Oracle® Revenue Management and Billing for Financial Services

Version 2.2.4.2

Transaction Feed Management - Batch Execution Guide

Revision 1.0

E37252-01

September, 2012

ORACLE

Oracle Revenue Management and Billing Transaction Feed Management - Batch Execution Guide

E37252-01

Preface

Purpose

This document details the steps for execution of batches and perquisites that must be carried out on environment before executing the batches and sequence for running the batches.

Intended Audience

Implementation team or any other team which plans to deploy ORMB for using the Transaction Feed Management system should understand this document.

Contents

1. Introduction

Oracle Revenue Management and Billing's Transaction Feed Management (TFM) system enables processing of transactions covering the details of product usage by the customers. This process involves importing the product charges and SQ calculation from the transaction data received into system involving the execution of multiple batch programs.

2. TFM Batch Running Sequence

Following Sequence must be followed when you run the TFM batch.

For aggregation, run the batches from 2.2 to 2.8 (F1-FLUSH, C1-TXNIP, C1-TXNVP, C1-TXNEX, C1-TXNSQ, C1-TXNCM, C1-TXNCU) in the sequence.

For roll backing the EROR or IGNR transactions, run the 2.1 (C1-TXNRB) batch. It will change the transaction status to "UPLD" status.

For disaggregation, run 2.10 to 2.11 (C1-DISTG, C1-TXNDA) batches together in sequence.

For cancellation of the transaction, run 2.12 (C1-TXCNC) batch. After running cancellation, run the all the batches for aggregation from 2.2 to 2.8 (F1-FLUSH, C1-TXNIP, C1-TXNVP, C1-TXNEX, C1-TXNSQ, C1-TXNCM, C1-TXNCU) in the sequence. This batch can be run at any point of time.

Do not run any batch twice in the sequence. If any batch is run twice, then please run the disaggregation batch for all the transactions. This should be followed by complete aggregation process 2.2 to 2.8 (F1-FLUSH, C1-TXNIP, C1-TXNVP, C1-TXNEX, C1-TXNSQ, C1-TXNCM, C1-TXNCU) in the sequence.

While running the batches, if any configuration is changed then run the disaggregation batch for all the transactions.

Batch Sequence table:

Batch sequence	Batch Control	Description
2.1	C1-TXNRB	Rollback Transaction Batch
2.2	F1-FLUSH	Flush the cache
2.3	C1-TXNHV	Header validation
2.4	C1-TXNIP	Initial Product Derivation Batch
2.5	C1-TXNVP	Price Assignment Batch
2.6	C1-TXNEX	Update Error Batch
2.7	C1-TXNSQ	SQ Calculation Batch
2.8	C1-TXNCM	Finalize Transaction Batch
2.9	C1-TXNCU	Transaction Clean up Batch
2.10	BILLING	Billing batch

2.1 Rollback Transaction Batch (C1-TXNRB)

This is an optional batch and is required to be run when you want to rollback EROR or IGNR transactions To UPLD.

Execution
Following is the set of inputs to be provided while executing this batch.

Note: Values shown are only for reference. The actual values would be based on the available hardware.

Input Property Name	Values	Required Y/N	Description
threadCount		Mandatory	Number of parallel threads to be spawned
chunkSize		Mandatory	Chunk size for creating work units
maxBatchCount		Mandatory	Batch size
status	EROR or IGNR	Mandatory	Possible values are: "EROR" OR "IGNR". If this parameter is entered as "EROR", then it rollbacks all transaction having status as "EROR". If this parameter is entered as "IGNR", then it rollbacks all transaction having status as "IGNR".

Post-Run Checks/Clean up

1. Successful completion of this batch would mark all transaction those are in EROR or IGNR status to UPLD status. It also deletes corresponding data in CI_TXN_DTL_PRITM table.

2.2 Flush Batch (F1-Flush)

This batch clears complete application cache. This is a single-threaded batch.

Pre-Checks:

Execution
Following is the set of inputs to be provided while executing this batch.

Input Property Name	Values	Required Y/N	Description

Post-Run Checks/Clean up

1. Successful completion of this batch would clear application cache completely.

2.3 Header validation Batch (C1-TXNHV)

This batch validates header of the transactions.

Execution
Following is the set of inputs to be provided while executing this batch.

Note: Values shown are only for reference. The actual values would be based on the available hardware.

Input Property Name	Values	Required Y/N	Description
threadCount		Mandatory	Number of parallel threads to be spawned
transactionHeaderId		Optional	Source header Id
transactionSource		Optional	Transaction source

2.4 Initial Product Derivation Batch (C1-TXNIP)

Execution
Following is the set of inputs to be provided while executing this batch.

Note: Values shown are only for reference. The actual values would be based on the available hardware.

Input Property Name	Values	Required Y/N	Description
threadCount	75	Mandatory	Number of parallel threads to be spawned
chunkSize	40000	Mandatory	Chunk size for creating work units
maxBatchCount	40000	Mandatory	Batch size
transactionHeaderId		Optional	Source header Id
transactionSource		Optional	Transaction source
division		Optional	Transaction division

Post-Run Checks/Clean up

1. Check the CI_TXN_DTL_PRITM table if transactions are processed and inserted into this table.
2. Check the status of transactions in CI_TXN_DETAIL table. The status of transactions should be INPD, EROR or IGNR.

2.3.1 DBMS Stats Run For CI_TXN_DTL_PRITM

Run DBMS stats for CI_TXN_DTL_PRITM as huge data is inserted into this table during batch run.

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

Note: Above is not a mandatory step. It can be skipped.

2.5 Price Assignment Batch (C1- TXNVP)

This batch derives price assignment for each of the Account Id – Product – TOU combination.

Execution
Following is the set of inputs to be provided while executing this batch.

Note: Values shown are only for reference. The actual values would be based on the available hardware.

Input Property Name	Values	Required Y/N	Description
threadCount	75	Mandatory	Number of parallel threads to be spawned
transactionHeaderId		Optional	Transaction header
transactionSource		Optional	Transaction source code
division		Optional	Transaction division

Post-Run Checks/Clean up

1. Successful execution of this batch should populate the CI_TXN_DTL_PRITM_SUMMARY table.

2.6 Update Error Batch (C1-TXNEX)

This is error handling batch which marks the transactions in error or ignore status if the pricing is not available or ignore switch is Y respectively.

Execution

Following is the set of inputs to be provided while executing this batch.

Note: Values shown are only for reference. The actual values would be based on the available hardware.

Input Property Name	Values	Required Y/N	Description
threadCount	75	Mandatory	Number of parallel threads to be spawned
chunkSize	40000	Mandatory	Chunk size for creating work units
maxBatchCount	40000	Mandatory	Batch Size
transactionHeaderId		Optional	Transaction header
transactionSource		Optional	Transaction source code
division		Optional	Transaction division

Post-Run Checks/Clean up

1. Successful execution of this batch should mark all the transactions in CI_TXN_DETAIL table in EROR or IGNR status for which pricing could not be derived or ignore switch flag is Y respectively. This can be checked by checking the CI_TXN_DTL_PRITM_SUMMARY table and corresponding transactions in the CI_TXN_DETAIL and CI_TXN_DTL_PRITM table.

2.7 SQ Calculation Batch (C1-TXNSQ)

This batch checks for SQIs configured and calculate the SQ for given frequency.

Execution

Following is the set of inputs to be provided while executing this batch.

Note: Values shown are only for reference. The actual values would be based on the available hardware.

Input Property Name	Values	Required Y/N	Description
threadCount	75	Mandatory	Number of parallel threads to be spawned
chunkSize	40000	Mandatory	Chunk size for creating work units
maxBatchCount	40000	Mandatory	Batch size
transactionHeaderId		Optional	Transaction header
transactionSource		Optional	Transaction source code
division		Optional	Transaction division
maxParallelDBJobs		Mandatory	Number of parallel threads to be spawned on database

Note: Once this batch run is complete, it's mandatory to run the next batch Finalize Transaction (C1-TXNCM). Even in case of failure of this batch, it is mandatory to run Finalize Transaction batch.

Post-Run Checks/Clean up

1. Successful completion of this batch would create the distinct billable charges and populate the CI_BILL_CHG and corresponding SQs in CI_BCHG_SQ table. It will update billable charge id in CI_TXN_DTL_PRITM table against corresponding transactions

2. Status of summary records in CI_TXN_DTL_PRITM_SUMMARY will be marked as "C" if DO_NOT_AGG_SW = "N".

3. Status of summary records in CI_TXN_DTL_PRITM_SUMMARY will remain in "P" if DO_NOT_AGG_SW = "Y". Billable Charge Ids will be populated in CI_TXN_DTL_PRITM table against corresponding transactions.

2.8 Finalize Transaction Batch (C1-TXNCM)

This batch marks the transaction either into COMP (i.e. COMPLETE) or EROR (i.e. ERROR) status, if the transaction has been processed successfully by all the batches or failed during SQI calculation batch respectively.

Execution

Following is the set of inputs to be provided while executing this batch.

Note: Values shown are only for reference. The actual values would be based on the available hardware.

Input Property Name	Values	Required Y/N	Description
threadCount	75	Mandatory	Number of parallel threads to be spawned

chunkSize	40000	Mandatory	Chunk size for creating work units
maxBatchCount	40000	Mandatory	Batch size
transactionHeaderId		Optional	Transaction header
transactionSource		Optional	Transaction source code
division		Optional	Transaction division

Post-Run Checks/Clean up

1. Successful completion of this batch would mark the status of transactions in the CI_TXN_DETAIL table as 'COMP' or 'EROR'

2.9 Transaction Clean up Batch (C1-TXNCU)

This batch is used to clean up or recalculate the billable charges if the transaction is marked in EROR while running the C1-TXNCM batch. This batch is mandatory to run.

Execution

Following is the set of inputs to be provided while executing this batch.

Note: Values shown are only for reference. The actual values would be based on the available hardware.

Input Property Name	Values	Required Y/N	Description
threadCount		Mandatory	Number of parallel threads to be spawned
chunkSize		Mandatory	Chunk size for creating work units
maxBatchCount		Mandatory	Batch size
maxParallelDBJobs		Mandatory	Number of parallel threads to be spawned on database

Post-Run Checks/Clean up

1. This batch will delete the billable charges if it is due to only error transactions. It will recalculate the SQI's for billable charge if the billable charge is created for multiple transactions (where some transactions are in EROR and some are in COMP). It will also reset the billableChargeId to blank in CI_TXN_DTL_PRITM table for error transaction.

2.10 Billing Batch (BILLING)

The bill cycle process creates bills for accounts with an "open" bill cycle.

Execution

Following is the set of inputs to be provided while executing this batch.

Note: Values shown are only for reference. The actual values would be based on the available hardware.

Input Property Name	Values	Required Y/N	Description
threadCount		Mandatory	Number of parallel threads to be

			spawned
MAX-ERRORS		Optional	Override maximum errors
CIS-DIVISION		Optional	Division Code
OFFCYC-BILLING		Optional	Y/N. Enter 'Y' if user want to create OFFCYC-BILLSs

Post-Run Checks/Clean up

1. The bill cycle process creates bill for accounts with an “open” bill cycle.

2.11 Disaggregation Entry Batch (C1-DISTG)

This batch inserts entries in CI_TXN_DISAGG_REQ table with BO_STATUS_CD as PENDING for disaggregation.

Pre-Checks:

Execution

Following is the set of inputs to be provided while executing this batch.

Input Property Name	Values	Required Y/N	Description
threadCount		Optional	Number of parallel threads to be spawned
division		Optional	Transaction division
billingFrequency		Optional	Billing cycle code

Post-Run Checks/Clean up

1. Successful completion of this batch would insert entries in CI_TXN_DISAGG_REQ table with BO_STATUS_CD as PENDING.

2.12 Disaggregation Batch (C1-TXNDA)

This batch de aggregates the transactions and marks the transaction into UPLD status.

Pre-Checks: Accounts which are to be disaggregated needs to be present in CI_TXN_DISAGG_REQ with BO_STATUS_CD as PENDING.

Execution

Following is the set of inputs to be provided while executing this batch.

Input Property Name	Values	Required Y/N	Description
threadCount		Optional	Number of parallel threads to be spawned
division		Optional	Transaction division
accountId		Optional	Account ID
frequency		Optional	Billing cycle code

Post-Run Checks/Clean up

1. Successful completion of this batch would mark of transactions in the CI_TXN_DETAIL table as ‘UPLD’ and should remove respective entries from Bills having status as PENDING, Billable charge, CI_TXN_DTL_PRITM.

2. If user wants to Disaggregate Account using frequency parameter, then he has to execute both C1-DISTG and C1-TXNDA Batches using Frequency parameter.

2.13 Cancellation Batch (C1-TXCNC)

This batch marks all transaction of specific header id into CNCL i.e. CANCELLED status.

Pre-Checks:

Execution
Following is the set of inputs to be provided while executing this batch

Input Property Name	Values	Required Y/N	Description
threadCount		Optional	Number of parallel threads to be spawned
headerId		Mandatory	Transaction Header ID

Post-Run Checks/Clean up

1. Successful completion of this batch would mark the status of transactions in the CI_TXN_DETAIL table as ‘CNCL and should remove respective entries from Bills having status as PENDING, Billable charge, CI_TXN_DTL_PRITM.

3. Recommended Parameter Values for each Batch

Note: The below are only the recommended values. The actual values to get maximum performance will vary with different hardware set. The recommendations are based on number of CPU and RAM available on DB and Application server. The actual performance would depend on the number of CPU, RAM on application server and many other hardware parameters. Product provides various input parameters which can be used for tuning as per available hardware. Request to follow below recommendations as guidelines only.

Sequence No.	Batch Control Name	Batch Parameters	Recommended Values
2.1	C1-TXNRB	Thread Count	Recommended value is 5 Threads Per Core.
		Chunk Size	Recommended value is 5000 per 16 GB of RAM
		Max Batch Count	Recommended value is 5000 per 16 GB of RAM
2.4	C1-TXNIP	Thread Count	Recommended value is 5 Threads Per Core.
		Chunk Size	Recommended value is 5000 per 16 GB of RAM
		Max Batch Count	Recommended value is 5000 per 16 GB of RAM
2.5	C1-TXNVP	Thread Count	Recommended value is equal to number of DIVISIONS of transactions
2.6	C1-TXNEX	Thread Count	Recommended value is 5 Threads Per Core.
		Chunk Size	Recommended value is 5000 per 16 GB of RAM
		Max Batch Count	Recommended value is 5000 per 16 GB of RAM
2.7	C1-TXNSQ	Thread Count	Recommended value is 2 Threads Per Core.
		Max Parallel Jobs on Database	Recommended value is 1 Threads Per CPU on database server.
		Chunk Size	Recommended value is 5000 per 16 GB of RAM
		Max Batch Count	Recommended value is 5000 per 16 GB of RAM
2.8	C1-TXNCM	Thread Count	Recommended value is 5 Threads Per Core.
		Chunk Size	Recommended value is 5000 per 16 GB of RAM
		Max Batch Count	Recommended value is 5000 per 16 GB of RAM
2.9	C1-TXNCU	Thread Count	Recommended value is 5 Threads Per Core.
		Chunk Size	Recommended value is 5000 per 16 GB of RAM
		Max Batch Count	Recommended value is 5000 per 16 GB of RAM
2.11	C1-DISTG	Thread Count	Recommended value is 2 Threads Per Core.
2.12	C1-TXNDA	Thread Count	Recommended value is 2 Threads Per Core.
2.13	C1-TXCNC	Thread Count	Recommended value is 2 Threads Per Core.

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com/ financial_services/

