Oracle® Revenue Management and Billing

Version 2.2.4.4

Transaction Feed Management (TFM) - Batch Execution Guide

Revision 2.0

E50304-01

October, 2013

ORACLE®

Oracle Revenue Management and Billing Transaction Feed Management (TFM) - Batch Execution Guide

E50304-01

Third Party Content, Products, and Services Disclaimer

This software and documentation may provide access to or information on 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. 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.

Preface

About This Document

This document will help you to understand the sequence in which the batches should be executed while performing various tasks in TFM. It also helps you to improve the batch performance.

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 chapters:

Chapter No.	Chapter Name	Chapter Description
Chapter 1	Introduction	Provides an overview of the transaction feed management process.
Chapter 2	TFM Batch Execution Sequence	Explains the sequence in which the batches should be executed while performing various tasks in TFM.
Chapter 3	Recommended Parameter Values	Recommends parameter values for each batch.

Related Documents

You can refer to the following documents for more information:

Document	Description
Oracle Revenue Management and Billing Banking User Guide	Lists and describes various banking features in Oracle Revenue Management and Billing. It also describes all screens related to these features and explains how to perform various tasks in the application.

Contents

1. Introduction

Oracle Revenue Management and Billing provides you with a facility to upload banking transactions received from various product processors or banking applications for billing. Once the transaction data is uploaded in the system, you need to:

- Validate Header Details
- Validate Transaction Details and Determine Initial Product
- Verify Product Pricing
- Create and Update Billable Charge with the SQI values
- Clean-up Unwanted Data

The transaction feed management process includes various sub-processes, such as Header Validation, Transaction Validation and Initial Product Determination, Product Pricing Verification, Aggregation, Clean Up, Disaggregation, Cancellation, and Rollback. You can execute each sub-process through a batch.

2. TFM Batch Execution Sequence

The following table indicates the sequence in which the batches should be executed while performing various tasks in TFM:

TFM Task	Batch Sequence
Aggregation	Execute the following batches in the specified order: 1. Flush All (F1-Flush) 2. Header Validation (C1-TXNHV) 3. Transaction Validation and Initial Product Determination (C1-TXNIP) 4. Product Pricing Verification (C1-TXNVP) 5. Update Status (C1-TXNEX) 6. Service Quantity Calculation (C1-TXNSQ) 7. Mark Completion (C1-TXNCM) 8. Clean Up (C1-TXNCU) (with the **Request Type** parameter set to **EROR**) **Note:** The **Header Validation (C1-TXNHV)** batch is optional. You can directly execute the **Initial Product Determination (C1-TXNIP)** batch once the transactions are uploaded in the system. If you change the TFM configuration while executing these batches, you need to execute the **Disaggregation (C1-TXNDA)** batch. Once you disaggregate transactions, you must execute all above listed batches (from F1-Flush to C1-TXNCU) once again in the specified order to complete the aggregation process.
Rollback Transactions with the Error (EROR) or Ignored (IGNR) Status	Execute the Rollback (C1-TXNRB) batch.
Disaggregation	Execute the following batches in the specified order: 1. Disaggregation Entry (C1-DISTG) 2. Disaggregation (C1-TXNDA)
Cancellation	Execute the following batches in the specified order: 1. Bill Deletion (C1-DELBL) 2. Clean Up (C1-TXNCU) (with the **Request Type** parameter set to **CNCL**) 3. Cancellation (C1-TXCNC)

2.1 Rollback Batch (C1-TXNRB)

The **Rollback (C1-TXNRB)** batch is used to rollback transactions with the **Error (EROR)** or **Ignored (IGNR)** status. Once a transaction is roll backed, its status is changed to **Uploaded (UPLD)**.

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Count	Yes	Used to specify the number of threads you want to spawn in parallel.
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.
Transaction Status	Yes	Used to indicate whether you want to rollback transactions which were ignored or where an error occurred. The valid values are: • IGNR • EROR
Transaction Header ID	No	Used when you want to rollback transactions of a particular feed.
Transaction Source	No	Used when you want to rollback transactions received from a particular transaction source.
Division	No	Used when you want to rollback transactions belonging to a particular division.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of transactions which are roll backed is changed to **UPLD**. The corresponding data is deleted from the CI_TXN_DTL_PRITM table.

2.2 Flush All Batch (F1-Flush)

The **Flush All (F1-Flush)** batch is used to clean the application cache. It is a single-threaded batch. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Pool	No	Used to specify the thread pool whose cache you want to clean.

Post Execution Check/Clean Up:

On successful completion of this batch, the cache would be cleaned completely.

2.3 Header Validation Batch (C1-TXNHV)

The **Header Validation (C1-TXNHV)** batch is used to validate the file or header level information of transactions.

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Transaction Header ID	No	Used when you want to validate a particular transaction feed.
Transaction Source	No	Used when you want to validate transaction feeds received from a particular transaction source.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the feed is changed to **Valid (VALI)** and the status of all transactions in the feed remains as **UPLD**. If header validation fails, the status of the feed and all transactions in the feed is changed to **EROR**. Check the status of the feed in the CI_TXN_HEADER table, and the status of all transactions in the feed in the CI_TXN_DETAIL table.

2.4 Transaction Validation and Initial Product Determination Batch (C1-TXNIP)

The **Transaction Validation and Initial Product Determination** (C1-TXNIP) batch is used to validate transaction level information and then map each transaction to either of the following:

- One or more product
- One or more product and TOU (variance parameter) combination

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Count	Yes	Used to specify the number of threads you want to spawn in parallel.
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
Transaction Header ID	No	Used when you want to validate and derive product for transactions of a particular feed.
Transaction Source	No	Used when you want to validate and derive product for transactions received from a particular transaction source.
Division	No	Used when you want to validate and derive product for transactions belonging to a particular division.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check/Clean Up:

Do the following:

1. Check the status of transactions in the CI_TXN_DETAIL table. The status of transactions should be **Initial Product Determined (INPD)**, **EROR** or **IGNR**.
2. Check the CI_TXN_DTL_PRITM table to verify whether the transactions are processed and inserted in this table.

Note: For transactions in the **EROR** or **IGNR** status, there will be no records in the CI_TXN_DTL_PRITM table.

It is recommended that you execute DBMS Stats to collect statistics about the data entered in the CI_TXN_DTL_PRITM table. To execute DBMS Stats on the CI_TXN_DTL_PRITM table, use the following stored procedure:

```
BEGIN
DBMS_STATS.GATHER_TABLE_STATS(ownname=>'CISADM',
tabname=>'CI_TXN_DTL_PRITM', degree => 192);
END;
```

2.5 Product Pricing Verification Batch (C1-TXNVP)

The **Product Pricing Verification (C1- TXNVP)** batch is used to check whether effective pricing is available for the account, product and/or TOU combination (on the batch business date). You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Count	Yes	Used to specify the number of threads you want to spawn in parallel.
Chunk Size	Yes	Used to specify the number of transactions you want to execute in each work unit.

Parameter Name	Mandatory (Yes or No)	Description
Transaction Header ID	No	Used when you want to verify product pricing for transactions in a particular feed.
Transaction Source	No	Used when you want to verify product pricing for transactions received from a particular transaction source.
Division	No	Used when you want to verify product pricing for transactions belonging to a particular division.
Pricing Level	No	Used when you want to verify product pricing for transactions only at a particular level. The valid values are: • ACCOUNT_AGREED • CUSTOMER_AGREED • PARENT_CUSTOMER_AGREED • ACCOUNT_PRICELIST • ACCOUNT_INHERITED_PRICELIST • CUSTOMER_PRICELIST • CUSTOMER_INHERITED_PRICELIST • PARENT_CUSTOMER_PRICELIST • PARENT_CUSTOMER_INHERITED_PRICELIST **Note:** In this case, the system does not search for the product or bundle pricing at all levels defined in the price assignment search order. Instead, the system searches for the product or bundle pricing at the specified level. If the product or bundle pricing is not available at the specified level, the status of the transaction is changed to **Error (EROR)**.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check/Clean Up:

On successful execution of this batch, records are added in the CI_TXN_DTL_PRITM_SUMMARY table. The summary record contains pricing information for account, product, TOU, aggregation start and end date combination.

2.6 Update Status Batch (C1-TXNEX)

The **Update Status (C1-TXNEX)** batch is used to update the status of the transaction. If a transaction is ignored and not considered for billing for all products (to which it is mapped), the status of the transaction is changed to **IGNR**. However, if the effective pricing is not available for one or more products to which a transaction is mapped, the status of the transaction is changed to **EROR**.

Also, when there is no contract with the specified contract type available in the system or when the contract is inactive, the status of the transaction is changed to **EROR**.

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Count	Yes	Used to specify the number of threads you want to spawn in parallel.
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.
Transaction Header ID	No	Used when you want to change the status of transactions of a particular feed.
Transaction Source	No	Used when you want to change the status of transactions received from a particular transaction source.
Division	No	Used when you want to change the status of transactions belonging to a particular division.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check/Clean Up:

On successful execution of this batch, transactions which are ignored and for which effective pricing could not be determined are marked as **IGNR** and **EROR**, respectively, in the CI_TXN_DETAIL table. Check the CI_TXN_DTL_PRITM_SUMMARY table and the corresponding records in the CI_TXN_DETAIL and CI_TXN_DTL_PRITM table.

2.7 Service Quantity Calculation Batch (C1-TXNSQ)

The **Service Quantity Calculation (C1-TXNSQ)** batch is used to aggregate the transactions, create billable charges, and then update the SQI values in the billable charges.

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Count	Yes	Used to specify the number of threads you want to spawn in parallel.
Chunk Size	Yes	Used to specify the number of transactions you want to execute in each work unit.

Parameter Name	Mandatory (Yes or No)	Description
Maximum Batch Count	Yes	Used to specify the maximum number of transactions after which the data must be committed in the database.
Transaction Header ID	No	Used when you want to create billable charges for transactions in a particular feed.
Transaction Source	No	Used when you want to create billable charges for transactions received from a particular transaction source.
Division	No	Used when you want to create billable charges for transactions belonging to a particular division.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Note:

Once the **Service Quantity Calculation (C1-TXNSQ)** batch is executed, you must execute the **Mark Completion (C1-TXNCM)** and **Clean Up (C1_TXNCU)** batches. Even if the **Service Quantity Calculation (C1-TXNSQ)** batch fails, you must execute the **Mark Completion (C1-TXNCM)** and **Clean Up (C1_TXNCU)** batches.

The **Clean Up (C1_TXNCU)** batch must be executed with the **Request Type** parameter set to **EROR**.

Post Execution Check/Clean Up:

On successful completion of this batch, billable charges are created and added in the CI_BILL_CHG and CI_BILL_CHG_K tables. The corresponding SQIs are added in the CI_BCHG_SQ table.

If the **Aggregate Transaction** field is set to **Yes**, the billable charge ID is updated in the CI_TXN_DTL_PRITM_SUMMARY table corresponding to the transaction. However, if the **Aggregate Transaction** field is set to **No**, the billable charge ID is updated in the CI_TXN_DTL_PRITM table corresponding to the transaction. In addition, if the **Aggregate Transaction** field is set to **Yes**, the status of the records in the CI_TXN_DTL_PRITM_SUMMARY table is changed to **C**.

2.8 Mark Completion Batch (C1-TXNCM)

The **Mark Completion (C1-TXNCM)** batch is used to update the status of the transaction. If the SQI values are updated successfully in the billable charge, the status of the transaction is changed to **Completed (COMP)**. But, if the SQIs are not defined for the product — division combination, the transaction aggregation rule is not defined for the SQI, or if the exchange rate is not available during currency conversion, the status of the transaction is changed to **Error (EROR)**.

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Count	Yes	Used to specify the number of threads you want to spawn in parallel.
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.
Transaction Header ID	No	Used when you want to change the status of transactions of a particular feed.
Transaction Source	No	Used when you want to change the status of transactions received from a particular transaction source.
Division	No	Used when you want to change the status of transactions belonging to a particular division.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the transactions for which billable charges are successfully created and updated is changed to **COMP**. If any error occurs while creating or updating billable charges, the status of the transaction is changed to **EROR**. Check the status of the transactions in the CI_TXN_DETAIL table.

In addition, for aggregated transactions, the billable charge ID is updated in the CI_TXN_DTL_PRITM table.

2.9 Clean Up Batch (C1-TXNCU)

The **Clean Up (C1-TXNCU)** batch is used to delete the billable charges, if any, created for transactions in the **EROR** status. It also recalculates SQIs in a billable charge if the transactions in the **EROR** and **COMP** status are aggregated together in the billable charge.

This batch is also used during the cancellation process. It deletes non aggregated billable charges and recalculates SQIs in a billable charge if the transactions which are cancelled are aggregated in the billable charge.

You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Transaction Header ID	No	Used when you want to recalculate SQIs for transactions in a particular feed.

Parameter Name	Mandatory (Yes or No)	Description
Transaction Source	No	Used when you want to recalculate SQIs for transactions received from a particular transaction source.
Division	No	Used when you want to recalculate SQIs for transactions belonging to a particular division.
Request Type	Yes	Used to indicate whether you want to recalculate SQIs during the cancellation process or when the transactions are in the **Error (EROR)** status. The valid values are: • CNCL • EROR
Thread Count	Yes	Used to specify the number of threads you want to spawn in parallel.
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.

Post Execution Check/Clean Up:

On successful completion of this batch, billable charge records are either updated or deleted from the CI_BILL_CHG, CI_BILL_CHG_K, and CI_BCHG_SQ tables. In addition, while doing cleanup for error transactions, the status of the records in the CI_TXN_DTL_PRITM_SUMMARY table is changed to U when the SQIs on the corresponding billable charge are recalculated for aggregated transactions.

And, during cleanup of cancelled transactions, the billable charges for non aggregated transactions are deleted from the CI_BILL_CHG, CI_BILL_CHG_K, and CI_BCHG_SQ tables. The transactions which are cancelled are deleted from the CI_TXN_DTL_PRITM table.

2.10 Billing Batch (BILLING)

The **Billing (BILLING)** batch is used to generate bills based on the bill cycle defined for the account. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Count	Yes	Used to specify the number of threads you want to spawn in parallel.

Parameter Name	Mandatory (Yes or No)	Description
Maximum Number of Errors	No	Used to terminate the batch when the errors exceed the specified limit. In such case, changes are not committed to the database.
Off Cycle Billing	No	Used to indicate whether you want to generate off cycle bills. The valid values are: • Y • N **Note:** If you do not set the value of this parameter, the system will automatically set the value of this parameter to **N** during the batch job submission.
Division	No	Used to specify the division for which you want to generate the bills.
Bill Cycle	No	Used to specify the bill cycle for which you want to generate the bills.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check/Clean Up:

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

2.11 Disaggregation Entry Batch (C1-DISTG)

The **Disaggregation Entry (C1-DISTG)** batch is used to add an appropriate entry in the **Disaggregation** table. 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.
Division	No	Used when you want to create disaggregate transaction entry for the accounts belonging to a particular division.
Bill Cycle	No	Used when you want to create disaggregate transaction entry for the accounts having the specified bill cycle.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check/Clean Up:

On successful completion of this batch, accounts (account ID) whose transactions need to be disaggregated are added in the CI_TXN_DISAGG_REQ table. For each such account, the BO_STATUS_CD is set to PENDING.

2.12 Disaggregation Batch (C1-TXNDA)

The **Disaggregation (C1-TXNDA)** batch reads entries in the **Disaggregation** table and then disaggregates the transactions.

Pre-requisites:

Accounts whose transactions need to be disaggregated must be present in the CI_TXN_DISAGG_REQ table.

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.
Division	No	Used when you want to disaggregate transactions whose charges are borne by the accounts belonging to a particular division.
Account ID	No	Used when you want to disaggregate transactions whose charges are borne by the specified account.
Bill Cycle	No	Used when you want to disaggregate transactions whose charges are borne by the accounts having the specified bill cycle.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Note: If you have specified the **Bill Cycle** parameter while executing the **Disaggregation Entry (C1-DISTG)** batch, you will have to specify the same parameter while executing the **Disaggregation (C1-TXNDA)** batch.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the transaction is changed to **UPLD** in the CI_TXN_DETAIL table. The Billable Charge ID corresponding to the disaggregated transaction is removed from the CI_TXN_DTL_PRITM table.

The bills are deleted from the CI_BILL and CI_BILL_K tables. The corresponding bill segments are deleted from the CI_BSEG, CI_BSEG_K, CI_BSEG_ITEM, CI_BSEG_SQ, CI_BSEG_READ, CI_BSEG_MSG, CI_BSEG_EXCP, CI_BSEG_CL_CHAR, CI_BSEG_CALC, and CI_BSEG_CALC_LN tables. The corresponding billable charges are deleted from the CI_BILL_CHG, CI_BILL_CHG_K, and CI_BCHG_SQ tables.

The financial transactions (FTs) created corresponding to the bills (which are deleted) are also deleted from the CI_FT, CI_FT_GL, CI_FT_K, CI_FT_PROC, and CI_FT_GL_EXT tables.

2.13 Bill Deletion Batch (C1-DELBL)

The **Bill Deletion (C1-DELBL)** batch is used to delete the bills (with the Pending status) and their corresponding bill segments. This batch is used during the cancellation process. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Count	Yes	Used to specify the number of threads you want to spawn in parallel.
Chunk Size	Yes	Used to specify the number of transactions you want to execute in each work unit.
Max Batch Count	Yes	Used to specify the maximum number of transactions after which the data must be committed in the database.
Transaction Header ID	Yes	Used when you want to delete bills which are created for transactions in a particular transaction feed.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Post Execution Check/Clean Up:

On successful completion of this batch, the bills are deleted from the CI_BILL and CI_BILL_K tables. The corresponding bill segments are deleted from the CI_BSEG, CI_BSEG_K, CI_BSEG_ITEM, CI_BSEG_SQ, CI_BSEG_READ, CI_BSEG_MSG, CI_BSEG_EXCP, CI_BSEG_CL_CHAR, CI_BSEG_CALC, and CI_BSEG_CALC_LN tables. The financial transactions (FTs) created corresponding to the bills (which are deleted) are also deleted from the CI_FT, CI_FT_GL, CI_FT_K, CI_FT_PROC, and CI_FT_GL_EXT tables.

2.14 Cancellation Batch (C1-TXCNC)

The **Cancellation (C1-TXCNC)** batch is used to change the status of the feed and all transactions in the feed to **Cancelled (CNCL)**. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Thread Count	Yes	Used to specify the number of threads you want to spawn in parallel.
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.
Transaction Header ID	Yes	Used when you want to cancel a particular transaction feed.

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.

Post Execution Check/Clean Up:

On successful completion of this batch, the status of the feed is changed to **Cancelled (CNCL)** in the CI_TXN_HEADER table. The status of all transactions in the feed is changed to **Cancelled (CNCL)** in the CI_TXN_DETAIL table. The corresponding data is deleted from the CI_TXN_DTL_PRITM table.

Note: If any transaction in the feed which you want to cancel belongs to a frozen bill, the system will not cancel the feed.

3. Recommended Parameter Values

This section recommends parameter values for each batch. The actual values to achieve maximum performance will vary with different hardware set. The recommendations are based on the number of CPUs and RAM available on the database and application server. The actual performance would depend on the number of CPUs and RAM available on the application server, and many other hardware parameters. Oracle Revenue Management and Billing provides various parameters which can be used for tuning batch performance as per the available hardware.

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

Batch Name	Batch Parameter	Recommended Value
C1-TXNRB	Thread Count	4 Threads Per CPU
	Chunk Size	5000 Transactions per 16 GB of RAM
	Maximum Batch Count	5000 Transactions per 16 GB of RAM
C1-TXNIP	Thread Count	4 Threads Per CPU
	Chunk Size	5000 Transactions per 16 GB of RAM
	Maximum Batch Count	5000 Transactions per 16 GB of RAM
C1-TXNVP	Thread Count	4 Threads Per CPU
	Chunk Size	5000 Transactions per 16 GB of RAM
C1-TXNEX	Thread Count	4 Threads Per CPU
	Chunk Size	5000 Transactions per 16 GB of RAM
	Maximum Batch Count	5000 Transactions per 16 GB of RAM
C1-TXNSQ	Thread Count	4 Threads Per CPU
	Chunk Size	5000 Transactions per 16 GB of RAM
	Maximum Batch Count	5000 Transactions per 16 GB of RAM
C1-TXNCM	Thread Count	4 Threads Per CPU
	Chunk Size	5000 Transactions per 16 GB of RAM
	Maximum Batch Count	5000 Transactions per 16 GB of RAM
C1-TXNCU	Thread Count	4 Threads Per CPU
	Chunk Size	5000 Transactions per 16 GB of RAM
	Maximum Batch Count	5000 Transactions per 16 GB of RAM
C1-DISTG	Thread Count	4 Threads Per CPU
C1-TXNDA	Thread Count	4 Threads Per CPU
C1-TXCNC	Thread Count	4 Threads Per CPU

Batch Name	Batch Parameter	Recommended Value
	Chunk Size	5000 Transactions per 16 GB of RAM
	Maximum Batch Count	5000 Transactions per 16 GB of RAM
	Thread Count	4 Threads Per CPU
	Chunk Size	5000 Transactions per 16 GB of RAM
C1-DELBL	Maximum Batch Count	5000 Transactions per 16 GB of RAM