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the records are added in the **CI_TD_ENTRY** table. Also, the date when the To Do is generated is added in the **REVIEW_DT** column of the **CI_APAY_STGUP_EXC** table.

17.11 Distribute and Freeze Automatic Payments (APAYDSFR)

The **Distribute and Freeze Automatic Payments (APAYDSFR)** batch is used to distribute and freeze automatic payment records. It considers only those automatic payment clearing staging records from the **CI_APAY_CLR_STG** table whose distribution and freeze date is earlier than or equal to batch business date.

You must execute the **Distribute and Freeze Automatic Payments (APAYDSFR)** batch in the specified sequence when the **Autopay Creation Option** field in the installation options is set to the following:

Autopay Creation Option	Batch Execution Sequence
Create On Extract Date	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Distribute and Freeze Automatic Payments (APAYDSFR) 5. Create Tender Controls for Automatic Payments (BALAPY)

This batch is a multi-threaded batch. The multi-threading is based on clearing staging ID and chunks for multi-threading are created based on numerical distribution of clearing staging ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Batch Business Date	No	Used to identify the clearing staging records whose distribution date is earlier than or equal to the specified date. **Note:** If you do not specify any date, the batch business date is set to the current date.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.
Division or Division Group	No	Used when you want to distribute and freeze automatic payments of accounts belonging to a particular division or division group.
		Points to Note:
		You must specify a division or division group which is already defined in the system.
		If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division.
		If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.

Note: If the **Distribute and Freeze Automatic Payments (APAYDSFR)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of automatic payment is updated in the **PAY_STATUS_FLG** column of the **CI_PAY** table.

17.12 Create Tender Controls for Automatic Payments (BALAPY)

The **Create Tender Controls for Automatic Payments (BALAPY)** batch is used to create tender and deposit controls for each automatic payment which is frozen, but not yet linked to any tender control. One tender control is created for the **APAYACH** batch control and batch run number combination. The payment tenders of these automatic payments are then linked to the tender control. This batch also balances the open tender control records and changes the status of the tender and deposit controls, accordingly.

You must execute the **Create Tender Controls for Automatic Payments (BALAPY)** batch in the specified sequence when the **Autopay Creation Option** field in the installation options is set to the following:

Autopay Creation Option	Batch Execution Sequence
Create At Bill Completion	1. Activate Automatic Payments (ACTVTAPY) 2. Extract Automatic Payments (APAYACH) 3. Create Tender Controls for Automatic Payments (BALAPY)
Create On Extract Date	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Distribute and Freeze Automatic Payments (APAYDSFR) 5. Create Tender Controls for Automatic Payments (BALAPY)
Freeze Payment on Notification	1. Create Automatic Payments and Refunds (APAYCRET) 2. Activate Automatic Payments (ACTVTAPY) 3. Extract Automatic Payments (APAYACH) 4. Upload Response from Auto Clearing House (C1-APACK) 5. Freeze or Cancel Automatic Payments (APAYRA) or Cancel Automatic Refunds (AREFRA) 6. To Do Creation for Automatic Payment Exception Records (APAYUPTD) 7. Create Tender Controls for Automatic Payments (BALAPY)

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.
Division or Division Group	No	Used when you want to create tender controls for automatic payments of accounts belonging to a particular division or division group.
		Points to Note: You must specify a division or division group which is already defined in the system. If you specify the value for the **Division** parameter, the system will first check whether it is a division group. If so, the system will perform the expected behavior for all the divisions that belong to the division group. However, if the specified value is not a division group, the system will check whether it is a division and will accordingly perform the expected behavior for the respective division. If you do not specify the value for the **Division** parameter, the system will perform the expected behavior for all the divisions in the system.

Note: If the **Create Tender Controls for Automatic Payments (BALAPY)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the records are added in the following tables:

- CI_DEP_CTL
- CI_TNDR_CTL

The status is updated in the **DEP_CTL_STATUS_FLG** and **TNDR_CTL_ST_FLG** columns of the **CI_DEP_CTL** and **CI_TNDR_CTL** tables, respectively. In addition, the tender control ID is stamped in the **TNDR_CTL_ID** column of the **CI_PAY_TNDR** table.

17.13 Payment Request Periodic Monitor (C1-PAYRQ)

The **Payment Request Periodic Monitor (C1-PAYRQ)** batch is used to monitor or check whether there are any payment requests in the **Deferred Distribution** status. If there is a payment request in the **Deferred Distribution** status, the batch changes the status of the payment request to **Distributed**. Then, the algorithm attached to the **Distributed** status is triggered which distributes the tender or transfer amount.

> **Note:** On distributing the tender amount, the payment event, payments, payment segments, and payment tender are created. The payments are created in the **Freezable** status. If any error occurs while creating a payment, the payment is created in the **Error** status. If all payments of a payment event are in the **Freezable** status, the status of the payment event is set to **Balanced**. However, if any payment of a payment event is in the **Error** status, the status of the payment event is set to **Unbalanced**. On distributing the payment amount (during transfer), the new payments are created in the **Freezable** status. If any error occurs while creating a new payment, the payment is created in the **Error** status. The status of the payment event is set to **Unbalanced**. The old payments in the payment event that you want to transfer remains in the **Frozen** or **Error** status.

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor payment requests created using the business objects of a particular maintenance object.
		Note: By default, the parameter value is set to **C1-PAYREQ**.
Restrict By Batch Code	No	Used when you want to monitor the payment requests whose current status is linked to the **Payment Request Periodic Monitor (C1-PAYRQ)** batch. The valid value is **true**.
Restrict By Payment Request Type	No	Used when you want to monitor the payment requests which are created using a particular payment request type.
		Note: You must specify a payment request type which is already defined in the system.
Restrict By Business Object	No	Used when you want to monitor the payment requests which are created using a particular business object.
		Note: You must specify a business object which is already defined in the system.
Restrict By Status Code	No	Used when you want to monitor the payment requests which are in a particular status.

Parameter Name	Mandatory (Yes or No)	Description
		Note: This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Override maximum errors	No	Used to override the maximum number of errors allowed before the batch run is terminated.
		Note: You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Payment Request Periodic Monitor (C1-PAYRQ)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the payment request is changed from **Deferred Distribution** to **Distributed**.

17.14 Upload Lockbox Payment and Pay Instruction Files (C1-PUPSG)

The **Upload Lockbox Payment and Pay Instruction Files (C1-PUPSG)** batch functions differently depending on whether the **BPR** tag in the file is set to **D** or **I**. If the **BPR** tag in the file is set to **D**, this batch reads the 820 EDI lockbox file received from the bank containing the payment and remittance details for accounts, and upload the payment data in the payment upload staging tables.

It uses the source system customer number or taxpayer identification number specified in the payment or remittance record to identify the customer for whom the payment is made. If the customer is found in ORMB, it checks whether the identified customer has an account. If the identified customer has one account, the batch executes the algorithm (which is created using the **Payment Tender Staging Distribution - Pay Oldest Bill First (C1-CCPYTSADS)** algorithm type) attached to the **Payment Tender Staging Distribution** system event of the customer class to which the account belongs.

> **Points to Note:**
>
> This algorithm does the following:
>
> >> If the **Check Binder Payment** parameter is set to **Y**, the algorithm checks whether the payment is the first payment for the account and the tender used for the payment is not automatic payment. If so, the payment is applied against the contract which is created using the contract type specified in the payment distribution algorithm of the Binder Payment match type.
>
> >> If the **Check Promise To Pay** parameter is set to **Y**, the algorithm checks whether there is an active promise to pay for the account. If so, the payment is applied against the contract which is created using the contract type specified in the payment distribution algorithm of the On Account match type.
>
> >> If the **Check Payment Agreement Request** parameter is set to **Y**, the algorithm checks whether there is an active payment agreement request for the account. If so, the payment is applied against the contract which is created using the contract type specified in the payment distribution algorithm of the On Account match type.
>
> However, if the **Check Binder Payment**, **Check Promise To Pay**, and **Check Payment Agreement Request** parameters are set to **N**, the payment is applied to the account's unpaid bills in the order of the due date (i.e. oldest bill first). If the payment amount is greater than the account's billed balance plus overpayment threshold amount, the entire payment amount is applied against the excess credit contract of the account.
>
> However, if the payment amount is greater than the account's billed balance, but less than account's billed balance plus overpayment threshold amount, the payment is first applied against the unpaid bills of the account and the overpayment amount is then applied against the excess credit contract of the account.

However, if the identified customer has multiple accounts, the batch identifies the account using either of the following (in the specified order):

1. **Invoice ID** – It is used to find the account for which the invoice is created.
2. **MICR** - It is used find the payment where the MICR is stamped as a characteristic. Once a payment is identified, the batch finds the account for which the payment is created. In this way, the batch derives the account to which the payment must be applied. However, the MICR is used to derive the account only when the **Search Account Using MICR (Y or N)** parameter in the batch is set to **Y**.

Once the account is identified, the batch executes the algorithm (which is created using the **Payment Tender Staging Distribution - Pay Oldest Bill First (C1-CCPYTSADS)** algorithm type) attached to the **Payment Tender Staging Distribution** system event of the customer class to which the account belongs. However, if there are no accounts for the identified customer, or account could not be found in ORMB, or the customer could not be found in ORMB, the payment is applied against the suspense contract defined on the tender source which is associated with the external source (lockbox) ID.

If the **BPR** tag in the file is set to **I**, this batch uploads the files from the specified location on the server. Once a file is successfully validated, the reconciliation is created for the file in the **Draft** status. The reconciliation is immediately transitioned to the **Send Notification** status and the algorithms attached to the **Send Notification** status are executed. Once the To Do is created, the status of the reconciliation is changed to **Pending**. A pay instruction is created for each record in the file and its status is set to either **Pending** or **Error** depending on whether it is successfully validated or not.

This batch does not upload a file when another file with the same name is already uploaded and its reconciliation is in any other status other than the one specified in the **Reconciliation Cancel Status Code** parameter.

This batch is a multi-threaded batch. The multi-threading is based on file and chunks for multi-threading are created based on numerical distribution of files. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
File Directory	Yes	Used to specify the path on the server from where you want to upload the EDI 820 files. Here, you must specify relative path and not absolute path. The path should begin with either of the following: • **@SHARED_DIR** – Used when you want to upload the EDI 820 files from the shared directory on the server. • **@INSTALL_DIR** – Used when you want to upload the EDI 820 files from the SPLEBASE directory which is defined in the spl.properties file.
File Name Prefix	Yes	Used when you want to upload EDI 820 files with a particular prefix from the specified path. **Note:** By default, the parameter value is set to **usb_brms_lbx1_***.
Line Separator	Yes	Used to indicate the character specified in the EDI 820 files to represent the end of line. **Note:** By default, the parameter value is set to **~**.
Field Separator	Yes	Used to indicate the character specified in the EDI 820 files to represent the end of data record. **Note:** By default, the parameter value is set to *****.

Parameter Name	Mandatory (Yes or No)	Description
Search Account Using MICR (Y or N)	No	Used to indicate whether you want to search the account to which the payment must be applied using the MICR code. The valid values are: • Y • N **Note:** By default, the parameter value is set to **Y**.
Customer Class	No	Used when you want to apply payments for accounts which belong to a particular customer class. **Points to Note:** You must specify a customer class which is already defined in the system. You can specify comma-separated values for this parameter.
Division	No	Used when you want to apply payments for accounts which belong to a particular division. **Note:** You must specify a division which is already defined in the system.
General Suspense Match Type	Yes	Used to specify the match type using which you want to apply the payment against the suspense contract defined on the tender source. **Note:** By default, the parameter value is set to **SUSPENSE**.
On Account Match Type	Yes	Used to specify the match type using which you want to apply the payment against the contract of the account. **Note:** By default, the parameter value is set to **SUSPENSE**.

Parameter Name	Mandatory (Yes or No)	Description
Source System Customer Number Characteristic Type	Yes	Used to specify the characteristic type which you want to use to store the source system customer number. This type of characteristic is defined on the payment upload staging record when the account to which the payment is applied is derived using the source system customer number. **Note:** By default, the parameter value is set to **C1SRCSID**. You can change the characteristic type, if required. However, you must specify a characteristic type where the characteristic entity is set to **Payment**.
Remittance Invoice ID Characteristic Type	Yes	Used to specify the characteristic type which you want to use to store the invoice ID. This type of characteristic is defined on the payment upload staging record when the account to which the payment is applied is derived using the invoice ID. **Note:** By default, the parameter value is set to **C1RMIVID**. You can change the characteristic type, if required. However, you must specify a characteristic type where the characteristic entity is set to **Payment**.
Remittance Payment Amount Characteristic Type	Yes	Used to specify the characteristic type which you want to use to store the payment amount. This type of characteristic is defined on the payment upload staging record when the account to which the payment is applied is derived using the invoice ID. **Note:** By default, the parameter value is set to **C1RMPYMT**. You can change the characteristic type, if required. However, you must specify a characteristic type where the characteristic entity is set to **Payment**.

Parameter Name	Mandatory (Yes or No)	Description
Payment Review Reason Characteristic Type	Yes	Used to specify the characteristic type which you want to use to store the payment reason. This type of characteristic is defined on the payment upload staging record when the payment is applied either against the suspense contract defined on the tender source or against the contract of the account. **Note:** By default, the parameter value is set to **C1PYRVWR**. You can change the characteristic type, if required. However, you must specify a characteristic type where the characteristic entity is set to **Payment**.
Payment Review Reason for Multiple Accounts	Yes	Used to specify the characteristic value that must be specified when the customer has multiple accounts and the required account could not be derived using the Invoice ID or MICR code. **Note:** By default, the parameter value is set to **MULTCUAC**.
Taxpayer Identification Number Characteristic Type	Yes	Used to specify the characteristic type which you want to use to store the taxpayer identification number. This type of characteristic is defined on the payment upload staging record when the account to which the payment is applied is derived using the taxpayer identification number. **Note:** By default, the parameter value is set to **C1TXPYID**. You can change the characteristic type, if required. However, you must specify a characteristic type where the characteristic entity is set to **Payment**.
Excess Credit Contract Type	Yes	Used to specify the contract type using which the excess credit contract should be created if it does not exist for the account. **Note:** By default, the parameter value is set to **EXCSCRED**. You can change the parameter value, if required. However, you must specify a contract type which is already defined in the system.

Parameter Name	Mandatory (Yes or No)	Description
Person Identifier Type	Yes	Used to indicate the person identifier type. The system derives the customer or person using the person identifier type and person identifier combination.
		Note: By default, the parameter value is set to **SSN**. You can change the parameter value, if required. However, you must specify a person identifier type which is already defined in the system.
Remittance Source System Customer Number Characteristic Type	Yes	Used to specify the characteristic type which you want to use to store the remittance source system customer number. This type of characteristic is defined on the payment upload staging record when the account to which the payment is applied is derived using the remittance source system customer number.
		Note: By default, the parameter value is set to **C1RMSCID**. You can change the characteristic type, if required. However, you must specify a characteristic type where the characteristic entity is set to **Payment**.
Remittance Taxpayer Identification Number Characteristic Type	Yes	Used to specify the characteristic type which you want to use to store the remittance taxpayer identification number. This type of characteristic is defined on the payment upload staging record when the account to which the payment is applied is derived using the remittance taxpayer identification number.
		Note: By default, the parameter value is set to **C1RMTXID**. You can change the characteristic type, if required. However, you must specify a characteristic type where the characteristic entity is set to **Payment**.
Reconciliation Type	No	Used to indicate the reconciliation type using which you want to create the reconciliation.
		Note: You must specify a reconciliation type which is already defined in the system.

Parameter Name	Mandatory (Yes or No)	Description
Reconciliation Cancel Status Code	No	Used to specify the status code to which the reconciliation is transitioned on cancelation. This is used when you are trying to upload a file when another file with the same name is already uploaded and its reconciliation is in any other status other than the one specified in this parameter.
Archive Files (Y or N)	Yes	Used to indicate whether you want to archive the files once the payment records are added in the payment upload staging tables. The valid values are: • Y • N **Note:** By default, the parameter value is set to **N**.
Archive Directory	Yes (Conditional)	Used to specify the path where you want to move the files on the server once the payment records are added in the payment upload staging tables. Here, you must specify relative path and not absolute path. The path should begin with either of the following: • **@SHARED_DIR** – Used when you want to move the EDI 820 files to the shared directory on the server. • **@INSTALL_DIR** – Used when you want to move the EDI 820 files to the SPLEBASE directory which is defined in the spl.properties file. **Note:** This parameter is required when the **Archive Files (Y or N)** parameter is set to **Y**.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Upload Lockbox Payment and Pay Instruction Files (C1-PUPSG)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the records are added in the following tables when the **BPR** tag in the file is set to **D**:

- CI_DEP_CTL_ST
- CI_TNDR_CTL_ST
- CI_PAY_ST
- CI_PAY_ST_CHAR

However, when the **BPR** tag in the file is set to **I**, the records are added in the following tables:

- C1_RECONCILIATION
- C1_PAY_INS

A log entry generated for the reconciliation and pay instruction during the status transition is added in the **C1_RECONCILIATION_LOG** and **C1_PAY_INS_LOG** tables, respectively. In addition, if an error occurs while validating a pay instruction, it is recorded in the **C1_PAY_INS_LOG** table.

17.15 Payment Transfer Deferred Periodic Monitor (C1-PAYTR)

The **Payment Transfer Deferred Periodic Monitor (C1-PAYTR)** batch is used to monitor or check whether there are any payment transfer requests in the **Payment Derivation Pending** status. If there are any payment transfer requests in the **Payment Derivation Pending** status, the system identifies the payments which are eligible for transfer from the payment list of the payment transfer request in the following manner:

1. If the payments are created against the contracts, the system considers the following option types of the **C1_CMO** feature configuration to determine the payments which are eligible for transfer from the payment list:

   - Suspense Contract Type
   - On Account Contract Type
   - Excess Credit Contract Type

2. However, if the payments are created against the bills, the system considers the bill date and bill amount to determine the payments which are eligible for transfer from the payment list.

Once the eligible payments are identified, the system changes the status of the payment transfer request from **Payment Derivation Pending** to **Draft**.

Note: This batch is used when partial transfer is done at the payment event level through a payment transfer request and number of payments in the payment transfer request exceeds the defer payment processing count.

This batch is a multi-threaded batch. The multi-threading is based on payment transfer request ID and chunks for multi-threading are created based on numerical distribution of the payment transfer request ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor payment transfer requests which are created using the business objects of a particular maintenance object. **Note:** By default, the parameter value is set to **C1-PAYREQ**.
Batch Control	No	Used when you want to monitor payment transfer requests whose current status is linked to the **Payment Transfer Deferred Periodic Monitor (C1-PAYTR)** batch. The valid value is **true**.
Payment Request Type	No	Used when you want to monitor payment transfer requests which are created using a particular payment request type. **Note:** You must specify a payment request type which is already defined in the system.
Business Object	No	Used when you want to monitor payment transfer requests which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Status	No	Used when you want to monitor payment transfer requests which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated.
		Note: You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Payment Transfer Deferred Periodic Monitor (C1-PAYTR)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the payments from the payment list of the payment transfer request which are eligible for transfer are added in the **C1_PAY_DETAILS** table. The system stamps the transfer priority corresponding to eligible payments in the **TRANSFER_PRIORITY** column of the **C1_PAY_DETAILS** table. The system assigns transfer priority to each eligible payment in the following sequence:

1. Considers payments against the contracts which are created using the contract type specified in the **Suspense Contract Type** option type of the **C1_CMO** feature configuration. If there are multiple payments which are created on the suspense contracts, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1 and P2 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, and C1 and C2 are both suspense contracts, then the system sets the transfer priority of P1 and P2 to 1 in the payment transfer request.

2. Considers payments against the contracts which are created using the contract type specified in the **On Account Contract Type** option type of the **C1_CMO** feature configuration. For example, if you partially transfer P1 and P2 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, C1 is a suspense contract, and C2 is an on account contract, then the system sets the transfer priority of P1 to 1 and P2 to 2 in the payment transfer request. However, if you partially transfer P3 from the A1 account to the A2 account through a payment transfer request where P3 is created against C3 and C3 is an on account contract, then the system sets the transfer priority of P3 to 1 in the payment transfer request. If there are multiple payments on the on account contracts, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1, P2, and P3 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, P3 is created against C3, C1 is a suspense contract, and C2 and C3 are both on account contracts, then the system sets the transfer priority of P1 to 1 and P2 and P3 to 2 in the payment transfer request.

3. Considers payments against the contracts which are created using the contract type specified in the **Excess Credit Contract Type** option type of the **C1_CMO** feature configuration. For example, if you partially transfer P1 and P2 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, C1 is an on account contract, and C2 is an excess credit contract, then the system sets the transfer priority of P1 to 1 and P2 to 2 in the payment transfer request. However, if you partially transfer P3 from the A1 account to the A2 account through a payment transfer request where P3 is created against C3 and C3 is an excess credit contract, then the system sets the transfer priority of P3 to 1 in the payment transfer request. If there are multiple payments on the excess credit contracts, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1, P2, and P3 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, P3 is created against C3, C1 is a suspense contract, and C2 and C3 are both excess credit contracts, then the system sets the transfer priority of P1 to 1 and P2 and P3 to 2 in the payment transfer request.

4. Considers payments created against the bills in the descending order of the bill date (i.e. bill with the latest date). If there are multiple payments which are created against the bills having the same bill date, the system considers the payments which are created against the bills in the descending order of the bill amount (i.e. bill with the highest amount). However, if there are multiple payments which are created against the bills having the same bill date and bill amount, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1, P2, P3, and P4 from the A1 account to the A2 account through a payment transfer request where P1 is created against B1 (with the bill date 01-Jan-2020 and bill amount 100$), P2 is created against B2 (with the bill date 01-Feb-2023 and bill amount 20$), P3 is created against B3 (with the bill date 01-Mar-2023 and bill amount 30$), and P4 is created against B4 (with the bill date as 01-Mar-2023 and bill amount 40$), the system sets the transfer priority of P4 to 1, P3 to 2, P2 to 3, and P1 to 4, respectively, in the payment transfer request.

5. Considers payments which are created against entities other than contracts or bills and assigns the same transfer priority to these payments.

In addition, the system sets the **PY_CANCEL_SW** column corresponding the payments in the **C1_PAY_DETAILS** table to **Y** or **N** depending on whether these eligible payments are considered for transfer or not. Finally, the **BO_STATUS_CD** column corresponding to the payment transfer request in the **C1_PAY_REQ** table is set to **DRAFT**.

17.16 Binder Payment Monitor Batch (C1-BPMNT)

The **Binder Payment Monitor Batch (C1-BPMNT)** batch is used to monitor the binder payment for individual memberships. This batch checks whether there are any individual memberships where:

- Individual membership status is set to the value specified in the **Membership Status Code for Pending Effectuation** attribute of the binder payment preference.

- Membership status reason is set to the value specified in the **Membership Status Reason when Awaiting Binder Payment** attribute of the binder payment preference.

> **Note:** The batch considers the binder payment preference which is specified in the **Binder Payment Field Mapping** option type of the **C1-ASOBLLNG** feature configuration.

If so, the batch checks whether the **Binder Payment Applicability** option is set to **Y** for the individual membership. If so, the system calculates grace date using the membership start date and grace days. If the grace date is equal to or earlier than system date, the batch derives the account for the individual membership. While deriving the account for the individual membership, the system checks whether the characteristic types specified in the **Account Identifier Type Char Type** and **Account Identifier Value Char Type** option types of the **C1-ASOBLLNG** feature configuration are defined for the individual membership. If so, the system derives the account using these characteristics defined for the individual membership. If these characteristics are not defined for the individual membership, the system derives the account of financially responsible member person. The system checks whether a payment with the binder payment identification characteristic exists against any contract of the account. If so, the system derives the value from the **Binder Payment Identification ID Type** attribute of the binder payment preference. The system then checks whether the binder payments are received for the individual membership using the values from the following:

- Binder Payment Identification Characteristic
- Binder Payment Identification ID Type attribute of the binder payment preference

If the binder payments are received for the individual membership, the system does the following:

If the Consider Binder Lability Amount set to…	Then…
Y	The system validates whether the sum of binder payments is equal to or greater than the threshold amount. If so, the system does the following: • Creates a To Do using the To Do type which is specified in the **To Do Type for Binder Payment Received** attribute of the binder payment preference. • Updates the membership status reason to the value specified in the **Membership Status Reason when Binder Payment Received** attribute of the binder payment preference. • Updates the member person status reason using the membership status reason preference which is specified in the **Status Reason Mapping** option type of the **C1-ASOBLLNG** feature configuration. • Changes the bill after date of the billable charges when the **Hold Billing** option for the individual membership is set to **Y**.
N	The system does not validate the binder payment against the threshold amount. If the binder payment amount is greater than zero, the system does the following: • Creates a To Do using the To Do type which is specified in the **To Do Type for Binder Payment Received** attribute of the binder payment preference. • Updates the membership status reason to the value specified in the **Membership Status Reason when Binder Payment Received** attribute of the binder payment preference.

If the Consider Binder Lability Amount set to…	Then…
	• Updates the member person status reason using the membership status reason preference which is specified in the **Status Reason Mapping** option type of the **C1-ASOBLLNG** feature configuration.
	Note: In this case, the system does not update the bill after date of the billable charges when the **Hold Billing** option for the individual membership is set to **Y**. The system updates the bill after date of the billable charges when the status of the individual membership is changed to **Active** through a health care inbound message.

If the grace date is later than the system date, the batch does the following irrespective of whether the **Consider Binder Lability Amount** attribute is set to **Y** or **N**:

- Creates a To Do using the To Do type which is specified in the **To Do Type for Binder Payment Not Received** attribute of the binder payment preference.

- Updates the membership status reason to the value specified in the **Membership Status Reason when Binder Payment Not Received** attribute of the binder payment preference.

- Updates the member person status reason using the membership status reason preference which is specified in the **Status Reason Mapping** option type of the **C1-ASOBLLNG** feature configuration.

The following configurations must be considered before this batch is executed:

- The To Do creation process gets invoked only when the **To Do Creation Algorithm for Binder Payments (C1-BNFTODO)** algorithm is attached on the **C1-BPMNT** batch as a soft parameter.

- Status Reason must be configured in the **Field Mapping** screen.

- If the billable charges are kept on hold then those are released only when the system generates a message once the membership activation happens successfully.

This batch is a multi-threaded batch. The multi-threading is based on file and chunks for multi-threading are created based on numerical distribution of files. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Membership Business Object	Yes	Used when you want to monitor binder payments of individual memberships which are created using a particular business object.
		Note: You must specify a business object which is already defined in the system.
Algorithm for TO DO Creation	Yes	Used to specify the algorithm through which a To Do is created.

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Post Execution Check/Clean Up:

On successful completion of this batch, following new columns are added in the **CI_MEMBERSHIP** table:

- THRESHOLD_AMT
- NET_LIABILITY_AMT

The membership post processing algorithm generates an error, if the **Net Liability Amount** is not received for a membership. Further, this **Net Liability Amount** is used to calculate the threshold amount.

18. Pricing Management

This section provides information about the **Process Fees Request and Persist Fees for an Account (C1-ACFEE)** batch.

18.1 Process Fees Request and Persist Fees for an Account (C1-ACFEE)

The **Process Fees Request and Persist Fees for an Account (C1-ACFEE)** batch is used to calculate and persist the fees for the account in the **C1_FEES** table. It must be executed when a user wants to store fees.

It invokes the **C1-AccountFees** business service which determines effective price item pricing for an account and persist calculated fees for price items of type **Only Pricing** whose type is set to **Fees**.

This batch considers all those accounts which satisfy the given input parameters. While doing price assignment user can specify the pricing frequency to be used from the already define schedule periods. If the Batch business date falls in between the schedule period and the data corresponding to that has not been already persisted then batch persists the fees based upon the advance pricing switch and price assignment type.

Advance Pricing Switch conditions:

- Y - Batch business date should be on or after effective start date and on or before effective end date.
- N - Batch business date should be on effective end date.

> **Points to Note:**
>
> Effective start date - Price assignment start date and schedule period start date whichever is later.
>
> Effective end date - Price assignment end date and schedule period end date whichever is earlier.

This batch is a multi-threaded batch and the multi-threading is based on the account Id. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Product Code	No	Used when you want to calculate account fees for all accounts having this product linked. **Note:** You must specify a product which is already defined in the system.
Price Item Code	No	Used when you want to calculate account fees for given price item only. **Note:** You must specify a price item which is already defined in the system.

Parameter Name	Mandatory (Yes or No)	Description
Division	No	Used when you want to calculate account fees for all accounts of a specific division.
		Note: You must specify a division which is already defined in the system.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: Fees for an account will be calculated for POST processing price assignment types only.

Post Execution Check/Clean Up:

On successful completion of this batch, the fees is calculated and persisted for the fees type price items in the **C1_FEES**, **C1_FEES_CALC**, **C1_FEES_CALC_LN**, **C1_FEES_CALC_LN_CHAR**, **C1_FEES_CHAR**, **C1_FEES_K**, and **C1_FEES_PARAM**, **C1_FEES_SQ** tables.

19. Reconciliation

This section provides information about the following batches:

- Pay Instruction CSV File Upload (C1-RECUP)
- Reconciliation Periodic Monitor (C1-RCNM)
- Reconciliation Cancelation Periodic Monitor (C1-RCNMD)
- Pay Instruction Periodic Monitor (for Pending Status) (C1-PIPMO)
- Pay Instruction Periodic Monitor (for Open Status) (C1-PIMDF)
- Duplicate Reconciliation (C1-RECDU)

19.1   Pay Instruction CSV File Upload (C1-RECUP)

The **Pay Instruction CSV File Upload (C1-RECUP)** batch is used to upload the pay instruction files from the specified location on the server. Once a file is successfully validated, the reconciliation is created for the file in the **Draft** status. The reconciliation is immediately transitioned to the **Send Notification** status and the algorithms attached to the **Send Notification** status are executed. Once the To Do is created, the status of the reconciliation is changed to **Pending**. A pay instruction is created for each record in the file and its status is set to either **Pending** or **Error** depending on whether it is successfully validated or not.

This batch does not upload a file when another file with the same name is already uploaded and its reconciliation is in any other status other than the one specified in the **Reconciliation Cancel Status Code** parameter.

This batch is a multi-threaded batch. The multi-threading is based on file and chunks for multi-threading are created based on numerical distribution of files. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
File Directory	Yes	Used to specify the path on the server from where you want to upload the CSV files. Here, you must specify relative path and not absolute path. The path should begin with either of the following: • **@SHARED_DIR** – Used when you want to upload the CSV files from the shared directory on the server. • **@INSTALL_DIR** – Used when you want to upload the CSV files from the SPLEBASE directory which is defined in the `spl.properties` file.

Parameter Name	Mandatory (Yes or No)	Description
File Name	No	Used to specify the name of the CSV file that you want to upload. Here, you can specify either the exact file name or the file name prefix (for example, Member*) using which you want search and upload a set of files.
Reconciliation Type	Yes	Used to indicate the reconciliation type using which you want to create the reconciliation.
		Note: You must specify a reconciliation type which is already defined in the system.
Reconciliation Cancel Status Code	Yes	Used to specify the status code to which the reconciliation is transitioned on cancelation. This is used when you are trying to upload a file when another file with the same name is already uploaded and its reconciliation is in any other status other than the one specified in this parameter.
Pay Instruction CSV Files in Error - To Do Type	Yes	Used to specify the type of notification that you want to generate when the CSV file is not uploaded successfully.
		Note: You must specify a To Do type which is already defined in the system.
Pay Instruction CSV Files in Error - To Do Role	No	Used to specify the To Do role to indicate the users to whom you want to send the notification when the CSV file is not uploaded successfully.
		Note: You must specify a To Do role which is already associated with the specified To Do type.
Pay Instruction File Name Characteristic Type	Yes	Used to specify the characteristic type that you want to use to store the reconciliation file name along with the path. This characteristic is defined on the To Do which is generated when the CSV file is not uploaded successfully.
		Note: You must specify a characteristic type where the characteristic entity is set to **To Do Entry**.

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Pay Instruction CSV File Upload (C1-RECUP)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the records are added in the following tables:

- C1_RECONCILIATION
- C1_PAY_INS

A log entry which is generated for the reconciliation and pay instruction during the status transition is added in the **C1_RECONCILIATION_LOG** and **C1_PAY_INS_LOG** tables, respectively. In addition, if an error occurs while validating a pay instruction, it is recorded in the **C1_PAY_INS_LOG** table.

19.2   Reconciliation Periodic Monitor (C1-RCNM)

The **Reconciliation Periodic Monitor (C1-RCNM)** batch is used to monitor or check whether there is any reconciliation in the **Pending Reconciliation** status. If there is a reconciliation in the **Pending Reconciliation** status, the status of the reconciliation is changed to **Initial Reconciliation in Progress** and the algorithms attached to the **Initial Reconciliation in Progress** status are executed. If all pay instructions in the reconciliation are fully reconciled, the status of the reconciliation is changed to **Completed**. However, if one or more pay instructions in the reconciliation are not fully reconciled, the status of the reconciliation is changed to **Open**.

This batch is a multi-threaded batch. The multi-threading is based on reconciliation ID and chunks for multi-threading are created based on numerical distribution of reconciliation ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor reconciliations which are created using the business objects of a particular maintenance object.
		Note: By default, the parameter value is set to **C1-RECONCLT**.

Parameter Name	Mandatory (Yes or No)	Description
Batch Control	No	Used when you want to monitor reconciliations whose current status is linked to the **Reconciliation Periodic Monitor (C1-RCNM)** batch. The valid value is **true**.
Reconciliation Type	No	Used when you want to monitor reconciliations which are created using a particular reconciliation type. **Note:** You must specify a reconciliation type which is already defined in the system.
Business Object	No	Used when you want to monitor reconciliations which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Status	No	Used when you want to monitor reconciliations which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Reconciliation Periodic Monitor (C1-RCNM)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the pay instruction which is fully reconciled is changed to **Completed** in the **C1_PAY_INS** table. However, if a pay instruction is not fully reconciled, the status of the pay instruction is changed to **Open** in the **C1_PAY_INS** table.

If all pay instructions in the reconciliation are fully reconciled, the status of the reconciliation is changed to **Completed** in the **C1_RECONCILIATION** table. However, if one or more pay instructions in the reconciliation are not fully reconciled, the status of the reconciliation is changed to **Open** in the **C1_RECONCILIATION** table.

A log entry which is generated for the reconciliation and pay instruction during the status transition is added in the **C1_RECONCILIATION_LOG** and **C1_PAY_INS_LOG** tables, respectively. In addition, if an error occurs while reconciling a pay instruction, it is recorded in the **C1_PAY_INS_LOG** table.

19.3   Reconciliation Cancelation Periodic Monitor (C1-RCNMD)

The **Reconciliation Cancelation Periodic Monitor (C1-RCNMD)** batch is used to monitor or check whether there is any reconciliation in the **Pending Cancelation** status. If there is a reconciliation in the **Pending Cancelation** status, the status of the reconciliation is changed to **Canceled**. In addition, the status of all pay instructions in the reconciliation is changed to **Canceled**.

This batch is a multi-threaded batch. The multi-threading is based on reconciliation ID and chunks for multi-threading are created based on numerical distribution of reconciliation ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor reconciliations which are created using the business objects of a particular maintenance object. **Note:** By default, the parameter value is set to **C1-RECONCLT**.
Batch Control	No	Used when you want to monitor reconciliations whose current status is linked to the **Reconciliation Cancelation Periodic Monitor (C1-RCNMD)** batch. The valid value is **true**.
Reconciliation Type	No	Used when you want to monitor reconciliations which are created using a particular reconciliation type. **Note:** You must specify a reconciliation type which is already defined in the system.
Business Object	No	Used when you want to monitor reconciliations which are created using a particular business object. Note: You must specify a business object which is already defined in the system.

Parameter Name	Mandatory (Yes or No)	Description
Status	No	Used when you want to monitor reconciliations which are in a particular status.
		Note: This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.

Note: If the **Reconciliation Cancelation Periodic Monitor (C1-RCNMD)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of all reconciliations which are in the **Pending Cancelation** status is changed to **Canceled** in the **C1_RECONCILIATION** table. In addition, the status of all pay instructions in the reconciliation (which is canceled) is changed to **Canceled** in the **C1_PAY_INS** table.

A log entry which is generated for the reconciliation and pay instruction during the status transition is added in the **C1_RECONCILIATION_LOG** and **C1_PAY_INS_LOG** tables, respectively.

19.4 Pay Instruction Periodic Monitor (for Pending Status) (C1-PIPMO)

The **Pay Instruction Periodic Monitor (for Pending Status) (C1-PIPMO)** batch is used to monitor or check whether there are any pay instructions in the **Pending** status. If there is a pay instruction in the **Pending** status, it checks whether the number of pay instructions in the reconciliation exceeds the deferred processing threshold limit. If the number of pay instructions in the reconciliation does not exceed the deferred processing threshold limit, the status of the pay instruction is changed to **Derivation in Progress**. However, if the number of pay instructions in the reconciliation exceeds the deferred processing threshold limit, the status of the pay instruction is changed to **Derivation Pending Processing** which is then transitioned to **Derivation in Progress**. Once the status of the pay instruction is set to **Derivation in Progress**, the algorithms attached to the **Derivation in Progress** status are executed.

Once the derivation and validation process is successful, the status of the pay instruction is changed to **Reconciliation in Progress** and the algorithms attached to the **Reconciliation in Progress** status are executed. If the pay instruction is fully reconciled, the status of the pay instruction is changed to **Completed**. However, if the pay instruction is not fully reconciled, the status of the pay instruction is changed to **Open**.

This batch is a multi-threaded batch. The multi-threading is based on pay instruction ID and chunks for multi-threading are created based on numerical distribution of pay instruction ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Business Object	No	Used when you want to monitor pay instructions which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Status	No	Used when you want to monitor pay instructions which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Pay Instruction Periodic Monitor (for Pending Status) (C1-PIPMO)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the pay instruction which is fully reconciled is changed to **Completed** in the **C1_PAY_INS** table. However, if a pay instruction is not fully reconciled, the status of the pay instruction is changed to **Open** in the **C1_PAY_INS** table.

If all pay instructions in the reconciliation are fully reconciled, the status of the reconciliation is changed to **Completed** in the **C1_RECONCILIATION** table. However, if one or more pay instructions in the reconciliation are not fully reconciled, the status of the reconciliation is changed to **Open** in the **C1_RECONCILIATION** table.

A log entry which is generated for the reconciliation and pay instruction during the status transition is added in the **C1_RECONCILIATION_LOG** and **C1_PAY_INS_LOG** tables, respectively. In addition, if an error occurs while reconciling a pay instruction, it is recorded in the **C1_PAY_INS_LOG** table.

19.5   Pay Instruction Periodic Monitor (for Open Status) (C1-PIMDF)

The **Pay Instruction Periodic Monitor (for Open Status) (C1-PIMDF)** batch is used to monitor or check whether there are any pay instructions in the **Open** status. If there is a pay instruction in the **Open** status, it checks whether the number of pay instructions in the reconciliation exceeds the deferred processing threshold limit. If the number of pay instructions in the reconciliation does not exceed the deferred processing threshold limit, the status of the pay instruction is changed to **Reconciliation in Progress**. However, if the number of pay instructions in the reconciliation exceeds the deferred processing threshold limit, the status of the pay instruction is changed to **Reconciliation Pending Processing** which is then transitioned to **Reconciliation in Progress**. Once the status of the pay instruction is set to **Reconciliation in Progress**, the algorithms attached to the **Reconciliation in Progress** status are executed.

If the pay instruction is fully reconciled, the status of the pay instruction is changed to **Completed**. However, if the pay instruction is not fully reconciled, the status of the pay instruction remains as **Open**.

This batch is a multi-threaded batch. The multi-threading is based on pay instruction ID and chunks for multi-threading are created based on numerical distribution of pay instruction ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Business Object	No	Used when you want to monitor pay instructions which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Status	No	Used when you want to monitor pay instructions which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.

Note: If the **Pay Instruction Periodic Monitor (for Open Status) (C1-PIMDF)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the pay instruction which is fully reconciled is changed to **Completed** in the **C1_PAY_INS** table. However, if a pay instruction is not fully reconciled, the status of the pay instruction remains as **Open** in the **C1_PAY_INS** table.

If all pay instructions in the reconciliation are fully reconciled, the status of the reconciliation is changed to **Completed** in the **C1_RECONCILIATION** table. However, if one or more pay instructions in the reconciliation are not fully reconciled, the status of the reconciliation remains as **Open** in the **C1_RECONCILIATION** table.

A log entry which is generated for the reconciliation and pay instruction during the status transition is added in the **C1_RECONCILIATION_LOG** and **C1_PAY_INS_LOG** tables, respectively. In addition, if an error occurs while reconciling a pay instruction, it is recorded in the **C1_PAY_INS_LOG** table.

19.6   Duplicate Reconciliation (C1-RECDU)

The **Duplicate Reconciliation (C1-RECDU)** batch is used to create a copy of reconciliation which is in the **Canceled** status. The new reconciliation is created in the **Draft** status. The reconciliation is immediately transitioned to the **Send Notification** status and the algorithms attached to the **Send Notification** status are executed. Once the To Do is created, the status of the reconciliation is changed to **Pending**. A copy of each pay instruction is created and its status is set to either **Pending** or **Error** depending on whether it is successfully validated or not.

While copying a reconciliation, a batch job is created using this batch automatically when the number of pay instructions in the reconciliation exceeds the value defined for the **Defer Processing Count** option type in the **C1-RECDUPBH** feature configuration.

This batch is a multi-threaded batch. The multi-threading is based on pay instruction ID and chunks for multi-threading are created based on numerical distribution of pay instruction ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Existing Reconciliation ID	Yes	Used to indicate the reconciliation whose copy you want to create. **Note:** If a batch job is created automatically, the value for this parameter is automatically specified in the batch job.
New Reconciliation ID	No	Used to specify the ID that you want to assign to the new reconciliation on creation. **Note:** If a batch job is created automatically, the value for this parameter is automatically specified in the batch job.

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Duplicate Reconciliation (C1-RECDU)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the records are added in the following tables:

- C1_RECONCILIATION
- C1_PAY_INS

A log entry which is generated for the reconciliation and pay instruction during the status transition is added in the **C1_RECONCILIATION_LOG** and **C1_PAY_INS_LOG** tables, respectively. In addition, if an error occurs while validating a pay instruction, it is recorded in the **C1_PAY_INS_LOG** table.

20. Repricing

This section provides information about the following batches:

- Identify Entities for Repricing (C1-REPC1)
- Process Repricing Request (C1-REPC2)

20.1   Identify Entities for Repricing (C1-REPC1)

This batch can be used in the financial services and insurance domains. Its behavior changes based on the repricing mode specified while executing the batch. If the **Repricing Mode** parameter is set to **ACCT**, this batch considers the records in the **Pending (P)** and **Error (E)** status from the **CI_REPRC_ENTITY_DTL** table. For every record, the system identifies the account or person to whom the price list is assigned. If the price list is assigned to a person, the system fetches all accounts of the person and its child persons. Once the system identifies a set of accounts which are impacted due to change in the price list, it checks whether each account is eligible for repricing. If so, it creates a repricing request for the account, price item, and effective date combination

If the **Repricing Mode** parameter is set to **MEMB**, this batch considers the records in the **Pending (P)** and **Error (E)** status from the **CI_REPRC_ENTITY_DTL** table. For each record, the system invokes the algorithms attached to the following system events of the pricing rule type in the specified order:

- **Bill Group Derivation** – Used to derives the policy, bill group, and parent customer of the bill group.
- **Account and Price Item Derivation** – Used to derive the pricing rules for the price items (which are present in the pricing rule type) on the policy plan (to which the membership belongs).

Once the pricing rules are derived, the system creates a repricing request for the membership, pricing rule, and effective date combination.

This batch is a multi-threaded batch. The multi-threading is based on membership ID and chunks for multi-threading are created based on numerical distribution of membership ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Repricing Mode	Yes	Used to indicate whether you want to create repricing requests for accounts or memberships. The valid values are: • ACCT • MEMB
Membership ID	No	Used when you want to create repricing requests for a particular membership. **Note:** This parameter can be used only when the repricing mode is set to **MEMB**.

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.
Chunk Size	No	Used to specify the number of records you want to process in each work unit.
		Note: If you do not specify any value, by default, the parameter value is set to 2.

Note: If the **Identify Entities for Repricing (C1-REPC1)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the system behaves the following manner:

If the Repricing Mode is set to…	Then…
ACCT	A repricing request for the account, price item, and effective date combination is created in the **CI_REPRC_REQ_DTL** table.
MEMB	A repricing request for the membership, pricing rule, and effective date combination is created in the **CI_REPRC_REQ_DTL** table.

Finally, the status of the corresponding record is set to **Complete (C)** in the **CI_REPRC_ENTITY_DTL** table. If any error occurs while creating a repricing request, the status of the corresponding record is set to **Error (E)** in the **CI_REPRC_ENTITY_DTL** table.

20.2   Process Repricing Request (C1-REPC2)

This batch can be used in the financial services and insurance domains. Its behavior changes based on the repricing mode specified while executing the batch. If the **Repricing Mode** parameter is set to **ACCT**, this batch considers the records in the **Pending (P)** and **Error (E)** status from the **CI_REPRC_REQ_DTL** table. It calculates and persist the rate for the account, price item, and effective date combination in the **CI_PRCE_CALC** table using the **C1-PriceAccount** business service.

If the **Repricing Mode** parameter is set to **MEMB**, this batch considers the records in the **Pending (P)** and **Error (E)** status from the **CI_REPRC_REQ_DTL** table. If the pricing rule in the record is **Age Based**, the system invokes the algorithms attached to the following system events of the pricing rule type in the specified order:

- **Pricing Rule Pre Processing** – Used to identify the eligible member persons for premium calculation
- **Pricing Rule Premium Calculation** – Used to calculate the premium for the membership.

If the pricing rule in the record is **Tier Based**, the system invokes the algorithm attached to the following system event of the pricing rule type:

- **Pricing Rule Premium Calculation** – Used to calculate the premium for the membership.

This batch is a multi-threaded batch. The multi-threading is based on membership ID and chunks for multi-threading are created based on numerical distribution of membership ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Repricing Mode	Yes	Used to indicate whether you want to process repricing requests created for accounts or memberships. The valid values are: • ACCT • MEMB
Account ID	No	Used when you want to process repricing requests of a particular account. **Note:** This parameter can be used only when the repricing mode is set to **ACCT**.
Membership ID	No	Used when you want to process repricing requests of a particular membership. **Note:** This parameter can be used only when the repricing mode is set to **MEMB**.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Chunk Size	No	Used to specify the number of records you want to process in each work unit. **Note:** If you do not specify any value, by default, the parameter value is set to 2.

> **Note:** If the **Process Repricing Request (C1-REPC2)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the system behaves the following manner:

If the Repricing Mode is set to…	Then…
ACCT	The rate is calculated and persisted for the account, price item, and effective date combination in the **CI_PRCE_CALC** table.
MEMB	The premium is calculated for each eligible member person and stored in the **CI_PRCE_CALC_LN** table and the total premium for the membership is stored in the **CI_PRCE_CALC** table.

Finally, the status of the corresponding record is set to **Complete (C)** in the **CI_REPRC_REQ_DTL** table. If any error occurs while calculating the rate or membership premium, the status of the corresponding record is set to **Error (E)** in the **CI_REPRC_REQ_DTL** table.

21. Request

This section provides information about the **Generic Request Periodic Monitor (C1-GENRQ)** batch.

21.1 Generic Request Periodic Monitor (C1-GENRQ)

The **Generic Request Periodic Monitor (C1-GENRQ)** batch is used to monitor or check whether there are any requests in the status where this batch is configured for monitoring the process. If so, the batch transitions the requests to the default next status specified in the lifecycle and executes the algorithm (if any) attached to the default next status.

This batch is a multi-threaded batch. The multi-threading is based on request ID and chunks for multi-threading are created based on numerical distribution of request ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero:

- Thread Count
- Override Nbr Records to Commit

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor requests which are created using the business objects of a particular maintenance object. **Note:** By default, the parameter value is set to **C1-REQUEST**.
Batch Control	Yes	Used when you want to monitor requests whose current status is linked to the **Generic Request Periodic Monitor (C1-GENRQ)** batch. The valid value is **true**.
Request Type	No	Used when you want to monitor requests which are created using a particular request type. **Note:** You must specify a request type which is already defined in the system.
Business Object	No	Used when you want to monitor requests which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.

Parameter Name	Mandatory (Yes or No)	Description
Status	No	Used when you want to monitor requests which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Generic Request Periodic Monitor (C1-GENRQ)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the request is changed to the default next status specified in the lifecycle of the business object. In addition, the algorithms (if any) attached to the default next status are executed in the specified sequence.

22. Security

This section provides information about the **Garble Person Information (C1-GRBPI)** batch.

22.1   Garble Person Information (C1-GRBPI)

The **Garble Person Information (C1-GRBPI)** batch is used to garble the personal and account related information of a customer. It considers only those persons where the **Garble Data (C1-GRBL)** characteristic is set to **Y** and the **Garbling Completed (C1-GBCM)** characteristic is either not defined or set to **N**. Note that the system considers the **Garble Data (C1-GRBL)** characteristic value which is effective on the batch business date.

Once the batch is executed, the personal and account related information of a customer is masked with random characters across all records in the system. In addition, the access group of the person is changed to **C1GB** (which does not have any data access role). This ensures that the data of the person and his or her accounts cannot be accessed from any screens in the system.

This batch is a multi-threaded batch. The multi-threading is based on person ID and chunks for multi-threading are created based on numerical distribution of person ID. You can specify the following parameters while executing this batch:

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. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Garble Data Characteristic Type	Yes	Used to specify the characteristic type which indicates whether the person's data must be garbled. **Note:** You can set the parameter value to either **C1-GRBL** or any other custom predefined characteristic type where the characteristic entity is set to **Person**.

Parameter Name	Mandatory (Yes or No)	Description
Garble Person Characteristics	No	Used to specify the characteristic types of a person whose value you want to garble. **Points to Note:** You must specify a characteristic type where the characteristic entity is set to **Person**. You can specify maximum twenty comma-separated values for this parameter.
Garble Account Characteristics	No	Used to specify the characteristic types of an account whose value you want to garble. **Points to Note:** You must specify a characteristic type where the characteristic entity is set to **Account**. You can specify maximum twenty comma-separated values for this parameter.
Garble Person Identifiers	No	Used to specify the identifier types of a person whose value you want to garble. **Note:** You can specify maximum twenty comma-separated values for this parameter.
Garble Account Identifiers	No	Used to specify the identifier types of an account whose value you want to garble. **Note:** You can specify maximum twenty comma-separated values for this parameter.

Parameter Name	Mandatory (Yes or No)	Description
Garble Upload Request Data (Y or N)	No	Used to indicate whether you want to garble the personal and account related information in the upload requests. The valid values are: • Y • N **Note:** By default, the value is set to **N**.

Note: If the **Garble Person Information (C1-GRBPI)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the data in the following fields is masked with random characters across all records in the system:

- ACCOUNT_NBR
- ACCT_NBR
- ADDRESS1
- ADDRESS1_UPR
- ADDRESS2
- ADDRESS3
- ADDRESS4
- BIRTH_DT
- BROKER_NAME
- CITY
- COUNTRY
- COUNTY
- CUSTOMER_NAME
- EMAILID
- ENTITY_NAME
- ENTITY_NAME1
- ENTITY_NAME2
- ENTITY_NAME3
- GEO_CODE
- HOUSE_TYPE
- IN_CITY_LIMIT
- NAME1
- OVRD_MAIL_NAME1
- OVRD_MAIL_NAME2

- OVRD_MAIL_NAME3
- PER_ID_NBR
- PHONE
- POSTAL
- POSTAL_UPR
- STATE

The characteristics and identifiers of a person and account are garbled as mentioned in the batch parameters. The access group of the person is changed to **C1GB**. In addition, the **Garbling Completed (C1-GBCM)** characteristic is defined for each person whose data is garbled in the system and its value is set to **Y**.

23.   Statement

This section provides information about the following batches:

- Create Statements in CAMT or TWIST Format (CISTMEXT)
- Create Statements (STMPRD)
- Archive Statements (CI_ARCST)
- Download Statements (STMDWLD)

23.1   Create Statements in CAMT or TWIST Format (CISTMEXT)

The **Create Statements in CAMT or TWIST Format (CISTMEXT)** batch is used to generate statements in the CAMT or TWIST format. It generates statements for all those statement constructs whose statement cycle's processing date is earlier than the batch business date and is in the **Pending** status. Depending on the statement route type specified in the statement construct, the system accordingly generates the statement in the CAMT or TWIST format. The system considers the bills that meet the following conditions:

- Bill is in the **Complete** status
- Bill date range fall with the statement cycle date range
- Bill is created for the accounts which are included in the statement construct or Bill contains bill segments created against contracts which are included in the statement construct

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Create Statements in CAMT or TWIST Format (CISTMEXT)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the system generates the statements in the CAMT or TWIST format and stores them in the database. In addition, the **STMDWLD** batch control and incremental batch run number is stamped on each statement.

23.2　Create Statements (STMPRD)

The **Create Statements (STMPRD)** batch is used to generate statements in the pre-defined custom format. It generates statements for all those statement constructs whose statement cycle's processing date is earlier than the batch business date and is in the **Pending** status. The system generates the statement in the custom format as designed by the implementation team. The system considers the bills that meet the following conditions:

- Bill is in the **Complete** status
- Bill date range fall with the statement cycle date range
- Bill is created for the accounts which are included in the statement construct or Bill contains bill segments created against contracts which are included in the statement construct

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Create Statements (STMPRD)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the system generates the statements in the custom format and stores them in the database. In addition, the **STMDWLD** batch control and incremental batch run number is stamped on each statement.

23.3   Archive Statements (CI_ARCST)

The **Archive Statements (CI_ARCST)** batch is used to archive the data extracted for the statements in the database. This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Target Archive Environment	Yes	Used to specify the environment where you want to archive the data.
Maximum errors allowed before process termination	No	Used to override the maximum number of errors after which the batch must be terminated.

Note: If the **Archive Statements (CI_ARCST)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the system archives the data extracted for the statements in the specified database environment.

23.4   Download Statements (STMDWLD)

The **Download Statements (STMDWLD)** batch is used to download the data extracted for a statement in a flat file. It downloads the statements with the latest batch run number. This flat file can be interfaced to your bill print software, or whatever mechanism you want to use for sending statements to the appropriate persons.

The information that is extracted and placed on the flat file for each statement is controlled by the respective statement construct's statement route type's extract algorithm. This batch is a multi-threaded batch. The multi-threading is based on the statement ID and the chunks for multi-threading are created based on the numerical distribution of the statement ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Output Directory	No	Used to specify the path where you want to save the flat file. **Note:** If you do not specify the path, the flat file is, by default, saved in the `sploutput` directory.
Output File Name	No	Used to specify the name which you want to use for the flat file.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated.

Parameter Name	Mandatory (Yes or No)	Description
		Note: You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Download Statements (STMDWLD)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the system downloads the data extracted for a statement in a flat file.

24. Stop-Loss Insurance

This section provides information about the following batches:

- Calculate SSL Reinsurer Charges (C1-RISBC)
- Calculate ASL Reinsurer Charges (C1-RIABC)
- Cancel Reinsurer Charges (C1-CNREC)

24.1   Calculate SSL Reinsurer Charges (C1-RISBC)

The **Calculate SSL Reinsurer Charges (C1-RISBC)** batch is used to create reinsurer charges for specific stop-loss. If you have configured reinsurer for specific stop-loss, you need to execute this batch after executing specific stop-loss accumulation batches.

This batch considers the SSL accumulation records from the **CI_ACCUM_SSL_AMT** table where the reinsurer status is set to **P** and the window start date is earlier than or equal to batch business date. It creates reinsurer billable charge with opposite amount on the reinsurer account which is specified in the respective SSL pricing rule and stamps the reinsurer billable charge ID in the **C1_ACCUM_REINSR_SSL_AMT** table. It stamps the parent level accumulation group ID and bill group level accumulation group ID as the characteristics on the reinsurer billable charges.

This batch is a multi-threaded batch. The multi-threading is based on the bill group level accumulation group ID and chunks for multi-threading are created based on numerical distribution of bill group level accumulation group ID. You can specify the following parameter while executing this batch:

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Calculate SSL Reinsurer Charges (C1-RISBC)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the system creates a reinsurer billable charge with opposite amount on the reinsurer account and stamps the reinsurer billable charge ID in the **C1_ACCUM_REINSR_SSL_AMT** table. It also stamps the parent level accumulation group ID and bill group level accumulation group ID as the characteristics on the reinsurer billable charge.

24.2   Calculate ASL Reinsurer Charges (C1-RIABC)

The **Calculate ASL Reinsurer Charges (C1-RIABC)** batch is used to create reinsurer charges for aggregate stop-loss. If you have configured reinsurer for aggregate stop-loss, you need to execute this batch after executing aggregate stop-loss accumulation batches.

This batch considers the ASL accumulation records from the **CI_ACCUM_ASL_AMT** table where the reinsurer status is set to **P** and the window start date is earlier than or equal to batch business date. It creates reinsurer billable charge with opposite amount on the reinsurer account which is specified in the respective ASL pricing rule and stamps the reinsurer billable charge ID in the **C1_ACCUM_REINSR_ASL_AMT** table. It stamps the parent level accumulation group ID as a characteristic on the reinsurer billable charges.

This batch is a multi-threaded batch. The multi-threading is based on parent level accumulation group ID and chunks for multi-threading are created based on the numerical distribution of parent level accumulation group ID. You can specify the following parameter while executing this batch:

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.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Calculate ASL Reinsurer Charges (C1-RIABC)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the system creates reinsurer billable charge with opposite amount on the reinsurer account and stamps the reinsurer billable charge ID in the **C1_ACCUM_REINSR_ASL_AMT** table. It also stamps the parent level accumulation group ID as a characteristic on the reinsurer billable charge.

24.3   Cancel Reinsurer Charges (C1-CNREC)

The **Cancel Reinsurer Charges (C1-CNREC)** batch is used to cancel the reinsurer charges which are created for specific stop-loss and aggregate stop-loss. If the bill segments are generated for the reinsurer charges in the **Frozen** status, this batch cancels the bill segments and the reinsurer billable charges.

If the bill segments are generated for the reinsurer charges in the **Freezable** status, this batch deletes the bill segments and the reinsurer billable charges. However, if bill segments are not generated for the reinsurer charges, this batch deletes the reinsurer billable charges. In addition, when the reinsurer billable charges are canceled or deleted, this batch removes the reinsurer billable charge ID from the **C1_ACCUM_REINSR_SSL_AMT** or **C1_ACCUM_REINSR_ASL_AMT** table, respectively.

This batch is a multi-threaded batch. The multi-threading is based on billable charge ID and chunks for multi-threading are created based on the numerical distribution of billable charge ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Pricing Rule ID	Yes	Used when you want to cancel the reinsurer charges which are generated through a particular SSL or ASL pricing rule. **Note:** You can specify maximum fifteen comma-separated values for this parameter.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Cancel Reinsurer Charges (C1-CNREC)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the system either cancels or deletes the reinsurer billable charge which is created for specific stop-loss or aggregate stop-loss. It also sets the **REINSR_STATUS** column corresponding to the record in the **CI_ACCUM_SSL_AMT** or **CI_ACCUM_ASL_AMT** table to **P**.

25. Upload Request

This section provides information about the **Upload Request Periodic Monitor (C1-UPLRQ)** batch.

25.1   Upload Request Periodic Monitor (C1-UPLRQ)

The **Upload Request Periodic Monitor (C1-UPLRQ)** batch is used to monitor or check upload requests or adjustment upload requests depending on whether the **Maintenance Object** parameter is set to **C1-UPLOADREQ** or **C1-UPLREQ**. The following table describes the batch in detail:

If the Maintenance Object parameter is set to…	Then…
C1-UPLOADREQ	The **Upload Request Periodic Monitor (C1-UPLRQ)** batch is used to monitor or check whether there are any upload requests in the **Deferred Processing** status. If there is an upload request in the **Deferred Processing** status, the batch changes the status of the upload request to **Processing**. Then, the algorithm attached to the **Processing** status is triggered which creates the entities and accordingly changes the status of the upload request to **Processed**. This batch is a multi-threaded batch. The multi-threading is based on upload request ID and chunks for multi-threading are created based on numerical distribution of upload request ID. However, we recommend you to execute this batch in the single-threaded mode. Otherwise, erroneous results might occur. While executing this batch in the single-threaded mode, the following fields must be set to zero: • Thread Count • Override Nbr Records to Commit
C1-UPLREQ	The **Upload Request Periodic Monitor (C1-UPLRQ)** batch is used to monitor or check whether there are any adjustment upload requests in the **Deferred** status. If there is an adjustment upload request in the **Deferred** status, the batch changes the status of the adjustment upload request to **Creating Adjustment**. Then, the algorithm attached to the **Creating Adjustment** status is triggered which creates the adjustments. This batch is a single-threaded batch.

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor upload requests or adjustment upload requests created using the business objects of a particular maintenance object. **Points to Note:** If you want to monitor upload requests, you need to set the value of this parameter to **C1-UPLOADREQ**. And, if you want to monitor adjustment upload requests, you need to set the value of this parameter to **C1-UPLREQ**. By default, the parameter value is set to **C1-UPLOADREQ**.
Batch Control	No	Used when you want to monitor upload requests or adjustment upload requests whose current status is linked to the **Upload Request Periodic Monitor (C1-UPLRQ)** batch. The valid value is **true**.
Business Object	No	Used when you want to monitor upload requests or adjustment upload requests which are created using a particular business object. **Note:** You must specify a business object which is already defined in the system.
Status	No	Used when you want to monitor upload requests or adjustment upload requests which are in a particular status. **Note:** This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.

Parameter Name	Mandatory (Yes or No)	Description
Upload Request Type	No	Used when you want to monitor upload requests or adjustment upload requests which are created using a particular upload request type. **Note:** You must specify a upload request type which is already defined in the system.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Upload Request Periodic Monitor (C1-UPLRQ)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

The following table explains what happens on successful completion of this batch:

If the Maintenance Object parameter is set to…	Then…
C1-UPLOADREQ	• The status of the upload request is changed from **Deferred Processing** to **Processing**. • The entities are created using the entity business object specified in the respective upload request type. • Finally, the status of the upload request is changed to **Processed**.
C1-UPLREQ	• The status of the adjustment upload request is changed from **Deferred** to **Creating Adjustment**. • The adjustment is created for each adjustment record in the adjustment upload request. • Finally, the status of the adjustment upload request is changed to **Completed**.

26. Data Extractor

This section provides information about the **Data Extract (C1-DTEX)** batch.

26.1   Data Extract (C1-DTEX)

The **Data Extract (C1-DTEX)** batch is used to create a data extract request using the specified data extract request type. The data extract request type indicates the business object whose data you would like to extract at the specified location. The batch then derives the data created using the business object from each table associated with the respective maintenance object. A separate CSV file is created for each table from where the data is extracted.

In the first batch run, the data is fully extracted from the tables irrespective of the data extract type (i.e., full or incremental) specified in the data extract request type. However, in the subsequent batch run, the data is extracted based on the data extract type (i.e., full or incremental) specified in the data extract request type.

This batch is a multi-threaded batch. The multi-threading is based on the primary table's primary key and chunks for multi-threading are created based on numerical distribution of the primary table's primary key. For example, if you are extracting the bill information created using the **C1_BILL_BO** business object, chunks for multi-threading are created based on numerical distribution of the bill ID (i.e., **BILL_ID** field of the **CI_BILL** table). You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Data Extract Request Type Code	Yes	Used to indicate the data extract request type using which you want to create a data extract request. **Note:** You must specify a data extract request type which is already defined in the system.
Override Maximum Errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Chunk Size	No	Used to specify the number of records you want to extract in each work unit. **Points to Note:**

Parameter Name	Mandatory (Yes or No)	Description
		You must specify a positive integer value for this parameter.
		If you do not specify any value, by default, the parameter value is set to **500**.

Note: If the **Data Extract (C1-DTEX)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, all or filtered data (based on the condition) is extracted in the CSV file. A separate CSV file is created for each table from where the data is extracted. For example, if you extract the bill information using the **DEXT** data extract request type where the business object is set to **C1_BILL_BO**, the system will create the following CSV files at the specified location:

- DEXT_BILL_CI_BILL_07-01-2025_243.csv
- DEXT_BILL_CI_BILL_CHAR_07-01-2025_243.csv
- DEXT_BILL_CI_BILL_EXCP_07-01-2025_243.csv
- DEXT_BILL_CI_BILL_MSG_PRM_07-01-2025_243.csv
- DEXT_BILL_CI_BILL_MSGS_07-01-2025_243.csv
- DEXT_BILL_CI_BILL_ROUTING_07-01-2025_243.csv
- DEXT_BILL_CI_BILL_SA_07-01-2025_243.csv
- DEXT_BILL_ERROR_07-01-2025_243.csv

The CSV file includes all the columns from the table including two additional columns named **DT_EXTRACT_DTTM** and **RECORD_STATUS**. The **DT_EXTRACT_DTTM** column contains the date and time when the record is extracted from the table. On full data extraction, the system sets the **RECORD_STATUS** column for each record in the CSV file to **I** (i.e., Add). However, on incremental data extraction, the system sets the **RECORD_STATUS** column for each record in the CSV file to either **I** (i.e., Add) or **U** (i.e., Update) depending on whether the data is added or updated in the system since the last extraction.

If any error occurs while extracting the data, the errors are stored in the following file, which is created along with the CSV files:

<Data_Extract_Request_Type>_<MO>_ERROR_<Extraction_Date>_<Batch_Run_No>.csv

The summary of the extracted files is maintained in the **CI_DATA_EX_REQ** table.

27. Generic Extract from Datasets

This section provides information about the following batches:

- [Generic Extract for Dataset (C1-GENEX)](#)
- [Generic Extract for Data Load (C1-GENLD)](#)
- [Generic Extract Dataset Extraction (C1-GENDS)](#)

27.1 Generic Extract for Dataset (C1-GENEX)

The **Generic Extract for Dataset (C1-GENEX)** batch is used to extract data from the predefined dataset using an extraction template. It enables you to extract data in a file format supported by the extraction template. It enables you to specify the data explorer zone for filtering the data during extraction. If the query in the data explorer zone returns multiple records, the batch extracts the data of each record in a separate file. However, if the query in the data explorer zone returns a single record, the batch extracts the data of the record in a single file. Note that the batch creates each file in a separate thread when multiple records are returned in the result set. Otherwise, the batch is executed in a single-threaded mode.

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Template Code	Yes	Used to specify the extraction template using which you want to extract data from the predefined dataset. **Note:** You must specify an extraction template which is already defined in the system.
File Path	Yes	Used to specify the directory or bucket where you want to store the extracted file. If the application environment is an On Premises or Cloud Service environment, you can specify the directory name along with the following relative path: - **@SHARED_DIR** – Used when you want to store the extracted file in the shared directory on the server. For example, you can set the value to @SHARED_DIR/output. However, if the application environment is a Cloud Service, Premium Edition environment, you must specify the following information in the file-storage://<File Storage Extendable Lookup Value>/<Object Storage Bucket> format:

Parameter Name	Mandatory (Yes or No)	Description
		• The file storage extendable lookup value from where the cloud storage compartment and connection details are retrieved while storing the extracted data on OCI. • The bucket or container where you want the system to store the extracted file.
		Points to Note: The shared directory variable (i.e., **SHARED_DIR**) is case-sensitive. If the specified directory does not exist in the shared directory, the system will create the directory with the given name in the specified location.
File Name	Yes	Used to specify the name for the file in which you want to extract the data.
File Format	Yes	Used to indicate the file format in which you want to extract the data. The valid values are: • CSV • XML • PRN • JSON • PSV • TILD **Note:** You can only specify a file format that is supported by the template.
Zone Code	No	Used to specify the data explorer zone designed for the respective dataset. The batch then generates separate output files when multiple records are returned by the query defined in the zone. **Points to Note:** In the data explorer zone, you can define a SELECT statement to retrieve data from a table or view based on a condition. You can use the bind variables in the SELECT statement. The bind variables must be alphanumeric and prefixed with the **Colon** (:) symbol. The values for these bind variables can be either defaulted in the query or specified while executing the batch.

Parameter Name	Mandatory (Yes or No)	Description
Binding Parameter Names	No	Used to specify the bind variables that are used in the query of the data explorer zone.
		Points to Note:
		If you want to specify multiple bind variables, you need to separate the bind variables using the **Tilde (~)** symbol. For example,
		CIS_DIVISION~START_DATE~DATE_FORMAT.
		You can specify the date format through a separate bind variable.
Binding Parameter Values	No	Used to specify value for the bind variables used in the query of the data explorer zone.
		Note: If you want to specify multiple bind variable values, you need to separate the bind variable values using the **Tilde (~)** symbol. For example,
		CALIFORNIA~01-JAN-2000~DD-MON-YEAR.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.
		Note: If you do not specify any value, by default, it is set to **DEFAULT**.

Note: If the **Generic Extract for Dataset (C1-GENEX)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the data is extracted in the specified file format at the specified location. If the file format is set to **XML** or **JSON**, the system uses the XSL file specified in the data extraction template to generate the output in the XML or JSON format.

27.2 Generic Extract for Data Load (C1-GENLD)

The **Generic Extract for Data Load (C1-GENLD)** batch is used to load data in the target dataset tables. You can load the data for a single, multiple or all the modules at once. You need to schedule this batch to run at regular intervals to ensure that the target dataset tables are up-to-date. This batch always loads full data in the target dataset tables.

Note: A full load is a data loading strategy in which the entire dataset from a source system is transferred to a target data warehouse, replacing any data previously present.

This batch is a multi-threaded batch. It loads the data for each module in a separate thread when datasets of multiple modules are loaded at once. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Dataset Name	No	Used to indicate the module for which you want to load the dataset. The valid values are: • **ACCT** - Used when you want to load data for the accounts. • **ADJ** - Used when you want to load data for the adjustments. • **ASLD** - Used when you want to load data for the aggregate stop-loss. • **BILL** - Used when you want to load data for the bills. • **BLGR** - Used when you want to load data for the billing groups. • **BSEG** - Used when you want to load data for the bill segments. • **CUST** - Used when you want to load data for the customers. • **DELN** - Used when you want to load data for the delinquency processes. • **FT** - Used when you want to load data for the financial transactions. • **GRPM** - Used when you want to load data for the group memberships. • **INDM** - Used when you want to load data for the individual memberships. • **MMPR** - Used when you want to load data for the member persons. • **PAY** - Used when you want to load data for the payments. • **POHL** - Used when you want to load data for the policy holders. • **SSLD** - Used when you want to load data for the specific stop-loss. • **STAC** - Used when you want to load data for the account statements. • **STAD** - Used when you want to load data for the adjustment statements. • **STCU** - Used when you want to load data for the customer statements. • **STSS** – Used when you want to load data for the statement services.

Parameter Name	Mandatory (Yes or No)	Description
		• **TXND** - Used when you want to load data for the transactions. • **ALL** - Used when you want to load data for all the above at once. **Note:** If you do not specify any value, by default, it is set to **ALL**.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Email Mode	No	Used to indicate whether you want to send the email notification on the batch job completion depending on the batch job run status. The valid values are: • **ERROR** – Used when you want to send the email notification on the batch job completion only when the batch job run is in the **Error** status. • **SUCCESS** - Used when you want to send the email notification on the batch job completion only when the batch job run is in the **Complete** status. • **ALL** - Used when you want to send the email notification on the batch job completion irrespective of the batch job run status. **Points to Note:** If you do not specify any value, by default, it is set to **ALL**. This parameter is applicable only when the email address is specified in the batch job.
Override maximum errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.

Note: If the **Generic Extract for Data Load (C1-GENLD)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the data is loaded in the predefined datasets (i.e., target dataset tables).

27.3   Generic Extract Dataset Extraction (C1-GENDS)

The **Generic Extract Dataset Extraction (C1-GENDS)** batch is used to extract data directly from the predefined datasets without using an extraction template. You can extract the data of a single, multiple or all the modules at once. You can extract full or incremental data from the target dataset tables in the Comma Separated Values (CSV) format.

This batch is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Dataset Name	No	Used to indicate the module for which you want to extract the data. The valid values are: • **ACCT** - Used when you want to extract the account data. • **ADJ** - Used when you want to extract the adjustment data. • **ASLD** - Used when you want to extract the aggregate stop-loss data. • **BILL** - Used when you want to extract the bill data. • **BLGR** - Used when you want to extract the billing group data. • **BSEG** - Used when you want to extract the bill segment data. • **CUST** - Used when you want to extract the customer data. • **DELN** - Used when you want to extract the delinquency process data. • **FT** - Used when you want to extract the financial transaction data. • **GRPM** - Used when you want to extract the group membership data. • **INDM** - Used when you want to extract the individual membership data. • **MMPR** - Used when you want to extract the member person data. • **PAY** - Used when you want to extract the payment data. • **POHL** - Used when you want to extract the policy holder data. • **SSLD** - Used when you want to extract the specific stop-loss data.

Parameter Name	Mandatory (Yes or No)	Description
		• **STAC** - Used when you want to extract the account statement data. • **STAD** - Used when you want to extract the adjustment statement data. • **STCU** - Used when you want to extract the customer statement data. • **STSS** – Used when you want to extract the statement service data. • **TXND** - Used when you want to extract the transaction data. • **ALL** - Used when you want to extract all the above data at once.
		Note: If you do not specify any value, by default, it is set to **ALL**.
File Path	Yes	Used to specify the directory or bucket where you want to store the extracted file. If the application environment is an On Premises or Cloud Service environment, you can specify the directory name along with the following relative path: • **@SHARED_DIR** – Used when you want to store the extracted file in the shared directory on the server. For example, you can set the value to @SHARED_DIR/output. However, if the application environment is a Cloud Service, Premium Edition environment, you must specify the following information in the file-storage://<File Storage Extendable Lookup Value>/<Object Storage Bucket> format: • The file storage extendable lookup value from where the cloud storage compartment and connection details are retrieved while storing the extracted data on OCI. • The bucket or container where you want the system to store the extracted file.
		Points to Note: The shared directory variable (i.e., **SHARED_DIR**) is case-sensitive.

Parameter Name	Mandatory (Yes or No)	Description
		If the specified directory does not exist in the shared directory, the system will create the directory with the given name in the specified location.
Data Extract Flag	Yes	Used to indicate whether you want to extract full or incremental data from the target dataset tables. The valid values are: • FULL • INCR
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch. **Note:** If you do not specify any value, by default, it is set to **DEFAULT**.
Email Mode	No	Used to indicate whether you want to send the email notification on the batch job completion depending on the batch job run status. The valid values are: • **ERROR** – Used when you want to send the email notification on the batch job completion only when the batch job run is in the **Error** status. • **SUCCESS** - Used when you want to send the email notification on the batch job completion only when the batch job run is in the **Complete** status. • **ALL** - Used when you want to send the email notification on the batch job completion irrespective of the batch job run status. **Points to Note:** If you do not specify any value, by default, it is set to **ALL**. This parameter is applicable only when the email address is specified in the batch job.
Override maximum errors	No	Used to override the maximum number of errors allowed before the batch run is terminated. **Note:** You must specify a value greater than or equal to zero.

Note: If the **Generic Extract Dataset Extraction (C1-GENDS)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check/Clean Up:

On successful completion of this batch, the data is extracted in the CSV format at the specified location. The file is named as {tablename}.csv