

CGI Advantage[®] 4

Cost Accounting Run Sheets Guide



This document contains information proprietary to CGI Technologies and Solutions Inc. Unauthorized reproduction or disclosure of this information in whole or in part is prohibited.

CGI Advantage® is a registered trademark of CGI Technologies and Solutions Inc.

Due to the nature of this material, numerous hardware and software products are mentioned by name. In most, if not all, cases, the companies that manufacture the products claim these product names as trademarks. It is not our intention to claim these names or trademarks as our own.

Copyright © 2001, 2024, CGI Technologies and Solutions Inc. All Rights Reserved.

Table of Contents

1 Purpose of the System Administration Guide 4

 1.1 Common terms and glossary used 4

2 Description of Processes 5

 2.1 Cost Accounting Batch and Chain Processes 7

 2.1.1 Cost Allocation Base Load..... 8

 2.1.2 Cost Allocation Process..... 11

 2.1.3 Cost Allocation Multi Process 78

 2.1.4 Cost Allocation with Supplemental Reporting..... 85

 2.1.5 Cost Allocation Multi Process with Supp. Rpt..... 90

 2.1.6 Cost Allocation Standalone with Supp. Rpt. 95

 2.1.7 Credit Memo Multi Thread Chain..... 99

 2.1.8 Encumbrance Reclassification..... 112

 2.1.9 FHWA Cleanup Process..... 129

 2.1.10 FHWA FMIS Extract Process 134

 2.1.11 FMIS Load Process 157

 2.1.12 Internal Costing Chain 170

 2.1.13 Internal Overhead Process 202

 2.1.14 Labor Additive 206

 2.1.15 Labor Additive Reversal..... 220

 2.1.16 Labor Cost Distribution Load to Cost Allocation 222

 2.1.17 Maximo Other Cost Batch Job..... 254

 2.1.18 Overhead Rate Process 259

 2.1.19 Program Asset Generation 268

 2.1.20 Reclassification Process..... 279

 2.1.21 Reimbursable Expense Adjustment Process 296

 2.1.22 Reimbursement 301

 2.1.23 Reimbursement Generation Process 340

 2.1.24 Reimbursement Output Process..... 352

 2.1.25 Reimbursement Request Recycling Process 373

 2.1.26 Update Assistance Listing Numbers..... 379

 2.1.27 Warrant Reclassification 386

 2.2 Cost Accounting Report Processes 401

 2.2.1 CMIA Discrepancy Report 402

 2.2.2 CMIA Interest Calculation Process..... 404

 2.2.3 Negative Draw Prevention Process..... 411

1 Purpose of the System Administration Guide

This manual is intended to help system administrators initiate, configure, monitor, and control all processing for CGI Advantage. The manual has five parts:

1. The CGI Advantage System Administration Guide contains information about the CGI Advantage system architecture, and configuration (including the embedded third party components), post-installation setup, security configuration and considerations, workflow, job framework and its usage/maintenance, and other information pertinent to administering the application.
2. The CGI Advantage HRM run sheet guides describe each process of CGI Advantage HRM in detail with its input, output, parameters, sort sequence, and selection criteria.
3. The CGI Advantage Financial run sheet guides describe each process of CGI Advantage Financial in detail with its input, output, parameters, sort sequence, and selection criteria.
4. The CGI Advantage HRM Payroll Engine System Administration Guide describes the system control tables and utilities for CGI Advantage HRM.
5. The CGI Advantage VSS System Administration Guide describes each VSS process in detail with its input, output, parameters, sort sequence, and selection criteria.

System administration tasks include setting up and maintaining application security, querying and viewing the application status through logs and reports, managing workflow, setting up and maintaining system tables, and other critical application maintenance tasks.

1.1 Common terms and glossary used

The terms "Job" and "Batch" have been used interchangeably throughout the document. Please note that the CGI Advantage technical architecture is flexible enough to support the execution of jobs/batch processes while the application is available for online usage. In other words, the jobs/batch processes are technically not required to be "offline" processes.

2 Description of Processes

This chapter describes the processes in CGI Advantage that are considered system administration processes. For each process, you see information on these topics:

- Description
- Steps to Run this Process (if applicable)
- When to Run
- Major Input
- Output
- Parameters – Batch and Custom
- Sort Sequence
- Selection Criteria
- Notes
- Problem Resolution

System Wide Batch Parameters:

System wide batch parameter fields are available with each batch program, which provide the path for the input/output directory. These parameters allow sites to easily and quickly update the path for individual batch processes.

System wide batch parameters can be defined at the System Level, Area Level, Chain Job level, Chain Level or Job level. There has to be a default value set for the system wide batch parameters at any of these levels mentioned above so that the process will generate, read or write the respective files from the given location.

System wide batch parameters are defined at the System Level on the System Level Process Parameters (BATSETUP) reference page, searching for the Catalog Label of *Batch Catalog* and then choosing the record-level action of *Edit*.

- AMSROOT - Root directory of the batch files (for example, C:\AMSADV30\RTFiles)
- AMSEXPORT - For files that are created by the program and need to remain after the job is completed (i.e. cannot be temporary files). This could include interface files that come from/go to third party sources (for example, \$AMSROOT\ExportImport).
- AMSIMPORT - For files that are used by the program and need to remain after the job is completed (that is, cannot be temporary files). This could include interface files that come from/go to third party sources (for example, \$AMSROOT\ExportImport).
- AMSLOGS - For batch framework log files. If the job requires its own log files, this is where it is put (for example, \$AMSROOT\Logs).
- AMSPARM - Batch job parameter files specific to a single job instance only (for example, \$AMSROOT\Parms).
- AMSTEMP - For temporary files, usually stamped with process ID (for example, C:\TEMP).
- AMSSPOOL - Batch job report files, statistic files, exception reports, and so forth. These files may be sent to an OS print queue. File name is usually date and time stamped (for example, \$AMSROOT\Spool).

Note:

Assumptions while implementing system wide batch parameters: It is assumed that wherever in the Job processes system wide batch parameter variables (that is, AMSEXPORT, AMSIMPORT, AMSROOT, AMSLOGS, AMSPARM, AMSTEMP, AMSSPOOL) are declared as input parameters, care should be taken to set the overrideable flag for that variable to *true*, otherwise the process may fail.

Pivot Date/Year Validation:

Note:

Assumption for date attributes: Set the Earliest Year (EARLIEST_YEAR) and Latest Year (LATEST_YEAR) on the Application Parameter reference page. When defining the year range, attention should be given to setting a range vast enough to accommodate all system impacts (such as imported transactions). The Job input date/year must lie between the above year range; otherwise, the process will fail.

2.1 Cost Accounting Batch and Chain Processes

Cost Accounting is the tracking of accounting events that are associated with a special purpose. The purpose could be defined by funding received from an outside entity (for example, a Federal award or third-party award) or the purpose could be grouping costs together for internally defined reasons (for example, for a special spending initiative or specific costs incurred to provide a service). In many cases, Cost Accounting activity is tracked for purposes of billing an outside entity to reclaim some or all of the costs.

The processes that drive Cost Accounting in CGI Advantage Financial are listed below:

- [Cost Allocation Base Load](#)
- [Cost Allocation Process](#)
- [Cost Allocation Multi Process](#)
- [Cost Allocation with Supplemental Reporting](#)
- [Cost Allocation Multi Process with Supp. Rpt.](#)
- [Cost Allocation Standalone with Supp. Rpt.](#)
- [Credit Memo Multi Thread Chain](#)
- [Encumbrance Reclassification](#)
- [FHWA Clean Up Process](#)
- [FHWA FMIS Extract Process](#)
- [FMIS Load Process](#)
- [Internal Costing Chain](#)
- [Internal Overhead Process](#)
- [Labor Additive](#)
- [Labor Additive Reversal](#)
- [Labor Cost Distribution Load to Cost Allocation](#)
- [Maximo Other Cost Batch Job](#)
- [Overhead Rate Process](#)
- [Program Asset Generation](#)
- [Reclassification Process](#)
- [Reimbursable Expense Adjustment Process](#)
- [Reim Generation](#)
- [Reimbursement Output Process](#)
- [Reimbursement Request Recycling Process](#)
- [Reimbursement](#)
- [Update Assistance Listing Numbers](#)
- [Warrant Reclassification](#)

Descriptions of these processes are organized in this section in alphabetical order.

2.1.1 Cost Allocation Base Load

Job Name	Cost Allocation Base Load
Recommended Frequency	On demand before Cost Allocation process
Single Instance Required	No
Can be restarted?	No
Reports generated	No

Overview

When performing a Direct Financial Only or the Direct and Instream Financial allocation, there are instances where costs need to be allocated to all the accounting distributions where transactions have been charged. Manual setup of such an allocation must have base records based on unique combinations of Chart-of-Account (COA) distributions used. The Base Expansion Setup (BEXPNS) page and the Cost Allocation Base Load process combine to automate this work that is typically supported by one or more reports and uploads.

The process can be run in full or incremental modes. When run on incremental mode, the process only selects incremental records by considering the last processed records on Journal Log (JLOG) where the Process ID is CSTALOCBL. When run in a full mode, a Cost Allocation Parameter ID entered on Base Expansion Setup is used to determine the time frame for record selection.

The process performs the following steps with corresponding job log messaging:

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> Validating Batch Parameters Parameters are valid or invalid depending on the validation. If the parameter is invalid, the invalid value is displayed in the log. Batch Parameter validation completed
2. Record Selection	<ul style="list-style-type: none"> The process verifies that Selection and Accounting records are present on Base Expansion Setup for a step. If not, the following message is issued: "Selection and Accounting Base Expansion records are required for a step". For allocation where base type is Direct Financial and Instream if Allocation records are not present the following message is issued: "Accumulation Base Expansion records are required for Base Types of Direct Financial and Direct Financial Instream". If a pool record does not exist for a step, the process logs the following message: "At least one pool record must be entered for the step before a base record can be entered (A2270)". The process then identifies the journal records that need to be selected by using the COA combinations specified with Base Type of selection on Base Expansion Setup records. If the selection returns 0 records, the following message is issued: "No eligible records found" and the job run ends". For every record selected for processing, the following message is

Process Steps	Messages
	<p>written in the job log: “Begin processing of records from JRNL_ACTG table”.</p> <ul style="list-style-type: none"> • When run in Incremental mode the records are selected incrementally based on last JRNL_LOG entry. • Once selected records are processed, the following message is written in the job log to show count on number of records were added: “<n> Rows were saved to PLBS table and <n> rows were added to the PBDIST table.”
3. Update Cost Allocation Setup	<ul style="list-style-type: none"> • The process then summarizes the records using COA elements. • Each summarized record is written to Pool Base Setup (PLBS) and Distribution (PBDIST) page for the Allocation, Series and Step ID specified. • The process inserts a record to Journal Log (JLOG) with start and end record numbers processed with a Process ID of CSTALOCBL.

Major Input

- Base Expansion Setup (BEXPNS / R_BASE_EXPN_SETUP)
- Accounting Journal or Cost Accounting Journal (JACTG / JRNL_ACTG or JCA / JRNL_CA)
- Journal Log (JLOG / JRNL_LOG)

Batch Parameters

The following are the delivered parameter values which may have been updated through Batch Setup to meet local needs:

Parameter	Description	Default Value
Run Mode (MODE)	The required indication if the process is running as 1- <i>Full Mode</i> using the CAPA ID from BEXPNS or 2- <i>Incremental Mode</i> using JLOG.	N/A
Source Journal (SELECTION_TBL_NM)	The required journal for record selection. Only the following data objects are allowed: <i>JNRL_ACTG</i> (Accounting Journal) or <i>JRNL_CA</i> (Cost Accounting Journal).	JRNL_ACTG
Progression Counter Size (PROG_CTR_SZ)	A required count for the process to issue progression messages based on the number of journal records processed. If not entered, it defaults to 1000.	1000
Select Block (SELECT_BLOCK)	A required number of records selected from the journal in an instance. If not entered, it is defaulted to 1000.	1000

Major Output

- Pool/Base Setup (PLBS / R_PLBS_SETUP)

- Pool/Base Distribution (PBDIST / R_PLBS_DIST)
- Journal log (JLOG / JRNL_LOG)
- JRNL_ACTG_CA_TEMP

Sort Criteria

The summarized unique accounting distribution records are written to pool Base set up and distribution Tables.

Selection Criteria

The process selects records from source journal using the COA combination specified on Base Expansion Setup (BEXPNS) records with a Base Type of Selection.

Problem Resolution

If the job fails due to application errors, it is advisable to restart the job after correcting the errors instead of rescheduling the job. Alternatively, a new instance can be submitted that starts with the job that previously failed.

The following table shows the potential job return codes for this job:

Return Code	Condition
Successful (1)	This Return Code is issued when parameter validation is successful and records are written to the table.
Warning (4)	This Return Code is issued when no records found for processing.
Non-Fatal Error (8)	This job does not issue this Return Code.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for unexpected situations.
Terminated (16)	This Return Code is issued when the job is terminated by the user.
System Failure (20)	This Return Code is issued when the job is terminated because of database server or network issues.

2.1.2 Cost Allocation Process

Chain Job Name	Cost Allocation Process
Recommended Frequency	The Cost Allocation Process can be run daily, monthly, quarterly or on an annual basis.
Single Instance Required	Yes. Inside of a single chain, a single instance is allowed for each of the four job steps. However, multiple chain jobs may be run concurrently.
Can be restarted?	Yes, see the individual jobs for more details.
Reports generated	Yes, some of the jobs in the chain generate the exception report. Please refer to the individual jobs for more details.

Overview

The Cost Allocation Process in CGI Advantage Financial is a group of jobs that work together in five different run modes:

1. Offline Validation
2. Base Accumulation
3. Compute Allocation
4. Generate Transaction
5. Reversal

The purpose of the Cost Allocation Process is to allocate selected accounting activity to the appropriate entities. Activity for allocation is initially recorded to 'pool' entities. The Cost Allocation Process then allocates this activity to 'base' entities. Allocated activity may be eligible for reimbursement by various funding agencies. In addition to the allocation of expenditures (indirect costs) on a dollar-for-dollar basis, indirect revenues and charges may also be allocated. The three options are available on the Allocation record on the Cost Allocation Process Parameters (ALOC) page. The fourth option, a charge back, is also available. The Cost Allocation User Guide should be consulted on the details of these different types of allocations.

The Cost Allocation Process is one of two methods that can be used to record indirect costs, the other process being the Overhead Rate Process. The Overhead Rate Process does not perform a dollar-for-dollar allocation of indirect costs. Instead, the Overhead Rate Process acts as an "additive" process whereby an agreed upon additive or overhead rate is applied to a pool of indirect costs. The Cost Allocation Charge Back process functions in a similar manner, with some additional flexibility in the accounting output that is generated. The Charge Back process will calculate the expenditure or revenue additive amount, and that additive will then be posted to the system.

The Cost Allocation Process has the following jobs (each of the jobs listed below, is described in subsequent sections):

1. [Cost Allocation Process](#)
2. [Load CA](#)
3. [Submit CA](#)
4. [CA Transaction Exception Report](#)

There are two basic run cycles of the Cost Allocation Process that are controlled by the Run Cycle field on the Cost Allocation Process Parameters (CAPA) page. The first is referred to as

the multiple run method where one instance of the process is run in *Offline Validation* mode. A user then changes the Run Mode on CAPA to *Base Accumulation*. This is then repeated for *Compute Allocation* and *Generate Transaction* modes. If needed the process is then run a fifth time in *Reversal* mode. The other run cycle is the single run where a user only needs to execute the process in *Offline Validation* mode. If the Exception Report job step finishes without any reported issues, the process will update CAPA, and submit another instance of the process. This repeats through the *Generate Transactions* mode.

The *Cost Allocation* Process is commonly executed with a single instance whether executing in either Run Mode of CAPA. However, it is possible to run multiple instances of the process (not the individual job steps inside of the chain) as desired for different Departments at different times or in parallel. In order to enable multiple instances, the Cost Allocation Process ID (that is, CAPA ID) is used to identify records for distinct instances of the Chain Job. When multiple instances of the Cost Allocation Process are executed simultaneously, please ensure:

- A. That the CAPA IDs are distinct amongst the multiple runs.
- B. That the setup inside of the associated Allocation and Pool Base Setup will not use overlapping records from the input ledger. If the same ledger record is selected and processed by multiple runs, then that amount will be allocated more than once, which is unlikely to be the intention.
- C. Any previous assumptions or requirements needed for the single run are also met.

The acceptable job return code configuration depends on the business requirement. For example, if the requirement is that the subsequent jobs in the chain should continue only if the job ends with a Return Code of *Successful*, the Pre Condition Return Codes for each job step should be set to *Successful* on the Setup for Job Steps page for the Cost Allocation Process on Batch Setup. If it is acceptable for a job step to end as *Non-Fatal* and have the subsequent job step run, that can also be configured on Job Setup. For baseline configuration, all of the jobs in the chain will be kicked off only when the previous job ends with a Return Code of *Successful*. If any of the jobs in the chain ends with a Return Code of *Failed*, *Terminated* or *System Failure*, all of the subsequent jobs will be set to Inactive.

Run Notes:

- If a Cost Allocation Process Parameter (CAPA) record has the Run Cycle set to *Multiple*, even though jobs in the chain can be run individually by disabling other jobs, it is recommended to run the entire chain in *Generate Transaction* and *Reversal* modes.
- When a Cost Allocation Process Parameter (CAPA) record has the Run Cycle set to *Multiple*, for the first three modes (*Offline Validation*, *Base Accumulation* and *Compute Allocation*) it is recommended to disable Load CA, Submit CA, and CA Transaction Exception Report jobs as these Run Modes do not produce any data for those later job steps. If your Cost Allocation Process Parameter (CAPA) record has the Run Cycle set to *Single*, then you must always run all steps inside of the Chain.
- If the Cost Allocation Process is run in Single mode and the Offline Validation job ends with a Return Code of *Failed*, the subsequent instances (Base Accumulation, Compute Allocation and Generate Transaction) will not be spawned.

Major Input

Offline Validation Mode

- Cost Allocation Process Parameters (CSAL_PROC_PARM)
- Cost Allocation Step Control (R_CSAL_STPCTR_SETP)
- Pool/Base Setup (R_PLBS_SETP)
- Pool/Base Distribution (R_PLBS_DIST)

- COA Element Reference tables (R_FUND, R_DEPT, R_LOC etc...)

Base Accumulation Mode

- Cost Allocation Process Parameters (CSAL_PROC_PARM)
- Cost Allocation Journal Summary (CSAL_JRNL_SUM)
- Cost allocation Control Setup (R_CSAL_CTRL_SETUP)
- Cost Allocation Step Control (R_CSAL_STPCTR_SETUP)
- Posting Code (R_PSCD)
- Object Rate Group (OBJ_RT)
- Pool/Base Distribution (R_PLBS_DIST)
- Pool/Base Setup (R_PLBS_SETUP)
- Unit Rollup (R_CA_UNIT_RLUP)
- Department Object Rollup (R_CA_DOBJ_RLUP)
- Department Object Rollup Exception (R_CA_DOBJ_RLUP_EX)

Compute Allocation Mode

- Cost Allocation Process Parameters (CSAL_PROC_PARM)
- Pool/Base Setup (R_PLBS_SETUP)
- Pool/Base Distribution (R_PLBS_DIST)
- Cost Allocation Journal Summary (CSAL_JRNL_SUM)
- Cost Allocation Control Setup (R_CSAL_CTR_SETUP)
- Cost Allocation Series Control Setup (R_CSAL_SRCTR_SETUP)
- Cost Allocation Step Control Setup (R_CSAL_STPCTR_SETUP)
- Cost Allocation Expansion Results (CSAL_EXPN_RSLT)
- Unit Rollup (R_CA_UNIT_RLUP)
- Department Object Rollup (R_CA_DOBJ_RLUP)
- Department Object Rollup Exception (R_CA_DOBJ_RLUP_EX)

Generate Transaction Mode

- Cost Allocation History (CSAL_HIST)
- Cost Allocation Process Parameters (CSAL_PROC_PARM)
- Program Transaction Cross Reference (R_PROG_DOC_CRSREF)
- Submit Parameter File(.txt)

Reversal Mode

- Cost Allocation History Table (CSAL_HIST)
- Submit Parameter File (.txt)

Major Output

Offline Validation Mode

- Pool/Base Verification Report
- COA Editing Report
- Exception Report

Base Accumulation Mode

- Cost Allocation Journal Summary (CSAL_JRNL_SUM)
- Cost Allocation Totals (CSAL_TOT)
- Base Accumulation Report
- Base Accumulation Exception Report

Compute Allocation Mode

- Cost Allocation Journal Summary (CSAL_JRNL_SUM)
- Cost Allocation Expansion Results (CSAL_EXPN_RSLT)
- Cost Allocation History (CSAL_HIST)
- Pool View Report
- Base View Report
- Summarization Report
- Expansion Exception Report

Generate Transaction Mode

- CA XML Transaction
- Cost Allocation History (CSAL_HIST)
- CA Exception Report
- Longer Term Retention tables:
 - CSAL_EXPN_RSLT_STRG
 - CSAL_HIST_STRG
 - R_CSAL_CTRL_SETP_STRG
 - R_CSAL_SRCTR_SETP_STRG
 - R_CSAL_STPCTR_SETP_STRG
 - R_PLBS_DIST_STRG
 - R_PLBS_SETP_STRG

Reversal Mode

- CA XML Transaction
- Cost Allocation History (CSAL_HIST)
- CA Exception Report
- Longer Term Retention tables:
 - CSAL_HIST_STRG

Chain Job Return code

The following table shows the potential return codes for the Cost Allocation job. Note that the Chain job ends with the highest return code across all of the jobs.

Return Code	Condition
Successful (1)	All of the jobs end successfully
Warning (4)	One of the jobs in the chain ends with a return code of "Warning"
Non-Fatal Error (8)	One of the jobs in the chain ends with a return code of "Non-Fatal Error"
Failed (12)	One of the jobs in the chain ends with a return code of "Failed"
Terminated (16)	One of the jobs in the chain ends with a return code of "Terminated"
System Failure (20)	One of the jobs in the chain ends with a return code of "System Failure"

Problem Resolution

If any of the jobs in the chain failed due to application errors it is advisable to reschedule the job after correcting the errors. If the job failed in Reversal Mode due to application errors then it is advisable to restart the job after correcting the errors instead of rescheduling the job. Restarting the job reduces the processing time since the job will resume from where it has last committed and select only the unprocessed records. If the job fails due to any exceptions on the Exceptions report, such exceptions should be addressed individually.

Please refer to the individual jobs for details regarding the specific job processes and problem resolution.

Cost Allocation Process Chain: Cost Allocation Process Job

Job Name	Cost Allocation Process Job
Recommended Frequency	The Cost Allocation Process can be run daily, monthly, quarterly or on an annual basis.
Single Instance Required	No. Inside of a chain it is a Single Instance; however, there can be many chains running simultaneously.
Can be restarted?	Yes, when job is run in Compute Allocation, Generate Transaction, and Reversal Modes.
Reports Generated	Yes.

Overview

The purpose of the Cost Allocation Process is to allocate indirect costs to the appropriate entities. Indirect costs are initially recorded to 'pool' entities. The Cost Allocation Process then allocates those costs on a dollar-by-dollar basis to 'base' entities. Those indirect costs may subsequently be eligible for reimbursement by various funding agencies. The Cost Allocation Process is one of two methods that can be used to record indirect costs. The Cost Allocation process will also perform Charge Back processing, where additional expenditures and revenues will be posted to the system based on selected expenditures.

This process involves considerable setup. There are five pages that are set up in the following order:

1. Cost Allocation Control Setup (ALOC): Global features for the allocation are defined such as the frequency of the allocation, whether the allocation will involve the allocation of revenues or expenditures, and the pool/base inheritance for all chart of account elements involved in the allocation.
2. Cost Allocation Series Setup (SRS): Multiple series can be set up within an allocation. Users can override the inheritance values set at the allocation level.
3. Cost Allocation Step Setup (STEP): Multiple steps can be set up within a series. A base record may be a pass-through entity. In this case, costs initially allocated to the base in one step are subsequently re-allocated in a 'forward step'.
4. Pool/Base Setup (PLBS): Pools and bases are defined for each step. Depending on the base type, various features will need to be recorded for each pool/base record. Negative entries are allowed for the Allocation Percentages.
5. Pool/Base Distribution (PBDIST): The 'accumulation' and 'accounting' distributions associated with each pool and base record are set up on this table.

The Cost Allocation Process will run in the following 5 different run modes. There is a logical sequence for running the Cost Allocation Process. Each Run Mode corresponds to the sequence of events that takes place in Cost Allocation. Table updates and/or transaction processing occurs in four (Generate Transaction) of the five run modes, Offline Validation being the exception.

1. Offline Validation Mode

No table updates take place when the process is run in this Run Mode. Because the chart of account elements cannot be validated at the time of setup (no fiscal year is defined), there may be some invalid data on Pool/Base Distribution. There may also be some other anomalies in the data setup. The process is capable of validating COA setup as a fiscal year is required as a run-time parameter field. If any such exceptions are found, then the exceptions will be updated on the Exception Report.

2. Base Accumulation Mode

The purpose of Base Accumulation is to build the Cost Allocation Journal Summary (CAJR) records (CSAL_JRNL_SUM). The Cost Allocation parameter record defines how records are selected. The input source is the Input Ledger for which records are selected based on the following field values on the Cost Allocation Parameter record:

- Fiscal Year
- Fiscal Month
- Fiscal Quarter
- Daily Run Date
- Selection From Date and Selection To Date

(Care should be taken not to skip a time period, and even more care should be taken not to run an allocation for the same time period or an overlapping time period without first reversing the prior run.)

CAPA has a Date Attribute Used field that indicates whether Record Date or Posted Date will be used for comparison with parameter date values given in case of Daily (Daily Run Date) or Date Range (Selection From Date and Selection To Date) frequencies. Record Date (DOC_REC_DT) is the date found on all transaction Headers that defaults to the Application Date from Application Parameters (APPCTRL) unless otherwise entered. Posted Date (PSTD_DT) is the date on which an Accounting Journal or Cost Accounting Journal record is created. When either of these dates is used for record selection, it must be present and populated on the input ledger based on Journal Ledger Control (JLCTRL) settings.

Based on whether the allocation is for revenues, charges, or expenditures, the process knows which types of accounting activity to select from the input ledger. Posting Code

(PSCD) records define whether or not a ledger record is eligible by matching the Cost Allocation Process value on PSCD to the selected type of accounting from Cost Allocation Control Setup (ALOC). In the event the Refined Posting Code Listing field on CAPA is used, the process will be limited to just that set of posting codes and will not use the entire set of matching posting codes.

The Perform COA Summarization flag on the Cost Allocation Process Parameters (CAPA) page indicates if the Cost Allocation Journal Summary (CSAL_JRNL_SUM) table will be summarized or not during this step. If the flag is selected, the Base Accumulation run mode will perform the extra step of re-summarizing the records based on the rollup tables.

This run mode also calculates a base percentage value for all base records whose base type is not a fixed percentage. All statistical, direct financial and direct and instream financial base records are not initially set up with an allocation percentage. That allocation percentage is computed during base accumulation:

- **Statistical:** The statistical units for each base record in an allocation step are added together. The allocation percentage is computed by determining each base's statistical units as a percentage of the total statistical units for all steps in a base.
- **Direct Financial/Direct and Instream Financial:** Base Accumulation Distributions are used to compute a total dollar amount for each base record. The dollar amounts are treated as statistical units at this stage. The allocation percentage is then computed in the same way as described above in the Statistical Method.

Note: It is recommended to disable Load CA, Submit CA and CA Transaction Exception Report jobs as this is a Report Only mode only when the Cost Allocation Process ID has the Run Cycle set to *Multiple*.

3. Compute Allocations Mode

The Rebuild Summary in Compute Allocation flag on the Cost Allocation Process Parameters (CAPA) page indicates if the Cost Allocation Journal Summary (CSAL_JRNL_SUM) table will be rebuilt or not during this step. If the flag is selected and Perform COA Summarization flag is also selected while rebuilding the Cost Allocation Journal Summary (CSAL_JRNL_SUM) table, the Compute Allocation run mode will perform the extra step of re-summarizing the records based on the rollup tables. If this flag is not selected, the CSAL_JRNL_SUM table will not be rebuilt during this step. Any necessary COA modifications can be made against this table between Base Accumulation and Compute Allocations without being lost.

- **Pool Expansion:** Pool accounting distributions are expanded against records on the Cost Allocation Journal Summary (CSAL_JRNL_SUM) table. Each pool record is compared to each record on the Cost Allocation Journal Summary (CSAL_JRNL_SUM) table to determine if a match exists. Wildcard functionality has been built into the system for pool expansion. Another new feature in CGI Advantage 3 pool expansion is the ability to expand against chart of account rollup elements. Once a match is found, the pool record is moved to the Cost Allocation Expansion Results (CSAL_EXPN_RSLT) table.
- **Base Expansion:** Base accounting distributions are expanded against previously expanded pool records on Cost Allocation Expansion Results (CSAL_EXPN_RSLT) table. The output for each step is that all base records expand against each pool record within that step. Inheritance determines whether the expanded record will assume the chart of account value from the expanded pool record or from the base accounting distribution.
- **Calculation of the Instream Amount:** This is new to CGI Advantage 3. Base accumulation distributions are compared to base accounting distributions from previous series/steps in the same allocation. If exact matches are found, then the allocated amount(s) from those previous series/steps are added to the base statistical unit (dollar

amount) in the current step. The updated statistical unit is used to determine an updated allocation percentage for each base record in the step (see Base Accumulation above for further explanation).

- Allocation to Base Records: Once the final allocation percentages have been determined, the process then applies the appropriate percentage to all base records on Cost Allocation Expansion Results.
- Each Cost Allocation Journal Summary record that has been allocated is updated. The dollar amount field is moved to \$0.00 to indicate that this record has been allocated.
- Cost Allocation Expansion Results records are created on Cost Allocation History that will be used to create transactions when the chain job is run in the mode below.
- The corresponding ledger record numbers for the selected pool records are concatenated and set on the summarized Cost Allocation History record.
- Note: It is recommended to disable Load CA, Submit CA, and CA Transaction Exception Report jobs as this is a Report Only mode only when the Cost Allocation Process ID has the Run Cycle set to *Multiple*.

4. Generate Transaction

This step is optional. If users wish to run Cost Allocation for report only purposes, then it is not necessary to run the process in this run mode. This step generates Cost Allocation transactions in XML format. The process selects all records that have been allocated in the previous run mode (Compute Allocations) and creates accounting lines. Transaction breaks are generated in the following situations:

- CH_DOC_ACTG line limit on the Transaction Component Requirements table is reached
- Change in Chart of Accounts elements (optional depending on Cost Allocation Process Parameter ID used)
- Change in Step Number
- Change in Series ID

Use of the line limit feature on the Transaction Component Requirements table is strongly encouraged since an allocation can result in many accounting lines being generated for a single step number and/or appropriation.

When running a Direct Cost Allocation, an accounting line is created for each pool record selected that reverses the initial recording of cost/revenue to that pool. For each base record selected, an accounting line is generated that allocates the correct amount from the pool to the base as defined or determined. When the Transaction Component Requirements limit is used, cost allocation transactions will not sum to zero dollars at the header level unless all lines for a Step Number and/or Appropriation can fit on a single transaction. However, a sum on the header amounts of all transactions created will be zero dollars.

When running a Charge Back, an accounting line is generated for each base record selected, allocating the correct charge to that base as defined or determined. Transactions from this type of allocation do not sum to zero dollars like the other type of allocation, so an accounting line limit will not have any effect on the header amounts, except they will be smaller if multiple transactions are generated.

If the Build Reference option is selected on the Parameter table, the Reference transaction information will also be populated on the newly generated transactions.

If the Pool/Base Apportion option is set to *Pool* on the Parameter table, then the Pool line will be apportioned across all the generated Cost Allocation (CA) transactions. For an allocation setup that has multiple pool lines and a single base line, Base Apportion will split the base lines across multiple CA transactions. Pool/Base Apportion will work only when Break by COA is specified.

The generated XML file is then loaded to the Transaction Catalog as part of the next step in the chain. A parameter file and a submit parameter file are passed from the first job in the chain to the second, which is a SysManUtil job. The third job in the chain will submit the transactions with another SysManUtil job. The final job in the chain is an exception report stating those transactions that failed to submit.

- Note 1: Default event types have been set up for the allocation of revenues/ expenditures/charges. Users have the option of overriding the default event types. If users wish to represent the results of Cost Allocation in a non-accounting transaction, then the appropriate event type would be a Charge event type that would only update the Cost Accounting Journals/Ledgers.
- Note 2: Even though the above jobs in the chain can be run individually by disabling other jobs, it is recommended to run the entire chain in mode 4 (Generate Transaction).
- Note 3: On the Header of the CA transaction, Allocation ID, Series, Step, Cost Allocation Parameter ID, and Selection APD, Selection Fiscal Year, Selection From Date, Selection To Date, and Data Source from the Cost Allocation Parameter page (CAPA) are concatenated and set on the Transaction Description field.
- On the Pool Accounting Line description, the comma delimited Ledger Record Numbers corresponding to the respective selected ledger records are set.

5. Reversal

This step should only be executed if the transactions generated in the previous step are deemed to be invalid. This step is accomplished in a separate running of the process in 'Reversal' mode. Transactions that were successfully submitted will be reversed. Cost Allocation History records that resulted in the initial generation of Cost Allocation transactions are deleted from the History table.

Note: Even though the above jobs in the chain can be run individually by disabling other jobs, it is recommended to run the entire chain in mode 5 (Reversal).

Offline Validation

The following table shows the various steps that the Cost Allocation Process Job goes through and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> • Validating Batch Parameters. • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. It will be followed by the message "Batch Parameter validation failed." • Batch Parameters validation completed.
2. Report Generation	<ul style="list-style-type: none"> • Generation of COA Editing Report started • Generation of COA Editing Report completed • Generation of Validation Exception Report started • Generation of Validation Exception Report completed • Generation of PBDF Verification Report started • Generation of PBDF Verification Report completed

Restartability Information

Job cannot be restarted in this run mode. If the job fails in any of above steps, a new job should be scheduled after correcting the errors that caused the job to fail.

Major Input

Tables

- Cost Allocation Process Parameters (CSAL_PROC_PARM) – Offline Validation Date is used to determine the fiscal year for COA validation.
- Cost Allocation Step Control (R_CSAL_STPCTR_SETP) - Base Type (Fixed Percentage) for the step is available from this table.
- Pool/Base Setup (R_PLBS_SETP) - Pool and Base Records are available from this table.
- Pool/Base Distribution (R_PLBS_DIST) - COA Elements defining the pool and base distribution are available from this table.
- COA Element Reference tables (R_FUND, R_DEPT, R_LOC etc...) - COA element is validated against these tables.

Batch Parameters

Parameter	Description	Default Value
(Optional) Include Additional reporting fields. Valid values are Yes or No (ADDTL_RPT_FLDS)	Optional. Include Additional reporting fields. Valid values are Yes or No.	No Default
Export Location at Cost Allocation Process Job (AMSEXPORT)	Required. (** Refer to Note: Assumptions for SWBP on page no. 5)	No Default
Logs Location at Cost Allocation Process Job (AMSLOGS)	Required. (** Refer to Note: Assumptions for SWBP on page no. 5)	No Default
Importing Location at Cost Allocation Process Job (AMSPARM)	Required. (** Refer to Note: Assumptions for SWBP on page no. 5)	No Default
Checkpoint Block Size (Number of transactions processed after which checkpoint is to be updated) (CHK_PT_SIZE)	Optional. This parameter is used to show the number of transactions processed after which a checkpoint is to be updated. The Value of this parameter should be a positive integer. If not entered, it is defaulted to 100.	100

Parameter	Description	Default Value
Client name for Report (CLIENT_NM)	Optional. Entry of a value in this field specifies the name that will appear on reports.	No Default
Parameter ID	Required. Cost Allocation Parameter ID of the record on the Cost Allocation Process Parameter (CSAL_PROC_PARAM) Table that will be used in this run of Cost Allocation.	No Default
Exclude Report (EXCL_COST_ALLOC_REP)	Optional. Enter numeric values separated by a comma to exclude the generation of the report by the Cost Allocation process. 1. Pool View 2. Base View 3. Expansion Exception 4. CA Transaction Exception	No Default
Commit Block Size (if not entered then defaulted to 1000) (COMMIT_BLOCK)	Optional. The value of this parameter should be a positive integer. If not entered, it is defaulted to 1000.	No Default
Progression Message Block Size (PROG_CTR_SZ)	Optional. The value of this parameter should be a positive integer. If not entered, it is defaulted to 1000. It is used to show the progress of record processing.	1000
Select Block Size (SELECT_BLOCK)	Optional. It is the number of records fetched at a time. The value of this parameter should be a positive integer. If not entered, it is defaulted to 1000. Can be used for Performance tuning.	1000
Submit Parameter File (.txt) (SUBMIT_FILE)	Required. The output file should be named CostAllocSubmit_CAPA<< Parameter ID >>. This is to ensure that different simultaneous runs will not be in conflict. .txt is required for the extension of this file name. For example, if the CAPA ID is 100, then the parameter should be set to CostAllocSubmit_CAPA100.txt. The Default Value should <u>not</u> be used, but the value can be quickly updated for each run.	CostAllocSubmit.txt

Parameter	Description	Default Value
Apply Overrides (APPLY_OVERRIDES)	Required. If overrides should be applied to transactions, this parameter should be <i>true</i> . A setting of <i>false</i> will result in transactions rejecting for override errors.	true
Override Level (OVERRIDE_LVL)	Optional. Override level to be used if the Apply Overrides parameter is <i>true</i> . If 0 or left blank and the Apply Overrides parameter is <i>true</i> , then the level of the user executing the chain job will be used.	No Default
Purge Previous Run CAPA ID (PURGE_PREV_RUN_CAPA_ID)	Optional. Previous run CAPA ID whose data is to be purged from the following processing tables: CSAL_JRNL_SUM, CSAL_TOT, CSAL_EXP_N_RSLT, CSAL_HIST. This parameter does not apply to the CAPA ID associated with the current run.	

Custom Parameters

Data Object Name: CSAL_PROC_PARM

Parameter	Description	Default Value
Cost Allocation Parameter ID (CSAL_PROC_PARM_ID)	Required. Unique identifier of a set of online parameters for Cost Allocation.	No Default
Run Mode (RUN_MOD)	Required. Select the <i>Offline Validation</i> mode.	No Default
Allocation ID (ALOC_ID)	Required. The Allocation ID from the Cost Allocation Control (ALOC) Setup record used for this run of Cost Allocation.	No Default
Allocation Frequency (ALOC_FREQ)	Required	Value inferred from Cost Allocation Control Setup Table (R_CSAL_CTR_SETUP) for the entered Allocation Id.
Selection APD (PER)	Conditionally required. If Allocation Frequency is <i>Monthly</i> , then Selection APD is required for selection from the Data Source.	No Default
Fiscal Quarter (FQTR)	Conditionally required. If Allocation Frequency is <i>Quarterly</i> , then Fiscal Quarter is required for selection from	No Default

Parameter	Description	Default Value
	the Data Source.	
Fiscal Year (FY)	Conditionally required. If Allocation Frequency is <i>Monthly</i> , <i>Quarterly</i> , or <i>Annual</i> , then Fiscal Year is required for selection from the Data Source.	No Default
BFY Option (BFY_OPT_FL)	Optional. When set to <i>True</i> , the selection of ledger records will use the BFY field on the ledger instead of the Fiscal Year. The BFY to be selected is then entered in the Fiscal Year parameter field.	No Default
Daily Run Date (DLY_RUN_DT)	If Allocation Frequency is <i>Daily</i> , then Daily Run Date is required for selection from the Data Source.	No Default
Selection From Date (SEL_FRM_DT)	Optional. If Allocation Frequency is Date Range, then Selection From Date is required for selection from the Data Source.	No Default
Selection To Date (SEL_TO_DT)	Optional. If Allocation Frequency is Date Range, then Selection To Date is required for Selection from the Data Source.	No Default
Data Source (DATA_SRC)	Conditionally required. For all modes other than <i>Offline Validation</i> , a ledger must be identified. It is important that the ledger contains all COA necessary and the date information necessary to match selection settings.	No Default
Build References (BLD_REF)	Optional. Indicates to the process that the allocation in <i>Generate Transaction</i> mode will look to Program Transaction Cross Reference (PRGXREF) for any encumbrance to reference. The Build References option may only be selected when the Cash Expenditures or Charge Back option is selected on Cost Allocation Control Setup (R_CSAL_CTR_SETUP).	No Default
Break by COA (BRK_COA)	Optional. When Break by COA is selected, the records will be grouped by the COA element(s) and a new Cost Allocation (CA) transaction will be created when the combination changes. Valid values are data objects in the COA tables, APPR_CD. Multiple values, comma-	No Default

Parameter	Description	Default Value
	separated, can be specified, FUND_CD, DEPT_CD, APPR_CD.	
Offline Validation Date (OFFLN_VLD_DT)	Conditionally required. Required to validate chart of account elements on Pool/Base Distribution in <i>Offline Validation</i> mode.	No Default
Pool/Base Apportion (PLBS_APPRTN)	Optional. Valid values are <i>Pool</i> and <i>Base</i> . Used with Break by COA parameter. When selected, the Pool/Base amounts get apportioned across the Cost Allocation (CA) transactions.	<Select>
Transaction Type (DOC_TYP)	Required	CH
Transaction Code (DOC_CD)	Conditionally required. When running in <i>Generate Transaction</i> mode, this is a required parameter defining the transaction code that will be created.	No Default
Prefix (PFX)	Optional. When running in <i>Generate Transaction</i> mode, this is an optional parameter defining a prefix used when creating transaction IDs.	No Default
Transaction Department Code (DOC_DEPT_CD)	Optional. When running in <i>Generate Transaction</i> mode, this is a required parameter defining the transaction department used when creating transaction IDs.	No Default
Transaction Unit Code (DOC_UNIT_CD)	Optional. When running in <i>Generate Transaction</i> mode, this is an optional parameter defining a unit used when creating transaction IDs.	No Default
Transaction Record Date (DOC_REC_DT)	Optional. When supplied the date is written as the Transaction Record Date on the Allocation transactions instead of having the Application Date default.	No Default
Expenditure Event Type (EXP_EVNT_TYP_ID)		CA01
Revenue Event Type (EXP_EVNT_TYP_ID)		CA02
Revenue Credit Event Type (REV_CR_EVNT_TYP_ID)		CG03
Charges Event Type (CH_EVNT_TYP_ID)		CG01
Inverse Event Type		CA03

Parameter	Description	Default Value
(INV_EVNT_TYP_ID)		
Retain FY (RET_FY)	Conditionally required. Indicates that the Fiscal year of the selected Data Source record should be placed on the generated transaction. Retain FY must be <i>true</i> when Allocation Frequency is <i>Monthly</i> , <i>Quarterly</i> or <i>Annual</i> .	No Default
Retain BFY (RET_BFY)	Optional. If Retain BFY is <i>true</i> , the BFY of the selected Data Source record is retained to the generated transaction, preventing the field from defaulting. Ensure BFY is populated on the Data Source if using this option.	No Default
Prior Year APD (PRY_YEAR_APD)	Conditional. For frequencies of Daily, Annual and Quarterly this period will be used on generated transactions instead of defaulting (e.g. using past APD that is still open). If Allocation Frequency is Monthly, the parameter is prohibited unless the Lapse FY Override parameter is True.	No Default
Perform COA Consolidation	When set to <i>true</i> , base records with identical COA combinations will be consolidated for base expansion purposes. Base Expansion records that are forward referenced to the same step will also be summarized by COA combination when generating associated pool expansion records.	Unchecked
Rebuild Summary in Compute Allocation (RBLD_CA_JRNL_SUM_FL)	Optional. When unchecked, the CSAL_JRNL_SUM table will not be rebuilt when the job's run mode is <i>Compute Allocation</i> .	True
Perform COA Summarization (COA_SUM_FL)	When set to <i>true</i> , the Base Accumulation and Compute Allocation run modes will perform the extra step of re-summarizing the records based on roll up setups.	Unchecked

Major Output

Three reports are generated.

- Pool/Base verification report:

The report summarizes the data entered into the Pool/Base Setup & Distribution table. It gives the pool and base definitions.

- COA editing report:
The report provides the list of the invalid COA element values that were entered in the Pool/Base Distribution table. Validations only occur for those COA elements whose rollup type is Code, that is, not an actual Rollup. However, the Rollups are only validated for Pool Distribution records. The report checks the COA active status and compares the Parameter's Offline Validation date to the effective dates of the COA. If Department is not entered on the distribution record, all COAs keyed by Department will be deemed invalid on the report.
- Exception report:
There are seven exceptions that could appear on this report:
 1. Base Record Percentages do not sum to 100%. This will apply to Fixed % Base Types only.
 2. Accounting Distribution not found: where a pool/base has been set up on Pool/Base Setup, but no accounting distributions exist on Pool/Base Distribution.
 3. Accumulation Distribution not found: where base type = Direct Financial, but no accumulation distributions have been set up.
 4. Forward Reference Step number does not exist (Fwd Step is not defined as a step in Step Control Setup table).
 5. Neither Pool nor Base defined for Step.
 6. Bases not defined for Step (Only pool exists).
 7. Base Record Percentages for the Step sums up to the value greater than 100%. This will apply to Fixed % Base Types only

Job Return code

The following table shows the potential job return codes for the Cost Allocation Process job.

Return Code	Condition
Successful (1)	COA elements validated successfully.
Warning (4)	N/A
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for unexpected situations. • Exception Report has one of the seven exceptions. When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Sequence

- Records are sorted in order of Series, Step, Pool/Base Indicator and Pool/Base Sequence on Cost Allocation Expansion Results (CSAL_EXP_N_RSLT).

Selection Criteria

N/A

Problem Resolution

No table updates take place when the process runs in Offline Validation Run Mode. So there is no need to back out any updates if this job fails in any of the steps. If the job fails due to the exceptions written on the Exception report, then such exceptions need to be addressed manually in one or more of the setup pages.

The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Required Parameters are not entered. Sample Message: Cost Allocation Parameter ID cannot be left blank.	Enter the Cost Allocation Parameter ID and schedule a new job.	Job should be rescheduled after correcting the parameters.
	Entered Parameters are not valid. Sample Message: Submit file should have file extension of .txt	Enter the correct Submit file name and schedule a new job.	
	ALOC_ID record not found. Sample Message: Cost	Make sure that the CSAL_PROC_PARM record exists for the ALOC_ID entered on the	

Possible Return Codes	Condition	Recommendation	Other Instructions
	Allocation Parameter xxxx (xxxx being the value from the parameter) not found on CSAL_PROC_PARM.	batch parameter and schedule a new job.	
	Runtime Exception occurred. Sample Message: Error occurred while getting parameters from CSAL_PROC_PARM: <Runtime Exception Message>	Reason for Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out the possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 2: Report Generation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the reports generated successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during Report Generation: <Runtime Exception Message>	Reason for Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out the possible reason of exception.
	Exceptions found in Exception Report	Individual exceptions need to be addressed on the	Exception Report should be

Possible Return Codes	Condition	Recommendation	Other Instructions
	Message: One or more exceptions found in the Validation Exception Report.	respective pages.	investigated to determine the reason(s) for the exception(s).
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Base Accumulation Mode:

The following table shows the various steps that the Cost Allocation Process Job goes through and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> Validating Batch Parameters. Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. It will be followed by message "Batch Parameter validation failed." Batch Parameters validation completed.
2. Purge data from tables	<ul style="list-style-type: none"> Purging data from CSAL_JRNL_SUM table started Purging data from CSAL_JRNL_SUM table completed Purging previous run data from CSAL_JRNL_SUM table started (if Parameter value for previous CAPA ID is supplied) Purging previous run data from CSAL_JRNL_SUM table completed (if Parameter value for previous CAPA ID is supplied) Purging data from CSAL_TOT table started Purging data from CSAL_TOT table completed Purging data from previous run CSAL_TOT table started (if Parameter value for previous CAPA ID is supplied) Purging data from previous run CSAL_TOT table completed (if Parameter value for previous CAPA ID is supplied)
3. Build Cost Allocation Journal	<ul style="list-style-type: none"> Building of CSAL_JRNL_SUM table started If the selection returns 0 records, then the following

Process Steps	Messages
Summary	<p>message will be issued: "No eligible record found in 'LDGR_xyz' table".</p> <ul style="list-style-type: none"> • Number of records loaded into CSAL_JRNL_SUM table : 'n' • Building of CSAL_JRNL_SUM table completed.
4. Accumulate Base Record Dollar and Calculate Base Percentage	<ul style="list-style-type: none"> • Base Accumulation started updating Statistical Units for all bases. • Base Accumulation completed updating Statistical Units for all bases.
5. Creation of Base Accumulation Report XML	<ul style="list-style-type: none"> • Creating XML for Base Accumulation Report started • Creating XML for Base Accumulation Report completed
6. Report Generation	<ul style="list-style-type: none"> • Generation of Base Accumulation Report started • Generation of Base Accumulation Report completed • Generation of Base Accumulation Exception Report started • Generation of Base Accumulation Exception Report completed

Restartability Information

The job cannot be restarted in this run mode. If the job fails in any of the above steps, a new job should be scheduled after correcting the errors that caused the job to fail.

Major Input

Tables

- Cost Allocation Process Parameters (CSAL_PROC_PARM) - The Data Source, Fiscal Year and/or month and/or quarter and/or date and/or Budget Fiscal Year is defined on this parameter table.
- Prior to the processing of any data, the records from the Cost Allocation Journal Summary (CSAL_JRNL_SUM) table and the Cost Allocation Total (CSAL_TOT) table for the supplied CAPA ID are purged. Also, the records inside of the Cost Allocation Journal Summary (CSAL_JRNL_SUM) table and Cost Allocation Total (CSAL_TOT) table for the supplied Purge Previous Run CAPA ID will also be purged.
- Cost Allocation Journal Summary (CSAL_JRNL_SUM): Base records are compared against this table.
- Cost allocation Control Setup (R_CSAL_CTRL_SETP) - Input table for parameters like Allocation Cycle. The Allocation is defined as a Revenue, Expenditures, Charges or Charges Back Allocation on this table.
- Cost Allocation Step Control (R_CSAL_STPCTR_SETP) - Steps with Base Type (Direct Financial, Direct Instream) are taken from this table.
- Posting Code (R_PSCD) - The Cost Allocation value of each posting code is set on this table. Valid Values are 1) N/A, 2) Cash Expenditures, 3) Collected Revenues, and 4) Charges.

- Object Rate Group (OBJ_RT) - Object Rate groups are specified on this table. If Object Rate Groups are defined, the corresponding values are retrieved from this table and populated on the CSAL Journal Summary table.
- Pool/Base Distribution (R_PLBS_DIST) - Accumulator distributions for Base records are taken from this table.
- Pool/Base Setup (R_PLBS_SETP) - Allocation Percentage is computed for all base records whose base type is not fixed percentage. Base records are taken from this table.
- Unit Rollup (R_CA_UNIT_RLUP) - When the Perform COA Summarization flag is selected, the Unit Rollup table will be used to reassign the Unit on the original record of the Cost Allocation Summary (CSAL_JRNL_SUM) table. For each unique combination of Budget Fiscal Year, Department, Bureau, and Program, identify the Unit value to be used in the replacement process.
- Department Object Rollup (R_CA_DOBJ_RLUP) - When the Perform COA Summarization flag is selected, the Department Object Rollup table will be used to reassign the Department Object of the original record on the Cost Allocation Summary (CSAL_JRNL_SUM) table. For each unique combination of Budget Fiscal Year, Department, Bureau, Program, and Department Object Category, identify the Department Object value to be used in the replacement process.
- Department Object Rollup Exception (R_CA_DOBJ_RLUP_EX) - When the Perform COA Summarization flag is selected, the Department Object Rollup Exception table will be used to prevent reassignment of the Department Object of the original record on the Cost Allocation Summary (CSAL_JRNL_SUM) table in the replacement process.

Batch Parameters

The batch parameters for this mode are not different than Offline Validation.

Custom Parameters

The following table lists any differences on the Cost Allocation Process Parameter ID necessary for Base Accumulation. The parameter record should remain unchanged from the prior mode except for:

Parameter	Description	Default Value
Run Mode (RUN_MOD)	Required. Select <i>Base Accumulation</i> mode.	No Default

Major Output

The following tables are updated:

- Cost Allocation Journal Summary (CSAL_JRNL_SUM)
- Cost Allocation Journal Summary is updated with all selected records from the Cost Allocation ledger used as an input to the process. A new dataset overwrites the pre-existing dataset every time the process is run in this mode. Object Rate group fields are populated based on the values obtained from the Object Rate Group table.
- Cost Allocation Totals (CSAL_TOT)
- Records are also created on Cost Allocation Totals. Those records are overwritten every time the process is run in the Base Accumulation mode.

Two Reports are generated:

- Base Accumulation Report

This report displays all base records with the allocation percentages and accumulated base amounts. When Additional Reporting fields parameter is set to Yes, the Report will include Program and Function data.

- Base Accumulation Exception Report

The report displays the following three exceptions:

- Base Percentages for the Step sum to a negative value.
- Base Percentages for the Step sum to Zero.
- Individual bases with the percentage Zero. → This is a warning – a single base can have a percentage of zero and thus that base will not receive any allocations.

The first two exceptions will be problems for a given allocation. Cost Allocation requires that the overall percentage sums to 100%, and for Direct allocations it requires an overall positive total for Accumulated amounts. For these exceptions, suggested remedies are as follows:

- **Fixed Percentage or Statistical Allocation** - Correct the setup of your Allocation Percentages for your Base Accumulation records. (For Statistical allocations, correct your Statistical Units so that they do not result in a total that is negative or zero.)
- **Direct or Direct & Instream Allocation** - For expenditure allocations, these exceptions indicate that significant credit(s) (likely due to corrections or refunds) were posted during the allocation period in amounts equal to or exceeding the total costs incurred for the bases during the period. The following course of action is recommended:
 1. Examine the setup of your bases to be sure your Base Accumulation records are established correctly.
 2. Run reports to determine the cause of the negative calculation (e.g., large credits posted to the wrong accounts that can be corrected to remove the exception).
 3. If your bases are determined to legitimately result in a negative or zero base total:
 - a. Consider performing this allocation less frequently so that more costs will accumulate in the allocation period to offset any credits.
 - b. Consider handling the allocation manually for this allocation period.

Job Return code

The following table shows the potential job return codes for the Cost Allocation Process job.

Return Code	Condition
Successful (1)	All of the Cost Allocation Leger records processed successfully.
Warning (4)	No eligible records found. This could be because of the following reasons: <ul style="list-style-type: none"> • No eligible record found in 'LDGR_xyz' table • If the Allocation Frequency is Date Range and the Transaction Record Date is not selected for

Return Code	Condition
	summarization on the JLCTRL table, then the system will end the job with a return code of 'Warning', and the job log will have a 'No eligible records found' message. If there are no records on the ledger with a Transaction Record Date that falls within the specified date range on CAPA (based on the Selection From Date and Selection To Date values in the CAPA table), the system will end the job with a return code of 'Warning', and the job log will have a 'No eligible records found' message.
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: <ul style="list-style-type: none"> Parameters are invalid. Run time exceptions for unexpected situations. When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated, subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Sequence

- Records are sorted in order of Series, Step, Pool/Base Indicator and Pool/Base Sequence on Cost Allocation Expansion Results (CSAL_EXPN_RSLT).

Selection Criteria

Select all ledger records specified by the user in the Data Source field where accounting period falls in the range of the input fiscal year/fiscal quarter/accounting period/Transaction Record Date range on the custom parameter record (CSAL_PROC_PARM).

If the allocation is an expenditures/charge back allocation:

- Select only those records whose posting codes (R_PSCD) have a Cost Allocation field value of 'Expenditure'.

If the allocation is a revenue allocation:

- Select only those records whose posting codes (R_PSCD) have a Cost Allocation field value of 'Revenue'.

Problem Resolution

There is no need to back out any updates if this job fails in any of the steps since it updates only the temporary table which gets cleared when the job is rescheduled next time. The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Required Parameters are not entered. Sample Message: Cost Allocation Parameter ID can not be left blank.	Enter the Cost Allocation Parameter ID and schedule a new job.	Job should be rescheduled after correcting the parameters.
	Entered Parameters are not valid. Sample Message: Submit file should have file extension of .txt	Enter the correct Submit file name and schedule a new job.	
	ALOC_ID record not found. Sample Message: Cost Allocation Parameter xxxx (xxxx being the value from the parameter) not found on CSAL_PROC_PARM table	Make sure that the CSAL_PROC_PARM record exists for the ALOC_ID entered on the batch parameter and schedule a new job.	
	Runtime Exception occurred. Sample Message: Error occurred while getting parameters from CSAL_PROC_PARM table: <Runtime Exception Message>	The reason for Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out the possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Possible Return Codes	Condition	Recommendation	Other Instructions
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 2: Purge data from tables

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the tables purged successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during Reset Program: <Runtime Exception Message>	The reason for Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out the possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 3: Build Cost Allocation Journal Summary

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the eligible records were inserted into the Cost Allocation Journal Summary table successfully.	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Warning (4)	No eligible record found in Ledger Table. Sample Message: No eligible record found in 'LDGR_xyz' table	Make sure that the LDGR_xyz table has eligible records for entered parameters. Correct the problem and schedule a new job.	Job should be rescheduled after correcting the problem.
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred while loading data into CSAL_JRNL_SUM table: <Runtime Exception Message>	The reason for Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out the possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 4: Accumulate Base Record Dollar and Calculate Base Percentage

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Base percentage calculated successfully for all base records whose base type is not fixed percentage.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during Base Accumulation: <Runtime Exception Message>	The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 5: Creation of Base Accumulation Report XML

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Report XML generated successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during Base Accumulation: <Runtime Exception Message>	The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out the possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated.	Job should be rescheduled after correcting

Possible Return Codes	Condition	Recommendation	Other Instructions
		Schedule a new job.	the problem.

Step 6: Report Generation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the reports generated successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error Message : <Runtime Exception Message>	The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out the possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Compute Allocation Mode:

The following table shows the various steps that the Cost Allocation Process Job goes through and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> Validating Batch Parameters. Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. It will be followed by message "Batch Parameter validation failed". Batch Parameters validation completed.
2. Purge data from	<ul style="list-style-type: none"> Purging data from CSAL_JRNL_SUM table started

Process Steps	Messages
tables	<ul style="list-style-type: none"> • Purging data from CSAL_JRNL_SUM table completed • Purging previous run data from CSAL_JRNL_SUM table started (if Parameter value for previous CAPA ID is supplied) • Purging previous run data from CSAL_JRNL_SUM table completed (if Parameter value for previous CAPA ID is supplied) • Purging data from CSAL_TOT table started • Purging data from CSAL_TOT table completed • Purging data from previous run CSAL_TOT table started (if Parameter value for previous CAPA ID is supplied) • Purging data from previous run CSAL_TOT table completed (if Parameter value for previous CAPA ID is supplied) • Purging data from CSAL_EXP_N_RSLT table started • Purging data from CSAL_EXP_N_RSLT table completed • Purging data from previous run CSAL_EXP_N_RSLT table started (if Parameter value for previous CAPA ID is supplied) • Purging data from previous run CSAL_EXP_N_RSLT table completed (if Parameter value for previous CAPA ID is supplied)
3. Build Cost Allocation Journal Summary	<ul style="list-style-type: none"> • Building of CSAL_JRNL_SUM table started • If the selection returns 0 records, then the following message will be issued: "No eligible record found in 'LDGR_xyz' table". • Number of records loaded into CSAL_JRNL_SUM table: 'n' • Building of CSAL_JRNL_SUM table completed.
4. Pool Expansion	<ul style="list-style-type: none"> • Pool Expansion started • Expanding Pool records for ALOC_ID : 'xxx', SERIES: 'yyy' and STEP: 'm' • If no eligible record found on R_PLBS_SETP and R_PLBS_DIST tables, then the following message will be issued: "No eligible record found for Pool record Expansion" • 'n' Pool records expanded (where n being the progression counter size) • Total Number of Pool records expanded: 'n' • Pool Expansion completed
5. Base Expansion	<ul style="list-style-type: none"> • Base Expansion started • Expanding Base records for ALOC_ID: 'xxx', SERIES: 'yyy' and STEP: 'm'

Process Steps	Messages
	<ul style="list-style-type: none"> • If no eligible record found on R_PLBS_SETP and R_PLBS_DIST tables, then the following message will be issued: “No eligible record found for Base record Expansion” • ‘n’ Base records expanded (where n being the progression counter size) • Total Number of Base records expanded: ‘n’ • Base Expansion completed
6. Delete Pool records with no expanded Base	<ul style="list-style-type: none"> • Purging Pool records with no expanded Base from CSAL_EXP_N_RSLT table started • Purging Pool records with no expanded Base from CSAL_EXP_N_RSLT table completed
7. Pool and Base View Report Generation	<ul style="list-style-type: none"> • Generation of Expansion Exception Report started • Generation of Expansion Exception Report completed • Generation of Pool View Report started • Generation of Pool View Report completed • Generation of Base View Report started • Generation of Base View Report completed
8. Offset routine	<ul style="list-style-type: none"> • Offset Routine for Pool records with Offset distributions started. • If no eligible record is found on the CSAL_EXP_N_RSLT table, then the following message will be issued: “No eligible Pool record updated by the Offset routine” • Total number of expanded Pool records updated by the Offset routine: ‘n’. • Offset Routine for Pool records with Offset distributions completed.
9. Summarization	<ul style="list-style-type: none"> • Summarization of expanded records into CSAL_HIST table started • Purging data from CSAL_HIST table started • Purging data from CSAL_HIST table completed • Inserting data into CSAL_HIST table started • Inserting records into CSAL_HIST table for ALOC_ID: ‘xxx’ , SERIES: ‘yyy’ and STEP: ‘m’ • If no eligible record found on CSAL_EXP_N_RSLT table, then the following message will be issued: “No eligible record inserted into CSAL_HIST table” • ‘n’ records inserted (where n being the progression counter size) • Total number of records inserted into CSAL_HIST table: ‘n’ • Inserting data into CSAL_HIST table completed • Summarization of expanded records into CSAL_HIST

Process Steps	Messages
	table completed
10. Summarization Report Generation	<ul style="list-style-type: none"> • Generation of Summarization Report started • Generation of Summarization Report completed

Restartability Information

The job can be restarted if it has failed. If a restart is not the immediate option, then a new job can be rescheduled, but before rescheduling the job, the updates made by the failed job should be backed out.

Overview of Restartability in Compute Allocation mode:

Pool Expansion Restartability:

If the job fails during Pool Expansion (that is, before or during Pool Expansion), then you can restart the job after resolving the error. If the job is restarted, it will start from the step or pool expansion where it failed.

If the job fails after completing Pool Expansion due to some fatal condition, then if the job is restarted after resolving the issue, the job will start in Base Expansion as Pool Expansion has already completed.

Base Expansion Restartability:

If the job fails during Base Expansion (that is, before or during Base Expansion) then the job can be restarted after resolving any applicable error. If the job is restarted, it will start from the step or base expansion where it failed.

If the job fails after completing Base Record Expansion due to some fatal condition, then if the job is restarted after resolving the issue, the job will start Report Generation (that is, Expansion Exception, Pool View, and Base View Reports) as Base Expansion has already completed.

Offset Pool Routine Restartability:

If the job fails during the Offset Routine for Pool records with Offset distributions (that is, before or during Offset Routine) then you can restart the job after resolving the error. Even if the Offset Pool Routine is unsuccessful, summarization will be performed; since Offset Routine is optional and not necessary. If the job is restarted, it will start from the step or Offset Routine where it failed.

If the job fails after the Offset Routine for Pool records with Offset distributions due to some fatal condition, then if the job is restarted after resolving the issue, the job will start Summarization of expanded records into the Cost Allocation History table.

Summarization Restartability:

Restartability logic is not required for Summarization as it will always purge records from the Cost Allocation History table where Transaction Code and Reversal Agent ID are blank for a particular Parameter Id and it will insert records into the CA History table after summarizing CA Expansion records.

Major Input

Tables

- Cost Allocation Process Parameters (CSAL_PROC_PARM) – Selection parameters are used as input.
- Pool/Base Setup (R_PLBS_SETP) – Pool/Base records are taken from this table.
- Pool/Base Distribution (R_PLBS_DIST) - COA Elements for the Direct and Instream Financial base accumulation records are taken from this table.
- Cost Allocation Journal Summary (CSAL_JRNL_SUM) - Pool Records are compared against this table.
- Cost Allocation Control Setup (R_CSAL_CTR_SETP) - Used to check inheritance of COA Element.
- Cost Allocation Series Control Setup (R_CSAL_SRCTR_SETP) - Used to check inheritance of COA Element.
- Cost Allocation Step Control Setup (R_CSAL_STPCTR_SETP) - Used to check inheritance of COA Element. Also the process reads the Base Type field to determine if the Instream Amount needs to be added to the previously accumulated amount from this table.
- Cost Allocation Expansion Results (CSAL_EXPN_RSLT) - Base Records are expanded against pool records in this table. Also expanded Pool record and pool amount to be allocated, base record percentage for computing allocation are taken from this table.
- Unit Rollup (R_CA_UNIT_RLUP) - When the Perform COA Summarization flag is selected, the Unit Rollup table will be used to reassign the Unit on the original record of the Cost Allocation Summary (CSAL_JRNL_SUM) table in the rebuild process. For each unique combination of Budget Fiscal Year, Department, Bureau, and Program, identify the Unit value to be used in the replacement process.
- Department Object Rollup (R_CA_DOBJ_RLUP) - When the Perform COA Summarization flag is selected, the Department Object Rollup table will be used to reassign the Department Object of the original record on the Cost Allocation Summary (CSAL_JRNL_SUM) table in the rebuild process. For each unique combination of Budget Fiscal Year, Department, Bureau, Program, and Department Object Category, identify the Department Object value to be used in the replacement process.
- Department Object Rollup Exception (R_CA_DOBJ_RLUP_EX) - When the Perform COA Summarization flag is selected, the Department Object Rollup Exception table will be used to prevent reassignment of the Department Object of the original record on the Cost Allocation Summary (CSAL_JRNL_SUM) table in the rebuild process.

Batch Parameters

The batch parameters for this mode are not different than Offline Validation.

Custom Parameters

The following table lists any differences on the Cost Allocation Process Parameter ID necessary for Compute Allocation. The parameter record should remain unchanged from the prior mode except for: *Data Object Name: CSAL_PROC_PARM*

Parameter	Description	Default Value
Run Mode (RUN_MOD)	Required. Select <i>Compute Allocation</i> mode.	No Default

Major Output

Three tables are updated.

- **Cost Allocation Journal Summary (CSAL_JRNL_SUM)**
All records on the Journal Summary table for the supplied Parameter ID that have been allocated are updated. The amount to be allocated is reduced to zero so that those records cannot be re-allocated if the process is repeated in Compute Allocations mode without first re-running in Base Accumulation mode.
- **Cost Allocation Expansion Results (CSAL_EXPN_RSLT)**
The results of Pool and Base Expansion are stored on the Cost Allocation Expansion Results table. Pool records are displayed first for each step. The base records display the final allocation percentage and allocation amount.
- **Cost Allocation History (CSAL_HIST)**
Cost Allocation History records are created. To ease in processing of generated accounting lines at a later stage, records appended to this table from the Expansion Results table and summarized if there are duplicate combinations of COA elements.

When summarizing, the respective ledger REC_NO on the pool records are concatenated. This value is set on the LDGR_REC_NO column.

Later, it will be set on the respective Accounting Line description on the generated CA transaction in the Generate Transaction run mode.

Four Reports are generated.

- **Pool View Report**
The report gives a view of Pool record and the base records to which the pool amount was allocated. When Additional Reporting fields parameter is set to Yes, the Report will include Department Object, Appropriation Unit and Function data.
- **Base View Report**
The report gives a view of each base record with the corresponding pool record from which the amount was allocated. When Additional Reporting fields parameter is set to Yes, the Report will include Department Object, Appropriation Unit and Function data.
- **Summarization Report**
The report gives a summary of Pool and base records. When Additional Reporting fields parameter is set Yes, the Report will include Department Object, Appropriation Unit and Function data.
- **Expansion Exception Report**
The report lists the pool records that are not picked up during expansion, base records that have no pool records in step, base records with allocation percentage zero, records with an improper allocation of pool amount to base and records with a sum of all base amount in a step is negative. There are five exceptions:
 - Pool Definition did not pick Journal record during expansion
 - Zero pool in step
 - Base with Allocation Percentage Zero

- Improper Allocation of Pool amount to Base
- Sum of all of the Base amounts in a Step is Negative

Note: Compute Allocations actually reruns the process in Base Accumulation first and then perform the extra processing described above.

Job Return code

The following table shows the potential job return codes for the Cost Allocation Process job.

Return Code	Condition
Successful (1)	All of the Pool/Base records processed successfully.
Warning (4)	No eligible records found. This could be because of the following reasons: <ul style="list-style-type: none"> • No eligible record found in 'LDGR_xyz' table • No eligible record found for Pool record Expansion • No eligible record found for Base record Expansion • No eligible record inserted into CSAL_HIST table
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for unexpected situations. When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated, subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Sequence

Records are sorted in order of Series, Step, Pool/Base Indicator and Pool/Base Sequence on the Cost Allocation Expansion Results (CSAL_EXPN_RSLT).

Selection Criteria

For Pool Expansion: Select all records from the Cost Allocation Journal Summary Table whose CAPA ID is the same as the supplied Parameter ID. Select all pool accounting distributions from Pool/Base Distribution for the allocation ID to determine if those records can be expanded against the Cost Allocation Journal Summary records.

For Base Expansion: Select all pool expansion records from Cost Allocation Expansion Results whose CAPA ID is the same as the supplied Parameter ID. Select all base accounting distributions from Pool/Base Distribution for the allocation ID for base expansion purposes.

For Instream Calculation: Select all Cost Allocation Expansion Results 'base' records whose CAPA ID is the same as the supplied Parameter ID and where base type = Direct and Instream Financial. Compute the Instream amount and subsequently the allocation percentage.

For Compute Allocation Amount: Select all Cost Allocation Expansion Results 'base' records whose CAPA ID is the same as the supplied Parameter ID. Compute the allocation amount using the allocation percentage for each base record on Pool/Base Setup.

Problem Resolution

There is no need to back out any updates if this job fails in any of the steps since it updates only the temporary table which gets cleared when the next time the job is rescheduled. The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Required Parameters are not entered. Sample Message: Cost Allocation Parameter ID cannot be left blank.	Enter the Cost Allocation Parameter ID and schedule a new job.	Job should be rescheduled after correcting the parameters.
	Entered Parameters are not valid. Sample Message: Submit file should have file extension of .txt	Enter the correct Submit file name and schedule a new job.	
	ALOC_ID record not found. Sample Message: Cost Allocation Parameter xxxx (xxxx being the value from the parameter) not found on CSAL_PROC_PARM table	Make sure that the CSAL_PROC_PARM record exists for the ALOC_ID entered on the batch parameter and schedule a new job.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	Runtime Exception occurred. Sample Message: Error occurred while getting parameters from CSAL_PROC_PARM table: <Runtime Exception Message>	The reason for Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 2: Purge data from tables

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the tables purged successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during Reset Program: <Runtime Exception Message>	The reason for Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System	When the job is terminated because of database server	The reason for the System Failure needs to be	Job should be rescheduled

Possible Return Codes	Condition	Recommendation	Other Instructions
Failure (20)	or network issues.	investigated. Schedule a new job.	after correcting the problem.

Step 3: Build Cost Allocation Journal Summary

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the eligible records were inserted into the Cost Allocation Journal Summary table successfully.	N/A	N/A
Warning (4)	No eligible record found in Ledger Table. Sample Message: No eligible record found in 'LDGR_xyz' table	Make sure that the LDGR_xyz table has eligible records for entered parameters. Correct the problem and schedule a new job.	Job should be rescheduled after correcting the problem.
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred while loading data into CSAL_JRNL_SUM table: <Runtime Exception Message>	The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 4: Pool Expansion

Possible Return Codes	Condition	Recommendation	Other instructions
Successful (1)	All Pool records expanded successfully.	N/A	N/A
Warning (4)	No eligible record found for Pool Expansion. Sample Message: No eligible record found for Pool record Expansion	Make sure that Pool/Base Setup and Distribution Table have pool records for the entered ALOC_ID. Correct the problem and schedule a new job.	Job should be rescheduled after correcting the problem.
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred while expanding pool records: <Runtime Exception Message>	The reason for the Runtime Exception needs to be investigated. Correct the problem restart the job, or schedule a new job.	System error log and VLS log should be investigated to find out the possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job or restart the job if applicable.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 5: Base Expansion

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All Pool records expanded successfully.	N/A	N/A
Warning (4)	No eligible record found for Base Expansion. Sample Message: No eligible record found for	Make sure that Pool/Base Setup and Distribution Table have base records for the entered ALOC_ID. Correct the problem and	Job should be rescheduled after correcting the problem.

Possible Return Codes	Condition	Recommendation	Other Instructions
	Base record Expansion	schedule a new job.	
Non Fatal Error (8)	N/A	This step doesn't issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during Base Expansion: <Runtime Exception Message>	The reason for Runtime Exception needs to be investigated. Correct the problem and restart the job, or schedule a new job.	System error log and VLS log should be investigated to find out the possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job or restart the job if applicable.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 6: Delete Pool records with no expanded Base

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All Pool records expanded successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: CleanUp delete failed for CA_EXPN_RSLT table during delPoolWithNoBase method: <Runtime Exception Message>	The reason for Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out possible reason of exception.

Possible Return Codes	Condition	Recommendation	Other Instructions
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 7: Pool and Base View Report Generation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the reports generated successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during genPoolViewRep method: <Runtime Exception Message>	The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out the possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 8: Offset Routine

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All Pool records with Offset distributions records processed successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during offset for Pools: <Runtime Exception Message>	The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 9: Summarization

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All Pool/Base records summarized successfully.	N/A	N/A
Warning (4)	No eligible record inserted into CSAL_HIST table. Sample Message: No eligible record inserted into CSAL_HIST table	Make sure that the Pool/Base Setup and Distribution Table have pool/base records for the entered ALOC_ID. Correct the problem and schedule a new job.	Job should be rescheduled after correcting the problem.
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Failed (12)	<p>Runtime Exception occurred.</p> <p>Sample Message: Error occurred during Summarization: <Runtime Exception Message> Error occurred during setLdgrRecNo method: <Runtime Exception Message></p>	<p>The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.</p>	<p>System error log and VLS log should be investigated to find out the possible reason of exception.</p>
Terminated (16)	<p>Job is terminated manually by the user.</p>	<p>The reason for the termination needs to be investigated. Schedule a new job.</p>	<p>Job should be rescheduled after correcting the problem.</p>
System Failure (20)	<p>When the job is terminated because of database server or network issues.</p>	<p>The reason for the System Failure needs to be investigated. Schedule a new job.</p>	<p>Job should be rescheduled after correcting the problem.</p>

Step 10: Summarization Report Generation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	<p>All of the reports generated successfully.</p>	<p>N/A</p>	<p>N/A</p>
Warning (4)	<p>N/A</p>	<p>This step does not issue this return code.</p>	<p>N/A</p>
Non Fatal Error (8)	<p>N/A</p>	<p>This step does not issue this return code.</p>	<p>N/A</p>
Failed (12)	<p>Runtime Exception occurred.</p> <p>Sample Message: Error Message : <Runtime Exception Message></p>	<p>The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.</p>	<p>System error log and VLS log should be investigated to find out the possible reason of exception.</p>
Terminated (16)	<p>Job is terminated manually by the user.</p>	<p>The reason for the termination needs to be investigated. Schedule a new job.</p>	<p>Job should be rescheduled after correcting the problem.</p>
System	<p>When the job is terminated</p>	<p>The reason for the System</p>	<p>Job should be</p>

Possible Return Codes	Condition	Recommendation	Other Instructions
Failure (20)	because of database server or network issues.	Failure needs to be investigated. Schedule a new job.	rescheduled after correcting the problem.

Generate Transaction Mode:

The following table shows the various steps that the Cost Allocation Process Job goes through and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> Validating Batch Parameters. Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. It will be followed by message "Batch Parameter validation failed". Batch Parameters validation completed.
2. Transaction XML Generation	<ul style="list-style-type: none"> Generation of Transaction XML started Selection of eligible records from CSAL_HIST table started If no eligible record found on CSAL_EXP_N_RSLT table, then the following message will be issued: "No eligible record found for Allocation ID 'xxx' in CSAL_HIST table" Selection of eligible records from CSAL_HIST table completed Generating CA transactions for ALOC_ID : 'xxx', SERIES : 'yyy' and STEP : 'm' 'n' CA transactions generated where n being the progression counter size Total Number of CA transactions generated : 'n' Generation of Transaction XML completed
3. Update Cost Allocation History Table	<ul style="list-style-type: none"> Updating CSAL_HIST table with generated Transaction information started Updating CSAL_HIST table with generated Transaction information completed
4. Backup data and clean up	<ul style="list-style-type: none"> About to migrate historical data for table CSAL_HIST into CSAL_HIST_STRG Copied ##### records from CSAL_HIST into longer term storage table CSAL_HIST_STRG About to migrate historical data for table CSAL_EXP_N_RSLT into CSAL_EXP_N_RSLT_STRG Stored ##### records from CSAL_EXP_N_RSLT into longer term storage table CSAL_EXP_N_RSLT_STRG

Process Steps	Messages
	<ul style="list-style-type: none"> • About to migrate historical data for table R_CSAL_CTRL_SETP into R_CSAL_CTRL_SETP_STRG • Stored ##### records from R_CSAL_CTRL_SETP into longer term storage table R_CSAL_CTRL_SETP_STRG • About to migrate historical data for table R_CSAL_SRCTR_SETP into R_CSAL_SRCTR_SETP_STRG • Stored ##### records from R_CSAL_SRCTR_SETP into longer term storage table R_CSAL_SRCTR_SETP_STRG • About to migrate historical data for table R_CSAL_STPCTR_SETP into R_CSAL_STPCTR_SETP_STRG • Stored ##### records from R_CSAL_STPCTR_SETP into longer term storage table R_CSAL_STPCTR_SETP_STRG • About to migrate historical data for table R_PLBS_DIST into R_PLBS_DIST_STRG • Stored ##### records from R_PLBS_DIST into longer term storage table R_PLBS_DIST_STRG • About to migrate historical data for table R_PLBS_SETP into R_PLBS_SETP_STRG • Stored ##### records from R_PLBS_SETP into longer term storage table R_PLBS_SETP_STRG • If Purge Previous Run CAPA ID is supplied then, Purging data from CSAL_HIST table for CAPA ID <<Purge Previous Run CAPA ID>> • If Purge Previous Run CAPA ID is supplied then, Purging prior successful runs of data from CSAL_HIST table completed

Restartability Information

The job can be restarted if it fails. If a restart is not the immediate option, then a new job can be rescheduled; but before rescheduling the job, the updates made by the failed job should be backed out.

If the job fails in Transaction XML Generation then the user can restart the job after resolving any applicable error. If the job is restarted, it will start from the last generated transaction where it failed.

Note: Transactions generated after restart are added to the same CAExport_<<CAPA_ID>>.xml file generated by the failed run.

Major Input

Tables

- Cost Allocation History (CSAL_HIST) - Summarized accounting lines for which transactions have not previously been generated are located on this table; records whose CAPA ID is the same as the supplied Parameter ID.
- Cost Allocation Process Parameters (CSAL_PROC_PARM) – Transaction information like Transaction Code, Transaction Type, etc. from this table.
- Program Transaction Cross Reference (R_PROG_DOC_CRISREF) - Reference transaction information will be populated on the accounting line generated, matching the specific COA combination obtained from this table.

Batch Parameters

The batch parameters for this mode are not different than Offline Validation.

Custom Parameters

The following table lists any differences on the Cost Allocation Process Parameter ID necessary for Base Accumulation. The parameter record should remain unchanged from the prior mode except for:

Parameter	Description	Default Value
Run Mode (RUN_MOD)	Required. Select <i>Generate Transaction</i> mode.	No Default

Major Output

- Cost Allocation History (CSAL_HIST)
 - Once History table records have been translated into transaction accounting lines, the History table is updated to reflect this fact so that those records are not selected again if the process is re-run in Generate Transaction mode.
- Longer Term Retention tables are captured with associated Transactional tables, once the transactions have successfully been written to the XML file:
 - CSAL_EXP_N_RSLT_STRG
 - CSAL_HIST_STRG
 - R_CSAL_CTRL_SETP_STRG
 - R_CSAL_SRCTR_SETP_STRG
 - R_CSAL_STPCTR_SETP_STRG
 - R_PLBS_DIST_STRG
 - R_PLBS_SETP_STRG

One Transaction XML is generated.

- Cost Allocation transaction XML
 - The XML is created based on all records for the current allocation for which accounting lines have not previously been generated.

Job Return code

The following table shows the potential job return codes for the Cost Allocation Process job.

Return Code	Condition
Successful (1)	All of the Cost Allocation Leger records processed successfully.
Warning (4)	N/A
Non Fatal Error (8)	No eligible records found. This could be because of the following reason: <ul style="list-style-type: none"> No eligible record found for Allocation ID 'xyz' in CSAL_HIST table
Failed (12)	The job will fail under the following conditions: <ul style="list-style-type: none"> Parameters are invalid. Run time exceptions for unexpected situations. When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Sequence

- Records are sorted in order of Allocation ID, Series, Step, CAPA ID, Agent ID, and CSAL_HIST Sequence Number on the Cost Allocation History (CSAL_HIST) Table.

Selection Criteria

Select records for the Parameter ID and Allocation ID specified on the custom parameter record from the Cost Allocation History table. Select only those records that have not previously been selected in this run mode. The Cost Allocation Transaction prefix is populated on the History table records after the process completes in this run mode to denote that those records have already been selected.

Problem Resolution

There is no need to back out any updates if this job fails in any of the steps since it updates only the temporary table which gets cleared when the job is rescheduled next time. The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Required Parameters are not entered. Sample Message: Cost Allocation Parameter ID cannot be left blank.	Enter the Cost Allocation Parameter ID and schedule a new job.	Job should be rescheduled after correcting the parameters.
	Entered Parameters are not valid. Sample Message: Submit file should have file extension of .txt	Enter the correct Submit file name and schedule a new job.	
	ALOC_ID record not found. Sample Message: Cost Allocation Parameter xxxx (xxxx being the value from the parameter) not found on CSAL_PROC_PARM table	Make sure that the CSAL_PROC_PARM record exists for the ALOC_ID entered on the batch parameter and schedule a new job.	
	Runtime Exception occurred. Sample Message: Error occurred while getting parameters from CSAL_PROC_PARM table: <Runtime Exception Message>	The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 2: Transaction XML Generation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Cost Allocation Transaction XML generated successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	No eligible record found for Allocation ID 'xyz' in CSAL_HIST table. Sample Message: No eligible record found for Allocation ID 'xyz' in CSAL_HIST table	Make sure that the cost allocation history records for provided ALOC_ID exists in the CSAL_HIST table. Correct the problem and schedule a new job.	Job should be rescheduled after correcting the problem.
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during generating Transaction XML: <Runtime Exception Message>	The reason for the Runtime Exception needs to be investigated. Correct the problem and restart the job, or schedule a new job.	System error log and VLS log should be investigated to find out possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job or restart the job if applicable.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 3: Update Cost Allocation History Table

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All Cost Allocation History records updated successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during generating Transaction XML: <Runtime Exception Message>	The reason for Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The Reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 4: Backup data and clean up

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Tables were backed up into longer term retention tables and previous run CSAL_HIST records were purged.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during processing: <Runtime Exception Message>	The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to determine the possible reason for the exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be	Job should be rescheduled

Possible Return Codes	Condition	Recommendation	Other Instructions
		investigated. Schedule a new job.	after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The Reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Reversal Mode:

The following table shows the various steps that the Cost Allocation Process Job goes through and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> Validating Batch Parameters. Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. It will be followed by message "Batch Parameter validation failed". Batch Parameters validation completed.
2. Reverse Cost Allocation History records	<ul style="list-style-type: none"> Selection of eligible records from CSAL_HIST table for Reversal of CA transaction started If no eligible record found on CSAL_HIST table, then the following message will be issued: "No eligible record found with DOC_CD <> NULL and RVRSL_AGNT_ID = NULL in CSAL_HIST table " Selection of eligible records from CSAL_HIST table for Reversal of CA transaction completed Inserting the records into CSAL_HIST table for Reversal of CA transaction started 'n' records inserted (where n being the progression counter size) Inserting the records into CSAL_HIST table for Reversal of CA transaction completed
3. Reversal Transaction XML Generation	<ul style="list-style-type: none"> Generation of Transaction XML for Reversal of CA transaction started Selection of eligible records from CSAL_HIST table started If no eligible record found on CSAL_HIST table, then the following message will be issued: "No eligible record found for Allocation ID 'xxx' in CSAL_HIST table" Selection of eligible records from CSAL_HIST table completed Generating CA transactions for ALOC_ID: 'xxx',

Process Steps	Messages
	<p>SERIES: 'yyy' and STEP: 'm'</p> <ul style="list-style-type: none"> 'n' CA transactions generated (where n being the progression counter size) Total Number of CA transactions generated: 'n' Generation of Transaction XML for Reversal of CA transaction completed
4. Update Cost Allocation History Table	<ul style="list-style-type: none"> Updating CSAL_HIST table with generated Transaction information started Updating CSAL_HIST table with generated Transaction information completed
5. Backup Data	<ul style="list-style-type: none"> About to migrate historical data for table CSAL_HIST into CSAL_HIST_STRG Stored ##### records from CSAL_HIST into longer term storage table CSAL_HIST_STRG

Restartability Information

The job can be restarted if it is failed. If the restart is not the immediate option, then the new job can be rescheduled but before rescheduling the job, the updates done by the failed job should be backed out.

If the job fails during Reverse Cost Allocation History records, then the user can restart the job after resolving the error. If the job is restarted, it will start from the step where it failed.

Note: Transaction XML is not saved when the job fails. A restarted run for Reversal Transaction XML Generation will process all the transaction records before saving the XML file at the end.

Major Input

Table

- Cost Allocation History (CSAL_HIST) - Source of accounting lines for CA transactions previously generated.

Batch Parameters

The batch parameters for this mode are not different than Offline Validation.

Custom Parameters

The following table lists any differences on the Cost Allocation Process Parameter ID necessary for Base Accumulation. The parameter record should remain unchanged from the prior mode except for:

Parameter	Description	Default Value
Run Mode (RUN_MOD)	Required Field. Select <i>Reversal</i> Mode.	No Default

Major Output

Two tables are updated.

- Cost Allocation History Table (CSAL_HIST)
Once history records that were processed in Generate Transaction mode are reversed, the relevant Cost Allocation History table records are deleted.
- Cost Allocation History Storage Table (CSAL_HIST_STRG)
Records from CSAL_HIST table will get backed up into the CSAL_HIST_STRG table.

One Transaction XML is generated.

- Cost Allocation transaction XML
A Cost Allocation XML transaction is created to reverse the transactions created and processed in Generate Transaction mode.

Job Return Code

The following table shows the potential job return codes for the Cost Allocation Process job.

Return Code	Condition
Successful (1)	All of the Cost Allocation history records that were processed in Generate Transaction Mode are reversed successfully.
Warning (4)	N/A
Non Fatal Error (8)	No eligible records found. This could be because of the following reasons: <ul style="list-style-type: none"> • No eligible record found with DOC_CD <> NULL and RVRSL_AGNT_ID = NULL in CSAL_HIST table
Failed (12)	The job will fail under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid • Run time exceptions for unexpected situations. When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Sequence

Records are sorted in order of Allocation ID, Series ID, Step Number, CAPA ID, Agent ID, and CSAL_HIST Sequence Number on the Cost Allocation History (CSAL_HIST) table.

Selection Criteria

Select all records previously selected in the Generate Transaction run mode for the supplied Parameter ID.

Problem Resolution

If this job fails in any of the steps, the job can be restarted. The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Required Parameters are not entered. Sample Message: Cost Allocation Parameter ID can not be left blank.	Enter the Cost Allocation Parameter ID and schedule a new job.	Job should be rescheduled after correcting the parameters.
	Entered Parameters are not valid. Sample Message: Submit file should have file extension of .txt	Enter the correct Submit file name and schedule a new job.	
	ALOC_ID record not found Sample Message: Cost Allocation Parameter xxxx (xxxx being the value from the parameter) not found on CSAL_PROC_PARM table	Make sure that the CSAL_PROC_PARM record exists for the ALOC_ID entered on the batch parameter and schedule a new job.	
	Runtime Exception occurred. Sample Message: Error occurred while getting parameters from CSAL_PROC_PARM table: <Runtime Exception	The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	Message>		exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 2: Reverse Cost Allocation History records

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All Cost Allocation history records that were processed in Generate Transaction Mode are reversed successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	No eligible record found with DOC_CD <> NULL and RVRSL_AGNT_ID = NULL in CSAL_HIST table. Sample Message: No eligible record found for Allocation ID 'xyz' in CSAL_HIST table	Make sure that the cost allocation history records that were processed in Generate Transaction Mode for provided ALOC_ID exist in CSAL_HIST table. Correct the problem and restart the job.	Alternatively the job can be rescheduled as well after correcting the parameters. Before rescheduling the job, the updates done on the following table should be backed out: CSAL_HIST
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred while reversing CSAL_HIST	The reason for the Runtime Exception needs to be investigated. Correct the problem and restart the job.	Alternatively, the job can be rescheduled as well after correcting the

Possible Return Codes	Condition	Recommendation	Other Instructions
	records: <Runtime Exception Message>		parameters. Before rescheduling the job, the updates done on the following table should be backed out: CSAL_HIST
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job or restart the job if applicable.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 3: Reversal Transaction XML Generation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Cost Allocation Transaction XML generated successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	No eligible record found for Allocation ID 'xyz' in CSAL_HIST table. Sample Message: No eligible record found for Allocation ID 'xyz' in CSAL_HIST table	Make sure that the cost allocation history records for provided ALOC_ID exists in the CSAL_HIST table. Correct the problem and restart the job.	Alternatively, the job can be rescheduled as well after correcting the parameters. Before rescheduling the job, the updates done on the following table should be backed out:

Possible Return Codes	Condition	Recommendation	Other Instructions
			CSAL_HIST.
Failed (12)	<p>Runtime Exception occurred.</p> <p>Sample Message: Error occurred while generating Transaction XML for reversing CA transaction: <Runtime Exception Message></p>	<p>The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.</p>	<p>Alternatively the job can be rescheduled as well after correcting the parameters.</p> <p>Before rescheduling the job, the updates done on the following table should be backed out: CSAL_HIST</p>
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 4: Update Cost Allocation History Table

Possible Return Codes	Condition	Recommendation	Other instructions
Successful (1)	All Cost Allocation History records are updated successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	<p>Runtime Exception occurred.</p> <p>Sample Message: Error occurred while generating Transaction</p>	<p>The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.</p>	<p>System error log and VLS log should be investigated to find out possible reason of</p>

Possible Return Codes	Condition	Recommendation	Other instructions
	XML for reversing CA transaction: <Runtime Exception Message>		exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 5: Backup data and clean up

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Tables were backed up into longer term retention tables and previous run CSAL_HIST records were purged.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during processing: <Runtime Exception Message>	The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to determine the reason for the exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	The job is terminated because of database server or network issues.	The Reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Cost Allocation Process Chain: Load CA Job

Job Name	Load CA
Recommended Frequency	This job can be run daily, monthly, quarterly or on an annual basis.
Parallel processing enabled	No. Inside of a chain it is Single Instance; however, there can be many chains running simultaneously.
Can the job be restarted?	Optionally, based on the Save Restart Information parameter.
Exception report produced	No. All of the exceptions are only written to the log.

Overview

The Load CA job loads the records from the XML File, generated by the Cost Allocation Process job, into the Transaction Catalog. This job uses the common utility to load the records into the Transaction Catalog. This job first validates the batch parameters. If the parameters are valid, then it loads the records into the Transaction Catalog. If the parameters are not valid, the job issues appropriate messages and ends with a status of Failed. Once the records are loaded into the Transaction Catalog, the summary information is written into the log as how many records were in the input file and how many records loaded successfully.

The “Load CA” job step contains a single parameter named “`PARM_FILE`” [Parameter File]. To enable parallel processing of different blocks of records, the name of the file is dependent upon the CAPA ID supplied in the “Cost Allocation Process” step (that is, `CSAL_PROC_PARM_ID / Parameter Id`). The format for the name of the “Parameter File” is:
`$$$AMSPARM$$/CostAllocParam_CAPA<<CSAL_PROC_PARM_ID>>.txt”`.

For example, let’s assume that you are executing the Cost Allocation Process Chain Job, and you supply the value for the `CSAL_PROC_PARM_ID` parameter in Step 1 as *GF1A*. This means that the “`PARM_FILE`” value for the “Load CA” Step should be:
`$$$AMSPARM$$/CostAllocParam_CAPAGF1A.txt”`

When a run is executed with a different CAPA ID this value must be changed to reflect the new CAPA ID. Note that although a default value does exist for this parameter, it should always be updated to reflect the CAPA ID according to the syntax provided above.

When the CAPA record has a Run Cycle value of “Single”, logic for the creation of each downstream instance will ensure that the proper parameter values are passed from instance to instance, that is, parameter values established for the first instance, Offline Validation mode, will be passed to Base Accumulation mode and so forth through the Cost Allocation cycle.

The job cannot be restarted if it fails. If the Save Restart Information parameter is not selected or if the restart is not the immediate option, then the new job can be rescheduled but before rescheduling the job, the transactions loaded by the failed job should either be processed or discarded so that they do not remain in the catalog.

Major Input

- CA Transaction XML file

Batch Parameters

Parameter	Description	Default Value
Parameter file (PARM_FILE)	Parameter file to Load Transactions	\$\$AMSPARM\$\$/CostAllocParam.txt

Note: This PARM_FILE only contains the following subset of SMU parameters.

Parameter	Default Value
Action Code (ACTN_CD)	171
Commit Block Size (COMMIT_BLOCK)	1
File Name To Be Imported (FILE_NM)	\$\$AMSEXPORT\$\$/CAExport<<DateTi meStamp>>.xml
Restart Flag (RESTART_FL)	True
Statistics (STATS)	True
By Pass Auto Transaction Number (BYPS_ADNT_FL)	True

Please refer to the “SMU Transaction Upload Job” run sheet in the *CGI Advantage Financial – Utilities Run Sheets* guide for the full list of SMU Transaction upload batch parameters.

Major Output

- CA Transactions in draft version

Batch Return Codes

The following table shows the potential job return codes for the Load CA job.

Return Code	Condition
Successful (1)	All of the records are loaded into the Transaction Catalog successfully or the input file is empty.
Warning (4)	This return code will be issued when some of the records failed to load; however, other records were loaded successfully.
Non Fatal Error (8)	None of the records get loaded into the Transaction Catalog.
Failed (12)	<ul style="list-style-type: none"> • Parameters are invalid • When the input file is not found in the specified directory • Restart failed because another instance of the Cost Allocation Process has already been run successfully • Runtime exceptions encountered for any unexpected situations <p>When the job ends with a return code of Failed, subsequent jobs</p>

Return Code	Condition
	in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Sequence

N/A

Selection Criteria

N/A

Problem resolution

If the job ends with a return code of Failed and above, the job can be restarted only when the Save Restart Information parameter is selected and another instance of the job has not been scheduled and run successfully. If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – a new job should only be scheduled.

If the job cannot be restarted immediately, then the transactions loaded by this job should either be processed or discarded before rescheduling the new job. To process the loaded transactions, submit the Submit CA job in that chain. This job would be set to Inactive since the Load CA failed. The Submit job will submit the transactions that were generated during that Cost Allocation Process.

The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain. For general errors and recommendations, refer to the “SMU Transaction Upload Job” run sheet in the *CGI Advantage Financial – Utilities Run Sheets* guide.

Possible Return Codes	Condition	Recommendation	Other Instructions
Failed (12)	Failed while restarting the job since another instance of the job has already been run successfully. Sample Message: Cannot restart the job since another instance of this job has already been run successfully.	Recommendation: Schedule a new job.	

Cost Allocation Process Chain: Submit CA Job

Job Name	Submit CA
Recommended Frequency	Daily, monthly, quarterly or annual basis.
Single Instance Required	No. Inside of a chain it is Single Instance, however, there can be many chains running simultaneously.
Can be restarted?	Yes. See the “Overview” and the “Problem Resolution” sections for more details.
Report Generated	No. All of the exceptions are written to the error file.

Overview

This job submits the transactions listed in the input parameter file that was generated by the Cost Allocation Load CA job.

Major Input

- SMU job parameter file
- Draft Cost Allocation (CA) Transactions in the catalog

Batch Parameters

Parameter	Description	Default Value
Exception Report File Name (EXCEP_REP_FILE_NM)	Exception Report File Name	\$\$AMSLOGS\$\$/CostAllocExcp.txt
Submit Parameter File (SUBMIT_FILE)	Submit Parameter File. This file name must be the same as the file name for the same parameter in the Cost Allocation Process job step. The default value should be updated to include the CAPA ID. This is to ensure that different simultaneous runs will not be in conflict. For example, if the CAPA ID is 100, then the parameter should be set to CostAllocSubmit_CAPA100.txt. The Default Value should not be used, but the value can be quickly updated for each run.	\$\$AMSPARM\$\$/CostAllocSubmit.txt

Note: This job uses only a subset of the SMU submit job parameters. For a full list of available parameters for the SMU submit job, refer to the “SMU Transaction Submit Job” run sheet in the *CGI Advantage Financial – Utilities Run Sheets* guide.

Similar to the “Load CA” Step, the “Submit CA” Step has parameters that must be updated to reflect the CAPA ID specified in the “Cost Allocation Process” step. The two parameters in the “Submit CA” Step are “EXCEP_REP_FILE_NM” [Exception Report File Name] with a default value of “\$\$AMSLOGS\$\$/CostAllocExcp.txt”, and PARM_FILE [Submit Parameter File(.txt)] with

a default value of “\$\$AMSPARM\$\$/CostAllocSubmit.txt”. The default values must be updated to include the CAPA ID using the following syntax:

“\$\$AMSLOGS\$\$/CostAllocExcp_CAPA<<CSAL_PROC_PARM_ID >>.txt”

“\$\$AMSPARM\$\$/CostAllocSubmit_CAPA<<CSAL_PROC_PARM_ID >>.txt”

For example, if the CAPA ID is *GF1A*, the above two parameter values will need to be changed to:

“\$\$AMSLOGS\$\$/CostAllocExcp_CAPAGF1A.txt”

“\$\$AMSPARM\$\$/CostAllocSubmit_CAPAGF1A.txt”

When the CAPA record has a Run Cycle value of “Single”, logic for the creation of each downstream instance will ensure that the proper parameter values are passed from instance to instance, that is, parameter values established for the first instance, Offline Validation mode, will be passed to Base Accumulation mode and so forth through the Cost Allocation cycle.

Major Output

- The Transactions would have been processed to final or rejected.

Batch Return codes

The following table shows the potential job return codes for the Submit CA job in the Cost Allocation Process.

Return Code	Condition
Successful (1)	All of the transactions generated in that run submitted successfully.
Warning (4)	Not Applicable for this job.
Non Fatal Error (8)	Not Applicable for this job.
Failed (12)	<ul style="list-style-type: none"> • Input parameter file is not found. • Restart failed because another instance of the Cost Allocation Process has already been run successfully. • Runtime exceptions encountered for any unexpected situations. <p>When the job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.</p>
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Sequence

N/A

Selection Criteria

N/A

Problem resolution

If the job ends with a return code of Failed and above, the job can be restarted only when the Save Restart Information parameter is selected and another instance of the job has not been scheduled and run successfully. If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – a new job should only be scheduled.

If the restart is not an immediate option and the fatal error is because of a few transactions, the rest of the transactions can be submitted manually or discarded manually depending on the Issue. The Transaction Ids can be found on the input parameter file.

The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain. For general errors and recommendations, refer to the “SMU Transaction Submit Job” run sheet in the *CGI Advantage Financial – Utilities Run Sheets* guide.

Possible Return Codes	Condition	Recommendation	Other Instructions
Failed (12)	Failed while restarting the job since another instance of the job has already been run successfully. Sample Message: Cannot restart the job since another instance of this job has already been run successfully.	Schedule a new job.	

Cost Allocation Process Chain: CA Transaction Exception Report Job

Job Name	CA Transaction Exception Report
Recommended Frequency	Daily, monthly, quarterly or annual basis
Single Instance Required	No. Inside of the chain it is Single Instance, however, there can be many chains running simultaneously.
Can be restarted?	No
Report Generated	Yes. The job generates the reports in pdf and html formats.

Overview

This job in the Cost Allocation Process generates an exception report that lists all of the errors encountered when the CA transaction was submitted in the earlier step. The report contains the following information:

- Rejected CA transaction
- Detailed error description along with the error code

The following table shows the progression messages issued in this job.

Process Steps	Messages
<ul style="list-style-type: none"> • Parameter Validation 	<ul style="list-style-type: none"> • Validating Batch Parameters • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. It will be followed by message “Batch Parameter validation failed”. • Batch Parameters validation completed.
<ul style="list-style-type: none"> • Creating exception report 	<ul style="list-style-type: none"> • Generation of CA Exception Report started • Generation of CA Exception Report completed

Restartability Information

- Job cannot be restarted in this run mode. If the job fails in any of the above steps, a new job should be scheduled after correcting the errors that caused the job to fail.

Major Input

- Uses the Parm File generated during the Cost Allocation step to get the name of the SMU Exception File. It then walks through this file to find draft CA Transactions in the catalog with a Rejected status

Batch Parameters

Parameter	Description	Default Value
AMSLOGS	Logs Location at CA Transaction Exception Job (** Refer to Note: Assumptions for SWBP on page no. 5)	No default
AMSPARM	Parameter Location at CA Transaction Exception Report Job (** Refer to Note: Assumptions for SWBP on page no. 5)	No default

Major Output

- CA Transaction Exception Report if the current Run Mode on the CAPA Record is Generate Transaction or Reversal.
- When the CAPA record's Run Cycle is *Single*, the CAPA Record's Run Mode is updated for the next state, unless the current state is *Generate Transaction* or *Reversal*. If in a run mode short of *Generate Transaction* then a new instance of the chain is submitted.

Batch Return Codes

The following table shows the potential job return codes for the CA Transaction Exception Report job in the Cost Allocation Process.

Return Code	Condition
Successful (1)	All of the transactions generated in that run submitted successfully.
Warning (4)	Not Applicable for this job.
Non Fatal Error (8)	Not Applicable for this job.
Failed (12)	<ul style="list-style-type: none"> • Input file is blank. • Input parameter file is not found in the specified folder. • Restart failed because another instance of the Cost Allocation Process has already been run successfully. • Runtime exceptions encountered for any unexpected situations. <p>When the job ends with a return code of failed, subsequent jobs in the chain will be set to inactive.</p>
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Sequence

N/A

Selection Criteria

N/A

Problem resolution

The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain.

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Required Parameters are not entered Sample Message: Parameter file name is not specified.	Specify the Parameter File Name and schedule a new job.	Job should be rescheduled after correcting the parameters.
	Entered Parameters are not valid Sample Message: Parameter file should have file extension of .txt	Enter the correct Parameter file name and schedule a new job.	
	Parameter File(from which Exception file name is to be read) not found Sample Message: Parameter File(from which Exception file name is to be read) not found 'Parameter File Name'	Make sure that the Parameter File exists and schedule a new job.	
	Runtime Exception occurred. Sample Message: Error occurred during validateParam: <Runtime Exception Message>	The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Possible Return Codes	Condition	Recommendation	Other Instructions
System Failure (20)	When the job is terminated because of database server or network issues	The reason for the System Failure needs to be investigated. Schedule a new job.	Job should be rescheduled after correcting the problem.

Step 2: Creating exception report

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Runtime Exception occurred. Sample Message: Error occurred during genCAExcpRep: <Runtime Exception Message>	If the job fails with fatal conditions on encountering unknown exceptions, then investigate the exception reported by the process, resolve the error and schedule a new job.	System error log and VLS log should be investigated to find out possible reason of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	Job should be rescheduled after correcting the problem.

2.1.3 Cost Allocation Multi Process

Cost Allocation Multi Process provides an alternative to the standard Cost Allocation process. The Multi Process Import version of the Cost Allocation chain leverages a Multi Process Import step (named Load and Submit CA Transactions) instead of the separate Load CA and Submit CA steps, with the intention of increasing the overall efficiency of the process and thereby reducing runtimes.

For more information on the Cost Allocation Process and CA Transaction Exception Report job steps please refer to the run sheet for the standard Cost Allocation process in the previous section of this transaction. The following job step replaces the above referenced Load CA and Submit CA steps in the Cost Allocation Process chain. Much of the detail that follows for the Load and Submit CA Transactions job is also available in the runsheet for the Multi-Threaded Transaction Loader, which is located in the *CGI Advantage Financial – Utilities Run Sheet Guide*.

Cost Allocation Multi Process Chain: Load and Submit CA Transactions Job

Job Name	Load And Submit CA Transactions
Recommended Frequency	Daily, monthly, quarterly or annual basis.
Single Instance Required	No
Can the job be restarted?	Yes
Reports generated	Yes. If any of the child jobs caused an exception, it will be merged in the Exception Report file for the job (specified in the Job Parameter).

Overview

After the generation of the CA transaction xml file in the Cost Allocation Process job step, the next step of the Cost Allocation Multi Process Import chain is the Load and Submit CA Transactions step. This step employs the Multi-Threaded Transaction Loader utility to import and submit the CA transactions.

The Load and Submit CA Transactions job splits large Advantage Transaction files into smaller sized Transaction files. The application only accepts an XML file as input, and does the process of splitting the input file. Splitting is based on following input parameters:

1. The Block size – that is the number of Advantage Transaction blocks to be written to a split file before the write moves to the next set of files.
2. The Thread count – that is the total number of split files to be created.

For example: If an input file has 31 Advantage table records and the Block size is 5 and the thread count is 3, then 3 split files are created and the first 5 table records from the input file are written to the first split file, and the next 5 written to the next split file, and so on. Thus, when completed the first file would contain 11 table records, and the second and third files would contain 10 each (the first five records are written to file 1, the next five to file 2, the next five to file 3, the next five to file 1 again, and so on. As a result the final record is written to file 1 resulting in eleven records in file 1, and 10 records in files 2 and 3).

Major Input

The primary input is an XML file, containing CA transactions when the Run Mode is Generate Transaction or Reversal. It is important that the FILE_LIST parameter be correctly specified.

When the Cost Allocation Parameters (CAPA) ID has a Run Cycle value of *Multiple*, and a Run Mode of *Offline Validation*, *Base Accumulation* or *Compute Allocation*, no XML file is created by the Cost Allocation job step. However, an XML file must be listed in the Load and Submit CA Transactions job step, and that file must reside in the directory specified in the FILE_INPUT_DIR parameter. For that reason, an XML file (hereafter referred to as the default XML file) must be placed in that directory prior to running the Cost Allocation Multi Process Import chain. The specifications for that default file are as follows:

1. The file should be named as follows: CAExport_CAPA<CAPAID>.xml. For example, if the CAPA ID for the chain is ABCD, then the file should be named CAExport_CAPAABCD.xml
2. The content of the file should be exactly as follows:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
  <AMS_DOC_XML_EXPORT_FILE VERSION="1.0" EXPORT_DATE="2017-03-06
02:41:58.000000">
  </AMS_DOC_XML_EXPORT_FILE>
```

When running with a Run Cycle of *Multiple in Generate Transaction Mode* or *Reversal Mode*, the default XML file is not required to be placed on the server as the Cost Allocation Process step will create a file of the same name in the designated input directory, but will also contain the CA transactions rather than the standard data in the default file. If running the chain job again for the same CAPA ID again with a Run Cycle of *Multiple in Offline Validation*, *Base Accumulation* or *Compute Allocation* mode after an XML file has been created in *Generate Transaction* or *Reversal Mode*, then the default XML file should be placed back in the default directory, replacing any file created by the latter two run modes.

Note that if running the process with a Run Cycle of *Single*, much of the same logic applies. The default XML file should be placed in the input directory. Prior to running the chain job again for the same CAPA ID (with either Run Cycle of *Single* or *Multiple*), the default XML file should be placed back on the server as it would have been updated during *Generate Transaction* mode for the previous run.

Batch Parameters

Parameter	Description	Default Value
BLOCK_SIZE	(Required) Number of records in each split segment of the input file.	100
COMMIT_BLOCK_SIZE	(Required) Number of records to commit at a time.	100
ADDTL_RPT_FLDS	Optional. Include Additional reporting fields. Valid values are Yes or No.	No Default
FILE_INPUT_DIR	(Required) The location of the source files of records.	\$\$AMSROOT\$\$/Export Import
FILE_LIST	(Required) This parameter should always be set as follows: CAExport_CAPA<CAPAID>.xml. For example, if the CAPA ID for the chain is ABCD, then the file should be named CAExport_CAPAABCD.xml. The default name should not be used	CAExport.xml

Parameter	Description	Default Value
	and is only provided as the base for the correct filename.	
FILE_OUTPUT_DIR	(Required) Output location for the file segments.	\$\$AMSROOT\$\$/Export Import
FILE_PREFIX	(Optional) Prefix used on the filenames for the output file segments.	
Exclude Report (EXCL_COST_ALLOC_REP)	Optional. Enter numeric values separated by a comma to exclude the generation of the report by the Cost Allocation process. 1. Pool View 2. Base View 3. Expansion Exception 4. CA Transaction Exception	No Default
I_SMU_APPLY_OVERRIDES	(Required) If overrides are to be applied to transactions as they are loaded, this parameter should be true. A setting of false will result in transactions rejecting for override errors.	true
I_SMU_BYPASS_APPROVAL	(Required) SMU Import flag to indicate Bypass Approvals in Import Mode	false
I_SMU_BYPS_ADNT_FL	(Required) Bypass ADNT Flag	true
I_SMU_COMMIT_BLOCK	(Required) SMU Import to indicate Commit Block Size for SMU Jobs	100
I_SMU_DOC_STA_CD	(Required) SMU Import Transaction Status for Import Mode	2
I_SMU_OVERRIDE_LVL	(Optional) SMU Import Override Level for Import Mode	
I_SMU_RESTART_FL	(Required) SMU Import Utility flag to indicate Save Restart info in Import Mode	true
I_SMU_STATS	(Required) SMU Import to indicate if Statistics should be captured for SMU jobs	true
LOG_STATUS_INTERVAL	(Required) Logging frequency (in seconds) for controller thread reporting status of child threads to the system log.	300
MODE	(Required) Mode (2=Import and Submit, Please set no other)	2
SLEEP_INTERVAL	(Required) Polling frequency (in seconds) for internal controller	5

Parameter	Description	Default Value
	thread for checking child processes.	
SMU_CTLG_ID	(Required) Catalog id of the System Maintenance Utility job which is spawned as the child process.	3
STAGGER_TIME	(Required) The lag time, in seconds, between the spawning of each child process.	30
S_SMU_APPLY_OVERRIDE	(Required) SMU Submit flag to indicate Apply Overrides in Submit Mode	true
S_SMU_BYPASS_APPROVAL	(Required) SMU Submit flag to indicate Bypass Approvals in Submit Mode	false
S_SMU_BYPS_ADNT_FL	(Required) SMU Submit to indicate if Bypass of Auto Transaction Numbering should be performed	true
S_SMU_COMMIT_BLOCK	(Required) SMU Submit to indicate Commit Block Size for SMU Jobs	100
S_SMU_EXCEP_REP_FILE_NM	(Required) The default value must be updated to include the CAPA ID using the following syntax: “\$AMSLOGS\$/CostAllocExcp_CAPA<<CSAL_PROC_PARM_ID >>.txt” For example, if the CAPA ID is <i>GF1A</i> , the parameter value will need to be changed to: “\$AMSLOGS\$/CostAllocExcp_CAPAGF1A.txt”	CostAllocExcp.txt
S_SMU_EXCEP_REP_IND	(Required) SMU Submit Exception Report Indicator for Submit Mode	4
S_SMU_EXP_SEV_FL	(Required) SMU Submit flag to indicate Severity Flag in Submit Mode	1
S_SMU_OVERRIDE_LVL	(Optional) SMU Submit Override Level for Submit Mode	
S_SMU_RESTART_FL	(Required) SMU Submit to indicate Save Restart info in Submit Mode	true
S_SMU_STATS	(Required) SMU Submit to indicate if Statistics should be captured for SMU jobs	true
THREAD_COUNT	(Required) Number of jobs to start	1

Please refer to the Multi-Threaded Transaction Loader run sheet in the *CGI Advantage Financial – Utilities Run Sheet Guide* for the full list of batch parameters.

Major Output

- Submitted CA transactions

Job Return Codes

If any of the child jobs were not successful, the greatest Job Return code among the child jobs will be returned. For the Job Return codes for the child jobs, refer to the Job Return Code section of the System Maintenance Utility job.

Return Code	Condition
Successful (1)	All of the Transaction records are imported successfully.
Warning (4)	N/A
Non-Fatal (8)	This job will issue a non-fatal error for the following: <ul style="list-style-type: none"> ▪ Error while importing records that already exist in the application.
Failed (12)	The job will fail under the following conditions: <ul style="list-style-type: none"> • Invalid Parameters • Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

There is not any sort criteria.

Selection Criteria

The input record files (specified in the Job Parameter) are selected.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	N/A	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Non-Fatal (8)	<p>Job Failed due to Non-fatal condition:</p> <p>Duplicate record being imported to application.</p>	<p>No action is needed. The job will continue and will import all other records (assuming they do not exist within the application).</p>	N/A
Failed (12)	<p>Job failed due to Fatal conditions.</p> <p>Sample Message: "Invalid value received for parameter MODE (expected values 1,2 or 3): aBCD"</p> <p>"No input file to process"</p>	<p>In this step, the job can fail under the following conditions:</p> <ol style="list-style-type: none"> 1) Job Parameter error 2) Encounters any runtime exceptions and 3) Failed during child job. <p>If the Job Parameter input was causing the error, examine the error message in the job log, correct the job parameter, then restart the job.</p> <p>If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.</p> <p>For child job errors, refer to the runsheet for the System Maintenance Utility job.</p>	<p>If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.</p>
Terminated (16)	<p>Job is terminated manually by the user.</p>	<p>The reason for the termination needs to be investigated before scheduling a new job. The job cannot be restarted.</p>	<p>If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.</p>
System Failure (20)	<p>When the job is terminated because of database server or network issues.</p>	<p>The reason for the System Failure needs to be investigated before scheduling a new job. The job cannot be restarted.</p>	<p>If another instance of the job has already been scheduled and ran successfully, then this job should not be</p>

Possible Return Codes	Condition	Recommendation	Other Instructions
			restarted – only a new job should be scheduled.

2.1.4 Cost Allocation with Supplemental Reporting

Chain Job Name	Cost Allocation with Supplemental Reporting (Supp. Rpt.)
Recommended Frequency	Daily, monthly, quarterly, or annual basis.
Single Instance Required	No
Can be restarted?	No
Reports generated	No

Overview

Cost Allocation with Supplemental Reporting capabilities provides an alternative to the standard Cost Allocation process. This job will spawn the Cost Allocation Process chain and wait for the chain execution to finish. Once the allocation is completed, this job will copy records from a set of cost allocation tables to a set of database tables created for reporting and audit purposes. This 'preserved' data is both setup and output. These also store additional information from the allocation like Cost Allocation ID, Date/Timestamp, and Batch ID.

As this job is responsible for executing the steps commonly found in the Cost Allocation Process chain, it will capture the same parameters.

Processing Logic

The process will perform the following steps:

1. Validate the batch parameters
2. Initiate and submit the Cost Allocation Process chain
3. Wait for the chain job to finish successful execution
4. Copy all records from the 10 tables listed in the Major Input section to the corresponding database tables in the Major Output section along with additional fields
5. Update the CST_ALLOC_LAST_RUN_DT record on Application Parameters (APPCTRL) with the Application Date (APPL_SYS_DT)

Below are the job log messages provided:

Process Steps	Messages
4. Parameter Validation	<ul style="list-style-type: none"> Validating Batch Parameters Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. Batch Parameter validation completed
5. Spawn Cost Allocation Chain	<ul style="list-style-type: none"> Starting Cost Allocation Chain job Enabled Chain Job <Job ID> Parameters assigned to job Enabled Job <Job ID> Cost Allocation Chain job submitted with chain ID <Job ID> Chain job running

Process Steps	Messages
	<ul style="list-style-type: none"> Chain Job <Job ID> completed and job return code is <Return code>.
6. Selection and Copying of Records	<ul style="list-style-type: none"> Started copying records from Cost Allocation Tables Copied <i>n</i> records from <Source database table> to <Target database table>. Please refer to the Major Input and Output sections for the details on 10 tables. For each table the job log will have individual entry. Data Copy ended Updating CST_ALLOC_LAST_RUN_DT Parm with SYSDATE on IN_APP_CTRL table Updated CST_ALLOC_LAST_RUN_DT on IN_APP_CTRL table

Major Input

- Cost Allocation Control Setup (ALOC) -- R_CSAL_CTRL_SETUP
- Cost Allocation Series Setup (SRS) -- R_CSAL_SRCTR_SETUP
- Cost Allocation Step Setup (STEP) -- SR_CSAL_STPCTR_SETUP
- Pool/Base Offset Requirement (PBOREQ) -- R_PLBS_OFST_REQ
- Statistical Unit (STAT) -- R_STAT_UNIT_SETUP
- Pool/Base Setup (PLBS) -- R_PLBS_SETUP
- Pool/Base Distribution (PBDIST) -- R_PLBS_DIST
- Cost Allocation Journal Summary Inquiry (CAJR) -- CSAL_JRNL_SUM
- Cost Allocation Totals Inquiry (CTOT) -- CSAL_TOT
- Cost Allocation Process Parameters (CAPA) – CSAL_PROC_PARM

Note: The default values listed are those delivered with the software. Actual values may vary based on your site’s setup.

Parameter	Description	Default Value
(Optional) Include Additional reporting fields. Valid values are Yes or No (ADDTL_RPT_FLDS)	Optional. Include Additional reporting fields. Valid values are Yes or No.	No Default
Export Location at Cost Allocation Process Job (AMSEXPORT)	Required file location to write the XML output.	\$\$AMSRROOT\$\$/ExportImport
Logs Location at Cost Allocation Process Job (AMSLOGS)	Required file location to write log files.	\$\$AMSRROOT\$\$/Logs

Importing Location at Cost Allocation Process Job (AMSPARM)	Required location to write the parameter file passed to subsequent job steps.	\$\$AMSR00T\$\$/Parms
Apply Overrides (APPLY_OVERRIDES)	Required indication if overrides are applied to transactions. If left blank, this parameter defaults to <i>false</i> .	True
Override Level (OVERRIDE_LVL)	Optional level of override applied when the Apply Overrides parameter is <i>true</i> . If set as 0 or left blank and the Apply Overrides parameter is <i>true</i> then the level of the User ID executing the chain job will be used.	(blank)
Client name for Report (CLIENT_NM)	Optional name to appear on report headers.	(blank)
Parameter ID	Required identification of a Parameter ID from the Cost Allocation Process Parameter page to control the allocation.	(blank)
Exclude Report (EXCL_COST_ALLOC_REP)	Optional parameter to turn off report generation 1. Pool View 2. Base View 3. Expansion Exception 4. CA Transaction Exception	(blank)
Purge Previous Run CAPA ID (PURGE_PREV_RUN_CAPA_ID)	Optional parameter of a Parameter ID from a previous run for purging data from the following tables: CSAL_JRNL_SUM, CSAL_TOT, CSAL_EXP_N_RSLT, CSAL_HIST.	(blank)
(NUMBER_OF_LOOPS)	Required number of sleep intervals to wait until child jobs finish. A negative number will wait until all child jobs have completed.	-1
(SLEEP_INTERVAL)	Required time to wait before checking child job progress (in milliseconds)	15000

Major Output

In addition to the tables that the Cost Allocation Chain updates, this job will additionally update the following tables:

- R_CSAL_CTRL_SETP_WIP
- R_CSAL_SRCTR_SETP_WIP

- R_CSAL_STPCTR_SETP_WIP
- R_PLBS_OFST_REQ_WIP
- R_STAT_UNIT_SETP_WIP
- R_PLBS_SETP_WIP
- R_PLBS_DIST_WIP
- CSAL_JRNL_SUM_WIP
- CSAL_TOT_WIP
- CSAL_PROC_PARM_WIP

Note: The Cost Allocation Chain can be run in two Run Cycles, *Single* or *Multiple*. When run with the cycle of *Single*, this job spawns 4 instances of the Cost Allocation Chain (one for each mode: Offline Validation, Base Accumulation, Compute Allocation and Generate Transaction). After each of these modes finishes successfully, the records from 10 source tables will be copied with Job ID and other additional details. When the cycle is *Multiple* with any of the four modes, only one instance of the chain is spawned and records are copied over for that single run.

If separate data sets are not needed for Offline Validation, Base Accumulation and Compute Allocation, when using the *Multiple* cycle for allocations, use Cost Allocation with Supp Rpt. for Generate Transaction mode only, since this mode indicates the successful completion of cost allocation chain and creation of cost allocation transactions.

Chain / Job Return Code

The following table shows the potential job return codes for the Cost Allocation Process job.

Return Code	Condition
Successful (1)	This Return Code is issued when the Cost Allocation chain ends successfully and Table Copy process completes successfully.
Warning (4)	This job does not issue this Return Code.
Non Fatal Error (8)	This job does not issue this Return Code.
Failed (12)	The Job will fail under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Cost Allocation chain returns a failure.
Terminated (16)	This Return Code is issued when the job is terminated by the user.
System Failure (20)	This Return Code is issued when the job is terminated because of database server or network issues.

Sort Criteria

Order retained from source table

Selection Criteria

Cost Allocation Process Parameter ID and Cost Allocation Process Chain ID

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	<p>Required Parameters are not entered.</p> <p>Sample Message: Cost Allocation Parameter ID cannot be left blank.</p>	Enter the Cost Allocation Parameter ID and schedule a new job.	Job should be rescheduled after correcting the parameters.
	Cost Allocation Process failed to complete successfully	Recommendation: Review the logs from that process and make corrections as necessary according to that run sheet	Schedule a new instance of the Cost Allocation with Supp. Rpt.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and run successfully, then this job should not be restarted – only a new job should be scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and run successfully, then this job should not be restarted – only a new job should be scheduled.

2.1.5 Cost Allocation Multi Process with Supp. Rpt.

Job Name	Cost Allocation Multi Process with Supp. Rpt.
Recommended Frequency	Daily, monthly, quarterly or annual basis.
Single Instance Required	No
Can be restarted?	No
Reports generated	No

Overview

The Cost Allocation Multi Process with Supplemental Reporting job provides an alternative to the Cost Allocation Multi Process. This job will spawn the Cost Allocation Multi Process chain and wait for the chain execution to finish. Once the Cost Allocation Multi Process chain is completed this job will copy records from a pre-defined set of cost allocation tables to the corresponding database tables created for reporting. Data that is 'preserved' is both setup data and processed records. The database tables will store additional information such as, Cost Allocation ID, Date/Timestamp, and Batch ID. The copied data in database tables can be used for reporting and audit purposes.

As this job is responsible for executing the steps commonly found in the Cost Allocation Multi Process chain job, it will capture the same parameters as the other chain.

Processing Logic

The process will perform the following steps:

1. Validate the batch parameters.
2. Initiate and submit the Cost Allocation Multi Process chain
3. Wait for the chain job to finish successful execution.
4. Copy all records from the 10 tables given in the Major Input section to the corresponding database tables given in the Major Output section along with additional fields.
5. Update the Application Parameter (APPCTRL) of CST_ALLOC_LAST_RUN_DT with the system date from APPCTRL.

Below are the job log messages provided:

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> • Validating Batch Parameters • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. • Batch Parameter validation completed
2. Spawn Cost Allocation Chain	<ul style="list-style-type: none"> • Starting Cost Allocation Multi Process Chain job • Enabled Chain Job <job ID>

Process Steps	Messages
job	<ul style="list-style-type: none"> Parameters assigned to job Enabled Job <job ID> Cost Allocation Multi Process Chain job submitted with chain ID <Job ID> Chain job running Chain Job <Job ID> completed and job return code is <Return code>.
3. Selection and copying of Records	<ul style="list-style-type: none"> Started copying records from Cost Allocation Tables Copied <i>n</i> records from <Source database table> to <Target database table>. Please refer to the Major Input and Output sections for the details on 10 tables. For each table the job log will have an individual entry. Data Copy ended Updating CST_ALLOC_LST_RUN_DT Parm with SYSDATE on IN_APP_CTRL table Updated CST_ALLOC_LST_RUN_DT on IN_APP_CTRL table

Major Input

- Cost Allocation Control Setup (ALOC) -- R_CSAL_CTRL_SETUP
- Cost Allocation Series Setup (SRS) -- R_CSAL_SRCTR_SETUP
- Cost Allocation Step Setup (STEP) -- SR_CSAL_STPCTR_SETUP
- Pool/Base Offset Requirement (PBOREQ) -- R_PLBS_OFST_REQ
- Statistical Unit (STAT) -- R_STAT_UNIT_SETUP
- Pool/Base Setup (PLBS) -- R_PLBS_SETUP
- Pool/Base Distribution (PBDIST) -- R_PLBS_DIST
- Cost Allocation Journal Summary Inquiry (CAJR) -- CSAL_JRNL_SUM
- Cost Allocation Totals Inquiry (CTOT) -- CSAL_TOT
- Cost Allocation Process Parameters (CAPA) – CSAL_PROC_PARM

Parameters

Note: The default values listed are those delivered with the software. Actual values may vary based on your site’s setup.

Parameter	Description	Default Value
Additional Reporting Fields (ADDTL_RPT_FLDS)	An optional output parameter that when Yes will result in additional report information. If left blank, No defaults.	No
Export Location (AMSEXPORT)	A required file location to write the XML output.	\$\$AMSROOT\$\$/ExportImport
Logs Location	A required file location to write log files for	\$\$AMSROOT\$\$/

(AMSLOGS)	exception reporting.	Logs
Parameter File Location (AMSPARM)	A required file location to write a parameter file for later job steps.	\$\$AMSROOT\$\$/ Parms
Apply Overrides (APPLY_OVERRIDES)	A required processing parameter, that when <i>true</i> indicates overrides are to be applied to generated transactions. If left blank, this parameter defaults to <i>false</i> .	True
Override Level (OVERRIDE_LVL)	An optional processing parameter of an override level applied when the Apply Overrides parameter is <i>true</i> . If set as 0 or left blank and the Apply Overrides parameter is <i>true</i> , then the override level of the User ID executing the chain job will be used.	(blank)
Client Name (CLIENT_NM)	An optional output parameter of a name that will appear on report headers.	(blank)
Parameter ID (CSAL_PARM_ID)	The required identification of a Parameter ID from the Cost Allocation Process Parameter page to control the allocation.	(blank)
Reschedule Job ID (RSCHED_JOB_ID)	An optional parameter to be provided when rescheduling for a failed run. The parameter indicates the Job ID to be entered of the job that previously failed as <i>Terminated</i> or <i>System Failure</i> , which now needs to be completed.	(blank)

Major Output

In addition to the tables that the Cost Allocation Multi Process Chain updates, this job will additionally update the following tables:

- R_CSAL_CTRL_SETP_WIP
- R_CSAL_SRCTR_SETP_WIP
- R_CSAL_STPCTR_SETP_WIP
- R_PLBS_OFST_REQ_WIP
- R_STAT_UNIT_SETP_WIP
- R_PLBS_SETP_WIP
- R_PLBS_DIST_WIP
- CSAL_JRNL_SUM_WIP
- CSAL_TOT_WIP
- CSAL_PROC_PARM_WIP

Note: The Cost Allocation chain can be run in two Run Cycles, *Single* or *Multiple*. When run with the cycle of *Single*, this job spawns 4 instances of the Cost Allocation Chain (one for each mode: Offline Validation, Base Accumulation, Compute Allocation and Generate Transaction). After each of these modes finishes successfully, the records from 10 source tables will be copied with

Job ID and other additional details. When the cycle is *Multiple* with any of the four modes, only one instance of the chain is spawned and records are copied over for that single run.

If separate data sets are not needed for Offline Validation, Base Accumulation, and Compute Allocation, when using the *Multiple* cycle for allocations, use Cost Allocation with Supp Rpt. for Generate Transaction mode only, since this mode indicates the successful completion of the cost allocation chain and creation of cost allocation transactions.

Job Return Code

The following table shows the potential job return codes for the Cost Allocation Multi Process with Supplemental Reporting job.

Return Code	Condition
Successful (1)	This Return Code is issued when the Cost Allocation Multi Process chain ends successfully and the table copy process completes successfully.
Warning (4)	This job does not issue this Return Code.
Non Fatal Error (8)	This job does not issue this Return Code.
Failed (12)	The Job will fail under the following conditions: <ul style="list-style-type: none"> Parameters are invalid. Cost Allocation Multi Process chain returns a failure.
Terminated (16)	This Return Code is issued when the job is terminated by the user.
System Failure (20)	This Return Code is issued when the job is terminated because of database server or network issues.

Sort Criteria

N/A

Selection Criteria

N/A

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Failed (12)	<p>Required Parameters are not entered.</p> <p>Sample Message: Cost Allocation Parameter ID cannot be left blank.</p>	<p>Enter the Cost Allocation Parameter ID and schedule a new job.</p>	<p>Job should be rescheduled after correcting the parameters.</p>
	<p>Cost Allocation Process failed to complete successfully</p>	<p>Recommendation: Review the logs from that process and make corrections as necessary according to that run sheet</p>	<p>Schedule a new instance of the Cost Allocation with Supp. Rpt. job.</p>
Terminated (16)	<p>Job is terminated manually by the user.</p>	<p>The reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.</p>	<p>If another instance of the job has already been scheduled and run successfully, then this job should not be restarted – only a new job should be scheduled.</p>
System Failure (20)	<p>When the job is terminated because of database server or network issues.</p>	<p>The reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.</p>	<p>If another instance of the job has already been scheduled and run successfully, then this job should not be restarted – only a new job should be scheduled.</p>

2.1.6 Cost Allocation Standalone with Supp. Rpt.

Job Name	Cost Allocation Standalone with Supp. Rpt.
Recommended Frequency	On-Demand
Single Instance Required	No
Can be restarted?	No
Reports generated	No

Overview

The Cost Allocation Standalone with Supplemental Reporting job provides the ability to copy records from a pre-defined set of cost allocation tables to corresponding database tables created for reporting. Data that is 'preserved' is both setup data and processed records. The database tables will store additional information like Cost Allocation ID, Date/Timestamp and Batch ID. The copied data in database tables can be used for reporting and audit purposes.

The intention of this job is to record the historical information in the event the Cost Allocation Chain with Supplemental Reporting or Cost Allocation Multi Process Chain with Supplemental Reporting was not used to initiate the Cost Allocation Process.

Processing Logic

The process will perform the following steps:

1. Validate the batch parameters.
2. Copy all records from the 10 tables given in the Major Input section to the corresponding database tables given in the Major Output section along with additional fields.

Below are the job log messages provided:

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> • Validating Batch Parameters • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. • Batch Parameter validation completed
2. Selection and copying of Records	<ul style="list-style-type: none"> • Started copying records from Cost Allocation Tables • Copied <i>n</i> records from <Source database table> to <Target database table>. Please refer to the Major Input and Output sections for the details on 10 tables. For each table the job log will have individual entry. • Data Copy ended

Major Input

- Cost Allocation Control Setup (ALOC) -- R_CSAL_CTRL_SETP

- Cost Allocation Series Setup (SRS) -- R_CSAL_SRCTR_SETP
- Cost Allocation Step Setup (STEP) -- SR_CSAL_STPCTR_SETP
- Pool/Base Offset Requirement (PBOREQ) -- R_PLBS_OFST_REQ
- Statistical Unit (STAT) -- R_STAT_UNIT_SETP
- Pool/Base Setup (PLBS) -- R_PLBS_SETP
- Pool/Base Distribution (PBDIST) -- R_PLBS_DIST
- Cost Allocation Journal Summary Inquiry (CAJR) -- CSAL_JRNL_SUM
- Cost Allocation Totals Inquiry (CTOT) -- CSAL_TOT
- Cost Allocation Process Parameters (CAPA) – CSAL_PROC_PARM

Parameters

Note: The default values listed are those delivered with the software. Actual values may vary based on your site's setup.

Parameter	Description	Default Value
Chain Job ID from Cost Allocation Run (CHAIN_JOB_ID)	Required identification of a chain job ID from Cost Allocation Process or Multi Process to identify the chain for which data is being backed up.	(blank)
Parameter ID (CSAL_PARM_ID)	Required identification of a Parameter ID from the Cost Allocation Process Parameter page to control the allocation.	(blank)
Cost Allocation Run Date (RUN_DT)	Required field to capture date when Cost Allocation Process was run. Enter as DD-MM-YYYY.	(blank)

Major Output

This job will update the following tables

- R_CSAL_CTRL_SETP_WIP
- R_CSAL_SRCTR_SETP_WIP
- R_CSAL_STPCTR_SETP_WIP
- R_PLBS_OFST_REQ_WIP
- R_STAT_UNIT_SETP_WIP
- R_PLBS_SETP_WIP
- R_PLBS_DIST_WIP
- CSAL_JRNL_SUM_WIP
- CSAL_TOT_WIP
- CSAL_PROC_PARM_WIP

Job Return Code

The following table shows the potential job return codes for the Cost Allocation Standalone with Supp. Rpt. job.

Return Code	Condition
Successful (1)	This Return Code is issued when the job completes successfully.
Warning (4)	This job does not issue this Return Code.
Non Fatal Error (8)	This job does not issue this Return Code.
Failed (12)	The Job will fail under the following conditions: <ul style="list-style-type: none"> Parameters are invalid.
Terminated (16)	This Return Code is issued when the job is terminated by the user.
System Failure (20)	This Return Code is issued when the job is terminated because of database server or network issues.

Sort Criteria

N/A

Selection Criteria

N/A

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Required Parameters are not entered. Sample Message: Cost Allocation Parameter ID cannot be left blank.	Enter the Cost Allocation Parameter ID and schedule a new job.	Job should be rescheduled after correcting the parameters.
	Entered Parameters are not valid.	Recommendation: Make sure that the CSAL_PROC_PARM	Job should be rescheduled after correcting the

Possible Return Codes	Condition	Recommendation	Other Instructions
	Sample Message: Parameter ID record not found.	record exists.	parameters.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and run successfully, then this job should not be restarted – only a new job should be scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and run successfully, then this job should not be restarted – only a new job should be scheduled.

2.1.7 Credit Memo Multi Thread Chain

Chain or Job Name	Credit Memo Multi Thread Chain
Recommended Frequency	On Demand. Preferably before the Automated Disbursement (AD) Chain is run.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes

Overview

When billing for grant reimbursements using accrued and cash expenditures, instead of just cash expenditures (which is most common), a credit memo entered against a grant reduces the draw amount in the grant fund. A central fund often has to make up that difference until the credit memo is applied through a disbursement. Furthermore, as a complex funding structure progresses, the COA values for fund can change at the posting line so that what was used on the credit memo is different from where the cash expenditure is eventually posted.

The Credit Memo Multi Thread Chain process identifies the credit memo transactions and generates a Journal Voucher that transfers funds from the central fund to the Federal grant fund to cover the difference between the draw amount and the disbursement amount without changing the overall Program/Phase balances. The process then reverses these adjustments by generating another Journal Voucher when the credit memo is adjusted as part of subsequent disbursement to the vendor through the AD Chain.

For each credit memo found, the process generates a transaction that does the following with a memo reference to the credit memo transaction on each accounting line (example uses default posting codes):

Federal Fund to reverse the credit memo

Debit D011
Credit D001

Federal Fund to increase cash for draw deficit

Debit A001
Credit D014

Central Fund to record the credit memo

Debit D001
Credit D011

Central Fund to decrease cash for draw deficit

Debit D014
Credit A001

For each applied credit memo fund, the process generates a transaction that reverses the above with a memo reference to the disbursement transaction on each accounting line. The same postings result from the cancellation of a credit memo transaction, but the memo reference is to the credit memo transaction.

Implementation Considerations

- The federal and central accounting distribution are captured on the respective funding lines under the Front-End-Split COA section.

- Even for a 100% Federally funded project/grant a central line with 0% is present to capture the central accounting distribution.
- It is recommended to isolate out the credit memo and adjustment transactions by creating specific credit memo transaction codes when not all grants reimburse off of accrued expenses. Credit memos are restricted to just a single transaction code in the Accounting Based Spending (ABS) transaction type for this process.
- If implementing this process where accounting activity has already been recorded, to prevent the process from reading the input journal from the beginning, a record should be inserted into the Journal Log as follows:

Process ID: CMAID

Run Date: Today (mm/dd/yyyy hh:mm)

Journal/Ledger Name: (JRNL_CA or JRNL_ACTG – the one used)

Description: Initial CMAID record

Source Journal/Ledger ID: 2 for JRNL_CA or 1 for JRNL_ACTG

Run Type: Initial (1)

Status: Final (2)

Begin Source Journal Record Number: 1

End Source Journal Record Number: Journal record number (REC_NO) from where processing should start

The Credit Memo Multi Thread Chain has the following jobs in the chain.

- [JV Export](#)
- [JV Multithreaded Transaction Loader](#)
- [JV Report](#)

Major Input

- Accounting or Cost Accounting Journal (JACTG / JRNL_ACTG or JCA / JRNL_CA)
- Journal Log (JLOG / JRNL_LOG)
- Funding Profile (FNDFLST / R_FPRFL)
- Vendor/Customer (VCUST / R_VEND_CUST)

Major Output

- Journal Voucher transaction XML (JVExport.xml)
- Journal Log (JLOG / JRNL_LOG)
- Credit Memo Processed and Credit Memo Exception Reports

Chain Return Code

The following table shows the potential return codes for the Credit Memo Multi Thread Chain. Note that the Chain job ends with the highest return code across all of the jobs.

Return Code	Condition
Successful (1)	All jobs ended as successful.
Warning (4)	One of the jobs in the chain ended as warning.
Non-Fatal Error (8)	One of the jobs in the chain ended as non-fatal error.
Failed (12)	One of the jobs in the chain ended as failed.
Terminated (16)	One of the jobs in the chain ended as terminated.
System Failure (20)	One of the jobs in the chain ended as system failure.

Problem Resolution

Please refer to the individual jobs for details regarding the specific job processes and problem resolution.

Credit Memo Multi Thread Chain - JV Export

Job Name	JV Export
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	Yes
Reports Generated	No

Overview

This job selects eligible credit memo records meeting selection criteria from the input journal, creates the initial or reversal journal voucher transactions into an XML file for a later step to load and submit.

The following table shows the various steps that the Record Selection process goes through and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	<p>The process validates the Batch Parameters. And logs the following messages for invalid parameters:</p> <ul style="list-style-type: none"> • Run Started • Each parameter is listed • Any parameter errors are listed
2. Selection of Credit Memo transactions	<p>The process selects from the source journal specified the incremental transactions since its previous run using the selection transaction code.</p> <ul style="list-style-type: none"> • “CMA Process started record processing records from record ID: #####” • Each credit memo and credit memo transaction found is listed with the journal voucher created.

Process Steps	Messages
	<ul style="list-style-type: none"> • “Max Record ID from <Journal>: #####” • “JV Export Completed: Last processed record is: #####” <p>When no new journal records are found:</p> <ul style="list-style-type: none"> • “Max Record Id from Jrnl Actg: 0” <p>When no credit memo or credit memo liquidation records are found:</p> <ul style="list-style-type: none"> • “No records for GAX processed”
3. Identifying Federal and State lines	<p>The process then identifies the Federal and Central lines by using the funding line set up and then identifies the federal lines using Cost Accounting Funding Type on Vendor Customer (VCUST). The funding types of 1-Federal (Non-CMIA), 2-Federal (CMIA), 10-FHWA (CMIA) and 11-FHWA (Non-CMIA) are considered federal. Records with other funding types are considered as Central.</p> <p>The process then retrieves the Federal and Central accounting distributions from the Front-End-Split section of Funding Lines.</p>
4. Creating JV.XML	<p>The process then starts writing the records to JV.XML using the transaction identifier information from the batch parameter.</p>

Major Input

- Accounting or Cost Accounting Journal (JACTG / JRNL_ACTG or JCA / JRNL_CA)
- Journal Log (JLOG / JRNL_LOG) for Process ID *CMAID*
- Funding Profile (FNDFLST / R_FPRFL)
- Vendor/Customer (VCUST / R_VEND_CUST)

Batch Parameters

The parameters shown below are with delivered values and what is implemented may vary. Once this process is run, the selection parameters should not change for Input Source, Selection Transaction Code, and Selection Department.

Parameter	Description	Default Value
Accrued Expense Posting Code (ACCUR_EXPNS)	A required selection parameter identifying the posting code for Accrued Expense. The parameter is also used as an output parameter.	D011
Action Code (ACTION_CODE)	A required and protected parameter used for system processing.	EXPORT
Export Location (AMSEXPORT)	The required location for which the XML file is written.	\$\$AMSROOT\$\$/ExportImport
Accounting Period (APD)	An optional output parameter for an accounting period on the generated transactions instead of letting a value default.	No default
Budget Fiscal Year (BFY)	An optional output parameter for a budget year on the generated transactions instead of letting a value	No default

	default.	
Cash Expense Posting Code (CASH_EXPNS)	A required selection parameter identifying the posting code for Cash Expense. The parameter is also used as an output parameter.	D014
Cash Posting Code (CASH_PSTNG)	A required output parameter identifying a cash accounting line.	A001
Disbursement Payable Posting Code (DISBRS_PAYBL)	A required output parameter identifying a payable accounting line.	D001
Fiscal Year (FY)	An optional output parameter for a fiscal year on the generated transactions instead of letting a value default.	No default
Output Transaction Code (GEN_DOC_CD)	A required output parameter of a transaction code in the Journal Voucher transaction type.	JVC
Output Transaction Department (GEN_DOC_DEPT_CD)	A required output parameter of a department code used for ADNT lookup and security for the generated transactions.	No default
Output Transaction Prefix (GEN_DOC_PFX)	An optional output parameter of a prefix used in the generation of transaction IDs, as defined on ADNT.	No default
Output Transaction Unit (GEN_DOC_UNIT_CD)	An optional output parameter of a unit that is used to secure the generated transactions within a department.	No default
XML File Name (GEN_EXPORT_FILE)	A required output parameter of the XML file name created with journal vouchers.	JVExport.xml
Processed File GEN_PROCESS_FILE	A required output parameter of the TXT file name created with each transaction created for later reporting.	JVProcessing.txt
Progress Counter (PROG_CTR_SZ)	A required processing parameter that defines a counter to be used in the progression messages issued for journal record processing. If left blank, 100 will default.	100
Select Transaction Code (SEL_DOC_CD)	A required selection parameter to identify a single transaction code that may contain credit memo events.	GAX
Selection Transaction Department SEL_DOC_DEPT	An optional selection parameter used to further filter the credit memo events when Selection Transaction ID is entered.	****
Selection Transaction ID (SEL_DOC_ID)	An optional selection parameter used to select an individual credit memo transaction for processing.	No default
Selection Transaction Version SEL_DOC_ID_VERSION	A conditionally required selection parameter used to select an individual credit memo transaction for processing when Selection Transaction ID is used. Enter a specific version or use the asterix (*) to indicate all versions.	*

Source Journal (SOURCE_JOURNAL)	A required input parameter that identifies either the Accounting Journal (JRNL_ACTG) or Cost Accounting (JRNL_CA) that is input into the process.	JRNL_ACTG
------------------------------------	---	-----------

Major Output

- JV Transaction XML file

Selection Criteria

The eligible credit memo transactions are selected from the Source Journal using the Selection Transaction Code unless the Selection Transaction Department, Transaction ID and Version are entered so that a single transaction is processed. The process only selects incremental transactions since the previous run by using Journal Log entries. The credit memo transactions with at least one Federal posting line is considered for selection.

Please note that when selecting a single transaction, a backup of JRNL_LOG is recommended before that selection that is restored after one or more single transactions are selected. An alternative to select a single transaction is to manually process a journal voucher if the credit memo is still outstanding.

Sort Sequence

Records are processed in the order they are found in the input journal.

Problem Resolution

If the job ends with a return code of Failed and above, the job can be restarted when another instance of the job has not been scheduled and run successfully. If another instance of the job has already been scheduled and run successfully, then this job should not be restarted, a new job should only be scheduled.

The following table shows the possible return codes and recommendations for the JV Export job step across all the processing steps in the job.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Records were found for processing and successfully processed	N/A	N/A
Warning (4)	No new journal records were found or some records were found but none matched selection criteria.	Manually verify if desired; otherwise, wait until the next run.	N/A
Non-Fatal Error (8)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for 	Review the condition for failure and make corrections when possible before submitting a new chain.	N/A

	<p>unexpected situations.</p> <p>When this job ends with a return code of Failed, subsequent jobs in the chain are set to inactive.</p>		
Failed (12)	<p>Failed while validating the parameter.</p> <p>Sample Message: Invalid Parameters (As specified in the Progressive log messages).</p> <p>Failed with a runtime Exception.</p>	<p>The job can be rescheduled after correcting the parameters. Alternatively, a new chain can be submitted.</p> <p>The reason for the failure needs to be investigated before rescheduling the job.</p>	N/A
Terminated (16)	<p>Job is terminated manually by the user.</p>	<p>The reason for the termination needs to be investigated. The job can be rescheduled.</p>	N/A
System Failure (20)	<p>When the job is terminated because of database server or network issues.</p>	<p>The reason for the System Failure needs to be investigated. The job can be rescheduled.</p>	N/A

Credit Memo Multi Thread Chain – JV Multithreaded Transaction Loader

Job Name	JV Multithreaded Transaction Loader
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	No
Reports Generated	No

Overview

The JV Multithreaded Transaction Loader job invokes the SMU job to load and submit journal voucher transactions using the JV Export.xml file.

Process Steps	Messages
1. Parameter Validation	<p>The process validates the Batch Parameters. And logs the following messages for invalid parameters:</p> <ul style="list-style-type: none"> • Run Started • Each parameter is listed • The process after completing the parameter validation logs the following message: • Batch Parameter validation completed

Process Steps	Messages
5. Loading and submitting the JV transactions	The process then calls the SMU process to load and submit the JV transactions.

Major Input

- JV Transaction XML file

Batch Parameters

The parameters shown below are with delivered values and what is implemented may vary.

Parameter	Description	Default Value
Block Size (BLOCK_SIZE)	A required number of records written into each split segment of the input file.	1
Commit Block Size (COMMIT_BLOCK_SIZE)	A required number of transactions to commit at a time.	1
File Input Directory (FILE_INPUT_DIR)	A required location where the xml file is read.	\$\$AMSROOT\$\$/ExportIm port
File List (FILE_LIST)	A required comma-separated list of files to be uploaded with multi-threaded processing.	JVExport.xml
File Output Directory (FILE_OUTPUT_DIR)	A required location for the file segments created.	\$\$AMSROOT\$\$/ExportIm port
File Prefix (FILE_PREFIX)	An optional prefix used on the filenames for the output file segments.	No default
Bypass Approval (I_SMU_BYPASS_APPROVAL)	A required indication if loaded transactions should have the bypass approval action applied (TRUE or FALSE).	TRUE
Logging Frequency (LOG_STATUS_INTERVAL)	A required number of seconds for the controller thread reporting on the status of the child threads to the system log.	300
Mode (MODE)	A required mode of operation for the SMU threads. (1=Import, 2=Import and Submit, 3=Import and Other Action)	2
Other Action (OTHER_ACTION)	A conditionally required action considered when the Mode is 3.	No default
Sleep Interval (SLEEP_INTERVAL)	A required frequency in seconds for internal controller thread for checking child processes.	5
SMU Catalog ID (SMU_CTLG_ID)	The required Catalog ID of the System Maintenance Utility job that is spawned.	3
Stagger Time	The required lag time, in seconds, between	30

(STAGGER_TIME)	the spawning of each child process.	
Transaction Status (S_SMU_DOC_STA_CD)	The required Transaction Status for loaded transactions (2-Ready, 1-Held).	1
Exception Report File Name (S_SMU_EXCEP_REP_FILE_NM)	The name of the exception file for System Maintenance Utility.	JVExcluded.txt
Exception Report Type S_SMU_EXCEP_REP_IND	A required level of detail and content in the exception report: (1-Detailed, 2-Failed Transactions, 3-Processed Transactions, 4-Failed Transaction Lines, 5-Transaction Status).	2
Thread Count (THREAD_COUNT)	Number of threads to use for processing.	6
Polling Interval (FILE_POLLING_INTERVAL)	A required number of seconds for file polling. Default value is 0.	0
Polling Maximum Time (FILE_POLLING_MAX_TIME)	A required number of seconds for file polling. Default value is 0.	0

Major Output

- Journal Voucher transactions loaded and submitted

Selection Criteria

N/A

Sort Sequence

N/A

Problem Resolution

If the job ends with a return code of Failed and above, the job can be restarted when another instance of the job has not been scheduled and run successfully. If another instance of the job has already been scheduled and run successfully, then this job should not be restarted, a new job should only be scheduled.

The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain.

Return Code	Condition
Successful (1)	All of the transactions loaded were submitted successfully.
Warning (4)	This return code is issued when some of the records failed to load whereas all other records were loaded successfully.
Non-Fatal Error (8)	This return code is issued when all records failed to load.
Failed (12)	The job fails under the following conditions:

Return Code	Condition
	<ul style="list-style-type: none"> Parameters are invalid. When the input file is not found in the specified directory. Restart failed because another instance of the Accrual Process chain has already been run successfully. Runtime exceptions encountered for any unexpected situations. <p>When the job ends with a return code of Failed, subsequent jobs in the chain are set to Inactive.</p>
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain are set to Inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain are set to Inactive.

Credit Memo Multi Thread Chain: JV Report

Job Name	JV Report
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	No
Reports Generated	Yes (2)

Overview

The JV Report job generates two reports. The Credit Memo Processed Report lists the credit memo transaction selected and the journal voucher created. The Credit Memo Exception Report displays the generated journal voucher transactions that rejected.

Process Steps	Messages
1. Parameter Validation	<p>The process validates the Batch Parameters and logs the following messages for invalid parameters:</p> <ul style="list-style-type: none"> Run Started Each parameter is listed The process after completing the parameter validation logs the following message: Batch Parameter validation completed
6. Generation of Report	The process generates both reports.

Major Input

- JVExcluded.txt
- JVProcessing.txt

Batch Parameters

The parameters shown below are with delivered values and what is implemented may vary.

Parameter	Description	Default Value
Action Code (ACTION_CODE)	A required code that controls report generation.	??
Export Location (AMSEXPORT)	The required directory of where both the export file and parameter file reside.	\$\$AMSROOT\$\$/ExportImp ort
Exception File (GEN_EXCLUDED_FILE)	The required file name created in the previous job step of rejected transactions.	JVExcluded.txt
Processed File (GEN_PROCESS_FILE)	A required input parameter of the TXT file name created with each transaction created for reporting.	JVProcessing.txt
Progress Counter (PROG_CTR_SZ)	A required processing parameter that defines a counter to be used in the progression messages issued for report creation. If left blank, 100 will default.	No default
Client Name (CLNT_NM)	An optional name to appear on each report header.	No default

Major Output

- Credit Memo Processed Report
- Credit Memo Exception Report

Job Return Code

Return Code	Condition
Successful (1)	All transactions submitted successfully.
Warning (4)	Job does not end with this return code.
Non-Fatal Error (8)	At least one transaction failed to submit.
Failed (12)	Parameters are invalid: <ul style="list-style-type: none"> • When the input file is not found in the specified directory • Runtime exceptions encountered for any unexpected situations When the job ends with a return code of Failed, subsequent jobs in the chain will be set to Inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain will be set to Inactive.

Return Code	Condition
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to Inactive.

Selection Criteria

N/A

Sort Sequence

N/A

Problem Resolution

If the job ends with a return code of Failed and above, the job can be restarted when another instance of the job has not been scheduled and run successfully. If another instance of the job has already been scheduled and run successfully, then this job should not be restarted, a new job should only be scheduled.

The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters are validated and the reports were created with no failures reported.	N/A	N/A
Warning (4)	Job does not end with this return code.	N/A	N/A
Non-Fatal Error (8)	One or more transactions failed to submit.	Need to complete submit process after the reason for the failure is resolved.	N/A
Failed (12)	Job failed due to fatal conditions.	Job can fail under the following conditions: 1) Invalid parameter 2) Runtime exception. Need to schedule a new chain starting at this point after the reason for the failure is resolved.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Need to schedule a new chain starting at this point after the reason for the termination is resolved.	N/A
System Failure	When the job is	Reason for the failure needs to	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
(20)	terminated because of database server or network issues.	be investigated. Need to schedule a new chain starting at this point after the reason for the failure is resolved.	

2.1.8 Encumbrance Reclassification

Chain Job Name	Encumbrance Reclassification
Recommended Frequency	As per requirement.
Single Instance Required	Yes.
Can be restarted?	Yes, see the individual jobs for more details.
Reports generated	Yes, some of the jobs in the chain generate the exception report. Please refer to the individual jobs for more details.

Overview

The Encumbrance Reclassification Process is a chain that reclassifies all of the ABS and PO Transaction Type transactions that have outstanding encumbrance balances. It selects the eligible PO and ABS transactions and processes the modification transaction to reflect the recent funding split of the Program.

The Encumbrance Reclassification Process has the following batch jobs in the chain.

1. [Encumbrance Reclassification Process](#)
2. [Load and Submit Transactions](#)
3. [Enc Reclass Exception Report](#)

The Encumbrance Reclassification Process, when run in report mode, generates a report having all of the eligible records that are selected through the selection criteria and undergoes processing when run in update mode. The job ends with a Non Fatal error, and all of the subsequent jobs will become inactive.

When run in Update mode, the xml of the selected records are created; transactions are uploaded and submitted by the subsequent jobs in the chain.

Major Input

- DOC_ACTG – PO and ABS Transaction Type transactions from the Transaction Accounting table.
- R_MJR_PROG table
- R_PROG table
- R_PHASE_PROG table

Major Output

The process loads and submits the modification version of the selected PO and ABS transactions that need reclassification.

Chain / Job Return Code

The job will end with the following return codes as per conditions encountered.

Return Code	Condition
Successful (1)	All of the jobs end successfully.

Warning (4)	One of the jobs in the chain ends with a return code of "Warning".
Non Fatal Error (8)	One of the jobs in the chain ends with a return code of "Non Fatal Error" or Job ran in Report only mode.
Failed (12)	One of the jobs in the chain ends with a return code of "Failed".
Terminated (16)	One of the jobs in the chain ends with a return code of "Terminated".
System Failure (20)	One of the jobs in the chain ends with a return code of "System Failure".

Problem Resolution

If any of the jobs in the chain failed due to application errors it is advisable to restart the job after correcting the errors instead of rescheduling the job. Restarting the job will reduce the processing time since the job will resume from where it was last committed and select only the unprocessed records.

Please refer to the individual jobs for details regarding the specific job processes and problem resolution.

Encumbrance Reclassification Chain Job: Encumbrance Reclassification Process

Job Name	Encumbrance Reclassification Process
Recommended Frequency	As per requirement
Single Instance Required	Yes.
Can be restarted?	Yes.
Reports Generated	Yes.

Overview

The Encumbrance Reclassification Process kicks off all the other jobs in the chain. It populates the eligible PO and ABS records into a temporary table (RCLS_ENC_RPT_TMP). Depending upon the run mode of the job, a report of the records is generated from a temporary table. PORECLASS.xml and ABSRECLASS.xml (one for PO and another for ABS transactions) are created for the above records at the AMSEXPORT location.

Steps

The process can be divided into the following logical steps: parameter validation, selection of eligible records, inserting of eligible records into temporary table (RCLS_ENC_RPT_TMP), generating report and creating XML.

- 1. Parameter Validation** – The job parameters are validated before the actual processing starts.
 - **AMSLOGS, AMSPARMS** and **AMSEXPORT** are required parameters and are also validated against the valid directories.

- **Event Type** – The user can enter comma delimited event types. Event Type is a required parameter and it also has a check to validate if the entered value is valid on the Event Type table or not.
 - **Department** – Department, if entered, must exist on the Department table with the effective date ranges.
 - **Run Mode** – Run mode is a required parameter. Valid values are 1 for Report only mode, 2 for Update only mode and 3 for Report and Update mode. The entered value must be 1, 2 or 3.
 - **Include Modification Draft** – This is a required parameter, and accepts either value Y for yes or N for no. If the user enters Y, the modification draft transactions that are already in the application will be moved to conflict draft phase; otherwise, if No is selected such records for which a modification transaction exists will not be picked up.
2. **Selection logic** – If the parameters are validated successfully, eligible records are then selected from the PO and ABS transaction accounting.

The selection criteria should match the below conditions, that is:

- **Event Type** - The transactions with the specific event type that the user entered in the job parameter will be selected.
- **Department** - The transactions with the specific Department code, if entered by the user, in the job parameter will be selected.
- **Accounting Line Close Date** - The accounting line of the transaction should not be closed.
- **Major Program** – The Major Program on the accounting line section of the transaction must be active and must have the Reclassification Exclusion Flag as false. Major Program must be Front End Split only as marked on the MJPROG table.
- **Program** – The Program on the accounting section of the transaction must be active and must have the Reclassification Exclusion Flag as false on the Program Setup table.
- **Phase** - If Phase is entered on the accounting line of the transaction, it must be active and should have the Reclassification Exclusion Flag as false.

However, an exception to the above selected record is considered when the Modification version of the transaction already exists and is either in Pending phase or MOD_DRAFT transaction parameter is 'N' in the job parameters while running the job.

3. **Populate RCLS_ENC_RPT_TMP** – Pre-existing records are first deleted from the temporary table, and the records selected from the above selection criteria are then inserted into the RCLS_ENC_RPT_TMP table.
4. **Report Generation** – The report is generated, if the job run mode is report only or report and update mode, for the records from the temporary (RCLS_ENC_RPT_TMP) table.
5. **XML creation** – PORECLASS.xml and ABSRECLASS.xml are then created for the transactions, with RCLS_ENC_RPT_TMP as the source table, in the modification phase.

The following progression messages indicate the status of the job.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> • “Parameter AMSEXPORT is not a valid directory.” This message is displayed if AMSEXPORT is not a valid directory. Similar logs will be

Process Steps	Messages
	<p>displayed for AMSLOGS and AMSPARM parameters.</p> <ul style="list-style-type: none"> • “The Department specified is not valid.” This message is displayed when the Department entered is invalid. • “Include Modification Draft parameter is required.” This message is displayed when the specified parameter is not entered. • “Run Mode parameter is required.” or “Run Mode parameter must be 1, 2, or 3”, is displayed when the user has not entered the run mode or entered an invalid value. • “Include Modification Draft parameter must be Y for Yes or N for No.” This message is displayed when the user has entered values other than Y or N. • “One of the Event Types specified is not valid.” This message is displayed when any record is not valid on the Event Type table among all the Event types entered by the user.
2. Selection of Records	<ul style="list-style-type: none"> • “Selecting eligible records.” This message is displayed while records are getting selected.
3. Report generation	<ul style="list-style-type: none"> • “HTML report file path: <patch>. PDF report file path: <patch>.” These logs are displayed once Reports are successfully generated.

- The job has the restart capability and can be restarted if it was failed during the parameter validation, while inserting records into the temporary table, and while generating reports.
- Check point logic has been implemented while validating parameters, inserting records into the temporary table and while generating the report.
- The job can be run as per requirement.

Major Input

- DOC_ACTG – PO and ABS Transaction Type transactions from the Transaction Accounting table.
- R_MJR_PROG table
- R_PROG table

- R_PHASE_PROG table

Batch Parameters

Note: The default values listed are those delivered with the software. Actual values may vary based on your site’s setup.

Parameter	Description	Default Value
AMSEXPORT	Required. Export Location at PO/ABS Transaction Upload Job.	\$\$AMSROOT\$\$/ExportImport
AMSPARM	Required. Parameter Location at PO/ABS Transactions	\$\$AMSROOT\$\$/Parms
AMSLOGS	Required. Logs Location at PO/ABS Transactions	\$\$AMSROOT\$\$/Logs
DEPT_CD	Not Required. Department Code to be used as selection criteria.	No Default
EVNT_TYP	Required Event Types (comma delimited)	PR05, PR07
RUN_MODE	Required. Run Mode - 1 = Report Only, 2 = Update, 3 = Report and Update	No Default
MOD_DRAFT	Required. Include Encumbrances with a Modification Draft transaction in the Transaction Catalog? Valid Values are Y = YES or N = NO)	No Default

Major Output

ABSRECLASS.xml and PORECLASS.xml are generated for the modification transactions at AMSEXPORT.

Batch Return Codes

The following table shows the potential job return codes for the Encumbrance Reclassification Process.

Return Code	Condition
Successful (1)	The job ran to completion without any errors or exceptions. The selected records are displayed in the report and a transaction xml will be created at the AMSEXPORT directory depending upon the

	run mode.
Warning (4)	When no records are selected for processing and inserted to the temporary table.
Non Fatal Error (8)	When the report run mode is Report only.
Failed (12)	Parameters are invalid. Runtime exceptions encountered for any unexpected situations. When the job ends with a return code of Failed, subsequent jobs in the chain are set to inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain are set to inactive.

Sort Sequence

Records having similar Major Programs are grouped and displayed together.

Selection Criteria

- Select all PO and ABS final submitted transactions that have open accounting lines with user specified Department and Event Types, from the job parameters, and where Major Program, Program and Phase (Phase, if entered) are active with the Reclass Exclusion flag unchecked on respective tables.
- The Major Program must be Front End Split.
- If the Modification version of a final transaction is in Pending Phase then it is not selected by the process.
- If a Modification version of a final transaction exists in draft phase and MOD_DRAFT parameter of the Encumbrance Reclassification process is kept as 'N', those records are not selected by the process.

Problem Resolution

If the job ends with a return code of Failed and above, the job can be restarted when another instance of the job has not been scheduled and run successfully. If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted, a new job should only be scheduled.

The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain.

Possible Return Codes	Condition	Recommendation	Other Instructions
Failed (12)	Failed while validating the parameter.	Recommendation: Schedule a new job, with the valid parameters.	Refer to the job log message for the reason of invalid

	<p>Sample Message: Invalid Parameters (As specified in the Progressive log messages).</p> <p>Failed with a runtime Exception.</p>	<p>Recommendation: Refer the application logs.</p>	<p>parameter.</p> <p>User need to try to find out the problem that made the job fail unexpectedly, through application logs if possible, and then try scheduling a new instance.</p>
--	--	---	--

Encumbrance Reclassification Chain Job: Load & Submit Transactions

Job Name	Load & Submit Transactions
Recommended Frequency	As per requirement.
Single Instance Required	Yes
Can be restarted?	Optionally, based on the Save Restart Information parameter.
Report Generated	No

Overview

This job starts by first validating the batch parameters. If the parameters are valid, then it loads the records from the XML file(s) generated by the Encumbrance Reclassification job into the Transaction Catalog. This job uses the MultiProcessImport feature to load and submit the transactions created in the XML file from the first job step. If the parameters are not valid, the job issues appropriate messages and ends with a status of *Failed*.

Major Input

- PO Transaction XML
- ABS Transaction XML

Process Steps	Messages
1. Parameter Validation	<p>Validating Batch Parameters.</p> <p>Each valid parameter is listed followed by any value.</p> <p>If the Thread Count is not a valid integer then "Invalid value received for parameter THREAD_COUNT (expected number): ##" is recorded in the job log.</p> <p>If the Thread Count is not a valid integer > 0, then</p>

	<p>“The valid job thread count starts with 1. Receive value: ##” is recorded in the job log.</p> <p>If the Mode is not provided then “MODE parameter cannot be left blank (expected values 1, 2, or 3)” is recorded in the job log.</p> <p>If the Mode is provided and mode is not => 1 and not =< 3 then “Invalid value received for parameter MODE (expected values 1, 2, or 3): ##” is recorded in the job log.</p> <p>If the Commit Block Size is not a valid integer > 1 then “Invalid value received for parameter COMMIT_BLOCK_SIZE (expected value >1): ##” is recorded in the job log.</p> <p>If the Block Size is not a valid integer > 1 then “Invalid value received for parameter BLOCK_SIZE (expected value >1): ##” is recorded in the job log.</p> <p>If the Polling Frequency is a valid integer and < 5 then “Invalid value received for parameter SLEEP_INTERVAL (expected value >=5): ##” is recorded in the job log.</p> <p>If the Logging Frequency is a valid integer and less than 30 then “Invalid value received for parameter LOG_STATUS_INTERVAL (expected value >= 30): ##” is recorded in the job log.</p> <p>If the Catalog ID of the System Maintenance Utility job is not a valid integer greater than one then “Invalid value received for parameter SMU_CTLG_ID (expected value >1): ##” is recorded in the job log.</p> <p>If the Other Action is not provided and run mode is 3, then “Other Action is required when run on mode 3” is recorded in the job log.</p> <p>If the Other Action is provided and run mode is 1 or 2 then “Other Action is valid only for Mode 3” is recorded in the job log.</p> <p>If the location of the source file of records is not a valid directory, then “Invalid value read for parameter FILE_INPUT_DIR (expected value = a valid directory): #####” is recorded in the job log.</p> <p>If the Stagger Time is not a valid integer > 1 then</p>
--	---

	<p>“Invalid value received for parameter STAGGER_TIME (expected value >1): ##” is recorded in the job log.</p> <p>If the Comma Separated list of input XML files is not provided then “No input file to process” is recorded in the job log.</p> <p>If the Comma Separated list of input XML files is provided and a file does not exist or does not end with xml then “Invalid / non existing file input part in value for parameter FILE_LIST (expected value = a valid list if file):####” is recorded in the job log.</p> <p>If the Output Location for the file segments is not a valid directory then “Invalid value read for parameter FILE_OUTPUT_DIR (expected value = a valid directory): ####” is recorded in the job log.</p> <p>Parameter validation completed.</p>
<p>2. Multiple Job Processing</p>	<p>Setting up job for parameter file: (Parameter Location path)\Parm_#####_#.txt</p> <p>Enabled job with Id=#####</p> <p>Created Job ##### for Parm file (Export Import directory path)\Parm_#####_#.txt</p> <p>(Repeated for each file and job combination)</p> <p>Committed child jobs</p> <p>Child job pending count:# Job Id[#####] Status=X (Repeated for each job)</p> <p>Deleting part exception file: (Export Import directory path)\EncRclsExep_#####_#.txt</p> <p>(Repeated for each file)</p> <p>Deleting split input XML file: (Export Import directory path)\PODocuments_#####_#.xml, ABSDocuments_#####_#.xml</p> <p>(Repeated for each file)</p>

	Run Ended
--	-----------

Batch Parameters

Parameter	Description	Default Value
BLOCK_SIZE	Optional block size for the number of transactions to be split out into files. If left blank, 1 will be the default. Set this performance parameter to a value that will split an XML file into individual files sufficient for timely processing.	10
COMMIT_BLOCK_SIZE	Optional number of records to commit at a time. If left blank, the ADV30Parms.ini file will supply a value.	10
FILE_INPUT_DIR	Required location of the source file of records.	\$\$AMSROOT\$\$/ExportImport/Encumbrance Reclassification
FILE_LIST	Required comma separated list of input XML files to load.	<ul style="list-style-type: none"> • PO Transaction XML file. • ABS Transaction XML file
FILE_OUTPUT_DIR	Required output location for the file segments.	\$\$AMSROOT\$\$/ExportImport/ Encumbrance Reclassification
FILE_PREFIX	Optional file prefix for Parameter and Data Files created by jobs.	No default
I_SMU_APPLY_OVERRIDES	Required System Maintenance Utility flag to indicate Apply Overrides in Import Mode.	true
I_SMU_BYPASS_APPROVAL	Required System Maintenance Utility flag to indicate Bypass Approvals in Import Mode.	false
I_SMU_BYPS_ADNT_FL	Required System Maintenance Utility flag to indicate Bypass ADNT in Import Mode.	true
I_SMU_DOC_STATUS_CD	Required System Maintenance Utility Transaction Status for Import Mode.	2
I_SMU_OVERRIDE_LEVEL	Optional System Maintenance Utility Override Level for Import Mode. If 0 or left blank and the Apply Overrides parameter is <i>true</i> then the level of the user executing the chain job will be used.	No default
I_SMU_RESTART_FLAG	Required System Maintenance Utility flag to indicate Save Restart info in	true

FL	Import Mode.	
LOG_STATUS_INTERVAL	Required logging frequency (in seconds) for the controller thread reporting status of child threads to the system log.	300
MODE	Required mode (1=Import Only, 2-Import & Submit, 3-Import, Other Action, & Submit).	2
SLEEP_INTERVAL	Required polling frequency (in seconds) for internal controller thread for checking child processes.	5
SMU_CTLG_ID	Required Catalog ID of the System Maintenance Utility job that is spawned as the child process.	3
STAGGER_TIME	Required lag time in seconds between the spawning of each child process.	30
S_SMU_APPLY_OVERRIDES	Required System Maintenance Utility flag to indicate Apply Overrides in Submit Mode.	true
S_SMU_BYPASS_APPROVAL	Required System Maintenance Utility flag to indicate Bypass Approvals in Submit Mode.	false
S_SMU_EXCEPT_FILE_NM	Required System Maintenance Utility Exception File for Submit Mode.	EncRclsExep.txt
S_SMU_EXCEPT_IND	Required System Maintenance Utility Exception Report Indicator for Submit Mode. Values are: 1 = Detailed, 2 = Failed Transactions, 3 = Processed Transactions, 4 = Failed Transaction Lines, and 5 = Transaction Status.	4
S_SMU_EXP_SEV_FL	Optional System Maintenance Utility flag to indicate the Severity Flag in Submit Mode.	1
S_SMU_OVERRIDE_LVL	System Maintenance Utility Override Level for Submit Mode. If 0 or left blank and the Apply Overrides parameter is <i>true</i> then the level of the user executing the chain job will be used.	No default
S_SMU_RESTART_FL	Required System Maintenance Utility to indicate Save Restart info in Submit Mode.	true
THREAD_COUNT	Required number of jobs to start.	1

Major Output

- The result of this job is the loading of transaction records from the input file. Files are first split and then imported.
- During job processing there will be two categories of temporary files (they will be deleted upon successful job completion):
 - Split input files
 - Exception report files for each child job.
- If any of the child jobs encounter an exception an appropriate exception message will be merged into the Exception Report file for the job (specified by name in the job parameters).
- A parameter file is generated for each split XML file, and then the spawned jobs read those input parameter files in a sequential order.

Batch Return Codes

The following table shows the potential job return codes for Load & Submit Transactions.

Return Code	Condition
Successful (1)	All of the records are loaded into the Transaction Catalog successfully or the input file is empty.
Warning (4)	This return code is issued when some of the records failed to load where as all other records were loaded successfully.
Non Fatal Error (8)	None of the records get loaded into the Transaction Catalog.
Failed (12)	<ul style="list-style-type: none"> • Parameters are invalid • When the input file is not found in the specified directory • Runtime exceptions encountered for any unexpected situations • When the job ends with a return code of Failed, subsequent jobs in the chain are set to inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain are set to inactive.

Sort Sequence

N/A

Selection Criteria

The input record files (specified in the Job Parameter) are selected.

Problem Resolution

If the process fails for any reason the Load and Submit Transactions job supports restart functionality. Upon restarting the Load and Submit Transactions job, the failed job will be picked up and run. A checkpoint is maintained for the overall Load and Submit Transactions job as well as the individual child jobs to identify how far the job successfully executed before failing. Therefore, when the user restarts the failed job the system does not start from the first step of the Load and Submit Transactions but instead starts processing from the exact point at which it failed.

For example, let's assume that the failed job split the input file into five child jobs and that the third and the fourth child jobs failed. When restarted, only those failed child jobs, that is, the third and the fourth ones will be picked up and processed. The restarted job would not process child jobs one, two or three given they have already completed successfully in the earlier run.

The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain. For general errors and recommendations, refer to the SMU Transaction Upload run sheet in the *CGI Advantage Financial – Utilities Run Sheet Guide*.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All transactions were loaded and submitted.	N/A	N/A
Warning (4)	This return code is issued when the job fails to load some of the transactions. Sample Message: Unable to load all of the transactions into the catalog.	Analyze the reason why records failed to load to the Transaction Catalog. When the current run is a re-execution of a previous run (the same input file is used in a new job) this error may be seen because the records that previously loaded successfully will not load again (duplicate entry on the Transaction Catalog is not allowed). If the records failed to load to the Transaction Catalog due to any other reason, then analyze the reason, resolve it and schedule a new job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.
Non-Fatal Error (8)	This return code is issued when the job failed to load all of the transactions.	Correct what is most likely a setup issue and restart the spawned SMU job and then restart the load/submit job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.
Failed (12)	Failed under the following conditions: <ul style="list-style-type: none"> Issues in spawning jobs 	The reason for the failure needs to be investigated and fixed before restarting the job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be

	<ul style="list-style-type: none"> Files not found Runtime exceptions Parameter Edits 		restarted – only a new job should be scheduled.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated and fixed before restarting the chain.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before restarting the chain	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.

Encumbrance Reclassification Chain Job: Enc Reclass Exception Report

Job Name	Enc Reclass Exception Report
Recommended Frequency	As per requirement.
Single Instance Required	Yes
Can be restarted?	No
Report Generated	Yes. The job generates the reports in PDF and HTML formats.

Overview

This job in the Encumbrance Reclassification Process generates an exception report that lists all of the errors encountered when the PO/ABS transactions were submitted in the earlier step. The report contains the following information:

- Rejected PO/ABS transactions
- Detailed error description along with the error code.

The following table shows the progression messages issued in this job.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> Validating Batch Parameters Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. It will be followed by message "Batch Parameter validation failed".

Process Steps	Messages
	<ul style="list-style-type: none"> Batch Parameters validation completed.
2. Creating exception report	<ul style="list-style-type: none"> Generation of Exception Report started. Generation of Exception Report completed.

Restartability Information

- Job cannot be restarted in this run mode. If the job fails in any of the above steps, a new instance of this job should be scheduled after correcting the errors that caused the job to fail.

Major Input

- Draft PO/ABS transactions in the catalog with a Rejected status
- Batch Parameters

Parameter	Description	Default Value
AMSPARM	Parameter Location Required. Parameter Location.	\$\$AMSROOT\$\$/Parms

Output

Encumbrance Reclassification Exception Report

Batch Return Codes

The following table shows the potential job return codes for the Enc Reclass Exception report job in the chain.

Return Code	Condition
Successful (1)	All of the transactions generated in that run submitted successfully.
Warning (4)	Not Applicable for this job.
Non Fatal Error (8)	Not Applicable for this job.
Failed (12)	<ul style="list-style-type: none"> Input file is blank. Input parameter file is not found in the specified folder. Runtime exceptions encountered for any unexpected situations. When the job ends with a return code of failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain are set to inactive.

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Input file is not entered. Sample Message: Exception file is required	Specify the Input file name and schedule a new job.	
	Input file is not found on the specified directory. Sample Message: The system cannot find the specified file.	Make sure that the specified file exists in the directory and restart the job. Alternatively, a new job can be scheduled. At times, the directory path may not be accurate. If the job cannot be restarted, schedule a new job.	Correct the path and reschedule the job.
	Failed while restarting the job since another instance of the job has already been run successfully. Sample Message: Cannot restart the job since another instance of this job has already been run successfully.	Schedule a new job.	

	<p>Failed because of runtime exceptions for an unexpected situation.</p> <p>Sample Message: Error occurred during validateParam: <Runtime Exception Message></p>	<p>The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.</p>	<p>System error log and VLS log should be investigated to find out the possible reason of exception.</p>
Terminated (16)	<p>Job is terminated manually by the user.</p>	<p>The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.</p>	
System Failure (20)	<p>When the job is terminated because of database server or network issues.</p>	<p>The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.</p>	

2.1.9 FHWA Cleanup Process

Chain Job Name	FHWA Cleanup Process
Recommended Frequency	Upon demand after notice of the rejected file and before the next run of the Reimbursement Output Process.
Single Instance Required	Yes
Can be restarted?	No
Report Generated	Yes (2)

Description

The Federal Highway Administration (FHWA) requires a specific file format for reimbursement. When the electronic file is sent to the FHWA, it is immediately run against an editing routine in FHWA's Fiscal Management Information System (FMIS). If the FMIS editing routine encounters any type of error on any record of the electronic file, then the entire file is rejected by FHWA. Examples of the types of validations performed in FMIS are:

- Project-Agreement-Number must be valid and active.
- Sufficient budget authority must be available.
- The type of costs (that is, FHWA-Activity, FHWA-Transaction-Type, etc.) must be allowable against the Project-Agreement.

If the file gets rejected in FMIS due to problems with data in FMIS, FHWA will automatically re-process the file after the FMIS data is corrected. It should be possible to re-submit the newly modified electronic file containing only the "good" records without being required to re-run the entire Reimbursement Output Process. The FHWA Cleanup Process takes care of this cleaning. Upon receipt of a rejected file, a user has to update the rejected records on FHWA Rejected Records (FWWACLN).

The FHWA Cleanup Chain consists of the following job steps:

1. FHWA Cleanup
2. RE Load
3. RE Submit
4. Failed RE Report
5. FHWA File Creation

Processing Steps

The five job steps above combine to perform the following processing steps.

Step 1 - Selection of Records

Selects all records from FHWA Summary (R_FHWA_REC) marked as rejected (REJCT_REC_FL = 1).

Step 2 - Select Rejected Records from Reimbursement History (REIMHIST / R_REIM_HIST)

Retrieve the related records from Reimbursement History. Obtain a listing of receivable transactions for the rejected records.

Step 3 - Create XML for Modified Receivables

Creates an XML file modifying the Line Amount to zero on those accounting lines that rejected.

Step 4 - Generate Detail Report

A report is generated for all the rejected records with the details of each record's original disbursement record information obtained from Reimbursement History.

Step 5 - Load and Submit Receivable Modifications

Using SysManUtil the XML for the modified RE's is loaded and submitted.

Step 6 - Generate Failed Transaction Report

From the flat file generated by SysManUtil a report is generated with all the failed lines and its associated errors.

Step 7 - Create a Corrected FHWA Electronic File

A new corrected flat file is created containing the remaining 'good' records in FHWA Summary. The name of the flat file (file prefix and job ID) is stored in FHWA Summary when the file was originally created.

Step 8 - Copy records to the Reimbursement Request or Ready for Reimbursement

Based on a parameter, the rejected records are moved to Reimbursement Request (R_REIM_REQ) or Ready for Reimbursement (CA_REIM). If put in Reimbursement Request, the records are processed in the next run of the Reimbursement Output Process. If put in Ready for Reimbursement, the records are processed in the next Reimbursement Selection and Calculation process.

Step 9 - Delete the Rejected Records

The rejected records are deleted from the Reimbursement History. The rejected records on FHWA Summary are deleted.

Input

- FHWA Summary (R_FHWA_REC)
- Reimbursement History (REIMHIST / R_REIM_HIST)
- RE Transaction Header (RE_DOC_HDR)
- RE Transaction Vendor (RE_DOC_VENDOR)
- RE Transaction Accounting (RE_DOC_ACTG)

Parameters

The default values listed are those delivered with the software. Actual values may vary based on your site's setup.

FHWA Cleanup

Parameter	Description	Default Value
Export Directory (AMSEXPORT)	The required folder location where the Receivable XML file will be written.	\$\$AMSROOT\$\$/ExportImport
Parameter File Directory	The required folder location where the parameter file	\$\$AMSROOT\$\$/Parms

(AMSPARM)	created in the first job will be written for subsequent.	
Log Directory (AMSLOGS)	The required folder location for writing out log files for report creation.	\$\$AMSROOT\$\$/Logs
Client Name	An optional name for the header section of reports generated.	(Blank)
Moving the Rejected Records (MOVE_REJECTED_RECORDS_TO)	The required indication whether rejected records should be moved to the Reimbursement Request (1) or Ready for Reimbursement (2)	1

RE Load

Parameter	Description	Default Value
SMU Action Code (ACTN_CD)	The required action of Import for SMU.	171
Commit Block Size (COMMIT_BLOCK)	A required performance parameter to control the number of records committed.	10
File Name (FILE_NM)	The required file name and location of the modified RE XML file to import, created in the first job step.	\$\$AMSEXPORT\$\$/REMOD.xml

RE Submit

Parameter	Description	Default Value
SMU Action Code (ACTN_CD)	The required action of Submit for SMU.	162
Transaction Phase (DOC_PHASE_CD)	The required transaction phase for submit processing (1-draft)	1
Transaction Status (DOC_STA_CD)	The required transaction status for submit processing (1-held).	1
Transaction Type (DOC_TYP)	The required transaction type for submit processing.	RE
Exception Report File Name (EXCEP_REP_FILE_NM)	The required file name and location of the exception report file.	\$\$AMSLOGS\$\$/FailedRE.txt

Exception Report Indicator (EXCEP_REP_IND)	A required output parameter directing the level of detail in the exception report: 1 = detailed, 2 = failed, 3 = processed, 4 = failed lines	4
Parameter File (PARAM_FILE)	The required file name and location of the parameter file created in the first job step.	\$\$AMSPARM\$\$/FHWA_PARM.txt

Failed RE Report

Parameter	Description	Default Value
Log Directory (AMSLOGS)	The required folder location for writing out log files for report creation.	\$\$AMSROOT\$\$/Logs
Client Name	An optional name for the header section of reports generated.	(Blank)
Exception Report File Name (EXCEP_REP_FILE_NM)	The required file name of the exception report file.	FailedRE.txt

FHWA File Creation

Parameter	Description	Default Value
Export Directory (AMSEXPORT)	The required folder location where the Receivable XML file will be written.	\$\$AMSROOT\$\$/ExportImport
Parameter File Directory (AMSPARM)	The required folder location where the parameter file created in the first job will be written for subsequent	\$\$AMSROOT\$\$/Parms

Outputs

- FHWA Electronic Bill
- Receivable XML file
- Report of Rejected Records
- Report of Failed RE Transactions
- Reimbursement History (REIMHIST / R_REIM_HIST)
- Reimbursement Request (R_REIM_REQ)
- Reimbursement Selection (CA_REIM)

- FHWA Summary (R_FHWA_REC)

Selection Criteria

- Select OTPT_DOC_ID and OTPT_DOC_ACTG_LN_NO from FHWA Summary (R_FHWA_REC) whose Reject Flag (REJCT_REC_FL) equals “true (1).”
- Select records from Reimbursement History (R_REIM_HIST) matching OTPT_DOC_ID and OTPT_ACTG_LN_NO.
- Select records from RE_DOC_HDR where DOC_ID, DOC_CD, and DOC_DEPT_CD match OTPT_DOC_ID, OTPT_DOC_CD, and OTPT_DOC_DEPT_CD for the Final version.
- Select records from RE_DOC_VEND where DOC_ID, DOC_CD, DOC_DEPT_CD and DOC_VERS_NO match the respective field on RE_DOC_HDR.
- Select the records from RE_DOC_ACTG where DOC_ID, DOC_CD, DOC_DEPT_CD and DOC_VERS_NO match the respective field on RE_DOC_HDR and DOC_ACTG_LN_NO matches the rejected accounting line numbers.

Sort Criteria

Records selected from the Reimbursement History (R_REIM_HIST) are sorted by:

- Customer
- Output Transaction ID
- Output Accounting Line Number

Records selected from the FHWA Summary (R_FHWA_REC) for the electronic file creation are sorted by Customer ID.

Problem Resolution

If the process fails for any reason then perform the corrective actions:

- Check that the customer record exists on the FHWA Summary.
- Check the RE.XML file that is created in this directory DATAFILES\ExportImport.
- Check the Log file for any errors that may have occurred while the program was running.
- This batch program produces new Advantage transactions, which are subject to the line limit functionality constraints. Sites should ensure that they run this job with parameters set to ensure that the created transactions are within the line limit controls.

2.1.10 FHWA FMIS Extract Process

Chain Job Name	FHWA FMIS Extract Process
Recommended Frequency	As needed
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes

Overview

The State Department of Transportation (DoT) sends Project details to the Federal Highway Authority (FHWA) for approval of project requests and to create an obligation request for the project amount. The request can be sent in an XML format through Electronic Data Sharing (EDS) to FHWA's Financial Management and Information System (FMIS). For each project, the XML contains a header, detail, and location details. FHWA in-turn communicates its response to the project request through another XML format, which has most of the details originally sent by the DoT.

The FHWA FMIS Extract process facilitates generating the XML file needed for sending Project Requests to FHWA in an electronic format. The process starts by selecting the new project budget transactions, which are pending for approval. The process uses the project budget details queries for additional project data from numerous set up pages to build the details needed such as Project details, Funding details, and Geospatial Information System (GIS) details on the XML, which can create obligation requests in FMIS. The process subsequently moves the Project Budget transactions into workflow by applying the required approval.

This chain has the following jobs:

- [Select Budget Transactions](#)
- [Build Federal Funding Associations](#)
- [Build Detailed Info](#)
- [Build GIS and Non-GIS Data](#)
- [Generate FHWA FMIS Extract File](#)
- [Apply Workflow Approval for Budget Transaction](#)

Note: Even though the above jobs in the chain can be run individually by disabling other jobs, it is recommended to always run the entire chain.

Major Input

- Budget Transaction Header (BG_DOC_HDR)
- Budget Line (BG_DOC_LN)
- Program (PROG / R_PROG)
- Program Phase (PHPRG / R_PHASE_PROG)

- Funding Profile (FPFLST / R_FPRFL)
- Funding Priority (FPFLST / R_FPRTY)
- Funding Line (FPFLST / R_FLINE)
- Vendor Customer (VCUST / R_VEND_CUST)
- Region (REGION / R_RGN)
- Workflow Approval Work List (WF_APRV_WRK_LST)
- FMIS Project Header (R_FMIS_PROJ_HDR)
- FMIS Project Detail (R_FMIS_PROJ_DET)
- FMIS Project Location - GIS (R_FMIS_LOC_PROJ_GIS)
- FMIS Project Location – Non-GIS (R_FMIS_LOC_PROJ_NON_GIS)

Major Output

- XML file for transmission to FMIS
- Approval of Project Budget Transactions

Chain / Job Return Code

The following table shows the potential return codes for the FMIS Extract Process. Note that the Chain job will end with the highest return code across all of the jobs.

Return Code	Condition
Successful (1)	All of the jobs end successfully.
Warning (4)	One of the jobs in the chain ends with a Return Code of Warning.
Non-Fatal Error (8)	One of the jobs in the chain ends with a Return Code of Non-Fatal Error.
Failed (12)	One of the jobs in the chain ends with a Return Code of Failed.
Terminated (16)	One of the jobs in the chain ends with a Return Code of Terminated.
System Failure (20)	One of the jobs in the chain ends with a Return Code of System Failure.

Problem Resolution

If any job in this chain fails, the failure conditions should be evaluated and fixed. Once the user has done so, a new chain should be scheduled from the beginning. Any records inserted into R_FMIS_PROJ_EXT from the failed chain will be deleted automatically by the process.

FHWA FMIS Extract Process: Select Budget Transactions

Job Name	Select Budget Transactions
Recommended Frequency	As needed
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This job selects new Project Budget transactions that are in the pending phase and at a level of approval defined as part of the job parameters and writes the records to the R_FMIS_PROJ_EXT table for further processing. Prior to writing these records, the process purges records that were not fully processed on R_FMIS_PROJ_EXT from previous runs.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> The job begins by validating parameters. When a required parameter is not specified the job fails by issuing the following messages: "Parameter <Name> is empty." "One or more job parameters are empty." If the parameters are valid then the batch parameter validation is completed.
2. Selection of Project Budget Transactions	<ul style="list-style-type: none"> The job first deletes all records that are not completely processed from R_FMIS_PROJ_EXT by identifying them using PROC_STA_IND = 'P' The job then selects new Project Budget Transactions and for Transaction processed the job writes the following message: "Processing BGPHER Transaction: <Transaction ID>. The job loads the selected records to R_FMIS_PROJ_EXT for further processing
3. Update Project Extract records	<ul style="list-style-type: none"> The job then performs various validations to ensure that only the budget lines for which project data can be processed further are considered. The job skips budget lines with Program having a Program Agreement Number of ERR9999 and writes the following message: "<Program Agreement Number> has already been processed." The job then runs series of queries to update Project Extract (R_FMIS_PROJ_EXT) table with data needed. The job sets Project Action to 'Modify' the Program Agreement Number exists in R_FMIS_PROJ_HDR and if it does not, exist considers it as new Project request. The job then updates Recipient details like signature and date for Approver, Certifier and Recommender based on the job parameters.

Process Steps	Messages
	<ul style="list-style-type: none"> Recipient Remarks (RECPNT_RMKS) is updated with null when the lumpsum amount is empty. When lumpsum amount is not empty the field is updated with “Lump Sum = \$<lumpsum amount>;not to exceed the legal pro rata.”

Input

BGPHR Transactions Submitted to “Pending” status for a Program with a Project Agreement Number.

Batch Parameters

The following are the delivered parameter values, which may have been updated through Batch Setup to meet local needs.

Parameter	Description	Default Value
Parameter location (AMSPARM)	Parameter Location at Select Budget Transaction Job	\$\$AMSPARM\$\$
Approval Override Indicator (APPLY_OVERRIDES)	Approval Override Indicator	False
Approval User ID (APRV_ID)	Approval User ID	(No Default)
Budget Transaction Approval Level (APRV_LVL)	Required approval level used to select Budget transactions.	(No Default)
Approval Role (APRV_ROLE)	Approval Role	(No Default)
Budget Transaction Phase Code (DOC_PHASE_CD)	Required Phase of the Budget transaction to be selected.	2 (Pending)
FWHA Approver (FHWA_APRVR_USER_DT)	Required preference for the approver name and date that will be written to the respective fields in the output tables and files. Valid values are 1 - Created By/Date and 2 - Modified By/Date.	(No Default)
FWHA Certifier (FHWA_CERT_USER_DT)	Required preference for the certifier name and date that will be written to the respective fields in the output tables and files. Valid values are 1 - Created By/Date and 2 - Modified By/Date.	(No Default)
Client DUNS ID (FHWA_CLIENT_DUNS_ID)	Required Dun and Bradstreet Number to be used when records are created. Enter a valid 9-digit number, as this	(No Default)

	value is not validated.	
FHWA Comment Subject Exclude List (FHWA_CMNT_SUB_EXCLD_LIST)	Required phrase used to filter Budget Transaction Comment Subjects (DOC_CMNT.CMNT_SUB_UP) from the Budget transactions to be excluded when populating the FMIS Recipient Remarks field on the FHWA FMIS Extract file. The length of the field must be no more than 4000 characters. Multiple values can be specified and are separated by commas. Any document comments that have that value, the entire comment will not be written. Comment field from Budgets docs Recipient Remarks field in the FMIS_PROJ_EXT table.	(No Default)
FHWA Budget Department Code (FHWA_DEPT_CD)	Required Department Code used to select Budget transaction lines. Enter a valid code, as this value is not validated.	(No Default)
FHWA Budget Transaction Code (FHWA_DOC_CD)	Required Transaction Code used to select Budget transactions. Enter a valid code, as this value is not validated.	(No Default)
FHWA Budget Transaction Department Code (FHWA_DOC_DEPT_CD)	Required Transaction Department Code used to select Budget transactions. Enter a valid code as this value is not validated.	(No Default)
FHWA Budget Transaction Unit Code (FHWA_DOC_UNIT_CD)	Optional Transaction Unit Code used to select Budget transactions. Multiple values can be specified separated by commas. Enter valid code(s) as the value(s) are not validated. If left blank, then all eligible records are selected.	(No Default)
FHWA Indirect Rate (FHWA_INDIRECT_RT)	Required rate used to set on newly inserted records in the output tables and files. It must be a number with up to two decimal places.	(No Default)
FHWA Recipient ID (FHWA_RCPT_ID)	Required Recipient ID used to select records from Project Header table. It is also as part of the file naming convention	(No Default)

	and updates a field in the FMIS file called RecipientId.	
FHWA Recommender (FHWA_RECOMM_USER_DT)	Required preference for the recommender name and date that will be written to the respective fields in the output tables and files. Valid values are 1 - Created By/Date and 2 - Modified By/Date.	(No Default)
State Code (FHWA_STATE_CD)	Required State code used as part of the name for the Emergency Relief Code when records are created in output tables and files. Enter a valid code, as this value is not validated.	(No Default)
Exclude Project Agreement Number (PROJ_AGMT_ERROR_CD)	Required Project Agreement Number that should be excluded from the selection of Budget transaction lines. This number is used to compare with the Project Agreement Number on the Program table.	BYPASS
User ID (USER_ID)	User ID	(No Default)
FWHA Is Subrecipient (FHWA_IS_SUBRECIP_PROJ)	FWHA Is Subrecipient. Valid values are Y or N.	N

Output

- Header entries on Project Extract (R_FMIS_PROJ_EXT) table

Problem Resolution

If this job does not finish successfully, there is no restarting. A new instance of the chain should be submitted after addressing the reason(s) for failure.

Job Return Codes

The following table shows the potential job return codes for this job.

Return Code	Condition
Successful (1)	The job ends with a Return Code of Successful when parameter validation is successful and at least one BGPFR transaction is selected and processed.
Warning (4)	The job ends with a Return Code of Warning when no eligible records are found.
Failed (12)	The job fails under the following conditions:

Return Code	Condition
	<ul style="list-style-type: none"> Parameters are invalid. Error encountered clearing temporary data from R_FMIS_PROJ_EXT Error in selecting eligible BGPFR transactions. Any data issues resulting in errors while processing selected BGPFR transactions. <p>When this job fails, subsequent jobs in the chain are set to <i>Inactive</i>.</p>
Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to <i>Inactive</i> .
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to <i>Inactive</i> .

Sort Criteria

N/A

Selection Criteria

The job selects new Project Budget Transactions with a Transaction Status of Submitted (4) from the Budget Transaction Header (BG_DOC_HDR) using the Transaction code, FHWA Department, Unit, Transaction Phase, and Approval Level from job parameters.

FHWA FMIS Extract Process: Build Federal Funding Associations

Job Name	Build Federal Funding Associations
Recommended Frequency	As needed
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This job performs additional updates to Project Extract (R_FMIS_PROJ_EXT) records containing project header information from the first job. The job determines the funding associations for a project by retrieving additional data from multiple cost accounting set up pages using the project budget line. The information needed are grouped into FMIS business key sets and updated to the Q database tables (R_FMIS_PROJ_EXT_Q1-Q5) on an iterative basis before finally writing them out to the Q6 table.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> The job begins by validating parameters. When a required parameter is not specified the job fails by logging the following messages:

Process Steps	Messages
	<p>“Exclude Project Agreement Number must not be empty.” “Parameter FHWA Budget Department Code is required.”</p> <ul style="list-style-type: none"> If the parameters are valid then the batch parameter validation is completed
2. Build Q1 – Q6 tables	<ul style="list-style-type: none"> The job deletes existing data from Q1-Q6 tables and writes the following message when records are successfully deleted: “Scratch tables successfully deleted”. The process then selects Project Agreement Number and Budget Transaction ID from R_FMIS_PROJ_EXT where PROC_STA_ID is not ‘Processed’ and where the Project Agreement Number is other than BYPASS and then proceeds to build Q1 to Q6 tables in that order. If there are any issues encountered in building the Q table records the process will issue error for Project Agreement Number and the job will fail. “Exception processing FPN <FPN>” “FPN <FPN> will not be processed” When Q tables are successfully built, the process logs the following message for each Q table: “<n> records added to Q(#Seq) tables”

Input

Please refer to the chain job

Batch Parameters

The following are the delivered parameter values which may have been updated through Batch Setup to meet local needs.

Parameter	Description	Default Value
FHWA At Risk PE Indicator (FHWA_AT_RISK_PE_IND)	Required parameter to determine if detail is an At-Risk PE detail. Specify the value that identifies the improvement type(s) associated with preliminary engineering. Parameters should be enclosed with colon delimiters (:IT01:). Multiple values can be specified separated by a colon (:IT01:IT02:).	(No Default)
FHWA Department Code FHWA_DEPT_CD)	FHWA Department Code	(No Default)
FHWA Business Key (FHWA_FMIS_BUS_KEY_LIST)	FHWA Business Key	FHWA_IMPR_TYP, FED_APPR_NO,

		PCT_FED_SHARE, DEMO_ID, TOLL_FL, TEMP_MTCH_FL, INDIAN_RESERV_FL, APPALACHIA_FL, HWYS_FOR_LIFE_FL
FHWA Innovative Project Delivery (FHWA_INNOV_PROJ_DEL_IND)	Required parameter used by FHWA to identify the federal appropriation number(s) for a project incorporating innovations is eligible for increased funding. This is identified with Funding Program code(s). Parameters should be enclosed with colon delimiters. Multiple values can be specified separated by a colon.	(No Default)
FHWA State Funding Agency List (FHWA_STATE_FNDG_AGY_LIST)	FHWA State Funding Agency List. Customer ID determines the population of State Amount.	(No Default)
Project Agreement Error Code (PROJ_AGMT_ERROR_CD)	Optional Project Agreement Number that should be excluded from the selection of Budget transaction lines. This number is used to compare with the Project Agreement Number on the Program table. Enter valid code(s) as the value(s) are not validated.	BYPASS

Output

- FMIS Detail Rollup Q1 (R_FMIS_PROJ_EXT_Q1)
- FMIS Detail Rollup Q2 (R_FMIS_PROJ_EXT_Q2)
- FMIS Detail Rollup Q3 (R_FMIS_PROJ_EXT_Q3)
- FMIS Detail Rollup Q4 (R_FMIS_PROJ_EXT_Q4)
- FMIS Detail Rollup Q5 (R_FMIS_PROJ_EXT_Q5)
- FMIS Detail Rollup Q6 (R_FMIS_PROJ_EXT_Q6)
- FMIS Project Extract Table (R_FMIS_PROJ_EXT)

Problem Resolution

The following table shows the potential job return codes for this job.

Job Return Codes

If this job does not finish successfully, there is no restarting. A new instance of the chain should be submitted after addressing the reason(s) for failure.

Return Code	Condition
Successful (1)	The job ends with a Return Code of Successful when parameter validation is successful and at least one record was selected and processed. Alternatively: All transactions in the XML file were loaded successfully.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameter validation failed. • Error is encountered clearing scratch tables Q1-Q6 • Error populating scratch tables for some FPNs • Run time exceptions for unexpected situations. When this job fails, subsequent jobs in the chain are set to <i>Inactive</i> .
Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to <i>Inactive</i> .
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to <i>Inactive</i> .

Sort Criteria

NA

Selection Criteria

This job selects all records from the R_FMIS_PROJ_EXT table created in the first job of the chain, and using data from other tables, it populates the Output tables for use in Detail construction.

FHWA FMIS Extract Process: Build Detail Info

Job Name	Build Detail Info
Recommended Frequency	As needed
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This job reads the Q6 (R_FMIS_PROJ_EXT_Q6) records and builds the data needed to capture the Project Details section under the Project Header on the XML file for FHWA. The job writes Detail records as child record of Project Header on R_FMIS_PROJ_EXT.

Process Steps	Messages
<p>1. Parameter Validation</p>	<ul style="list-style-type: none"> • The job begins by validating parameters. • If the Program Rollup Level specified is not numeric the process issues the following message: “Invalid Value for Program Rollup parameter” • If the Program Rollup Level is not between 1-5 the process issues the following message: “Program Roll-up Level should be between 1 and 5” • If the Program Rollup value is blank the process issues the following message: “Program Rollup value must not be empty” • If the Program Rollup value does not have a colon separator between multiple values, the process issues the following message: “Program Rollup value must be in the format VALUE1:VALUE2.” • For any parameter validation failure, the process logs the following message: “Job Parameter validation failed” and the job returns a code of Failed. • If a parameter is invalid, the invalid value is displayed in the log. • Where parameters are valid the process writes the following message: “Job Parameters have been validated.”
<p>2. Write Project Detail records</p>	<ul style="list-style-type: none"> • In this step the job builds Detail records for Project Header. It updates R_FMIS_PROJ_DET with Funding details like Federal Fund Amount, Advance Construction Fund Amount, Advance Construction Fund Amount and other details needed for FMIS like Toll Credits flag, Temporary Match flag, etc with details from the Budget Transaction or the set-up tables or the Process parameters. • Other details updated from Q6 table includes Funding Types like Advance Construction Funds, Federal Funds, State Funds, Local Funds, Private Funds, Soft Match, Flex Funds

Input

Please refer to the chain job.

Batch Parameters

The following are the delivered parameter values, which may have been updated through Batch Setup to meet local needs.

Parameter	Description	Default Value
FHWA Budget Department Code	Required Department Code	(No Default)

(FHWA_DEPT_CD)	used to select Budget transaction lines. Enter a valid code as this value is not validated.	
FHWA Business Key (FHWA_FMIS_BUS_KEY_LIST)	FHWA Business Key	FHWA_IMPR_TYP, FED_APPR_NO, PCT_FED_SHARE, DEMO_ID, TOLL_FL, TEMP_MTCH_FL, INDIAN_RESERV_FL, APPALACHIA_FL, DESIGN_BUILD_FL, CMGC_FL, OTHER_SEP_14_FL, HWYS_FOR_LIFE_FL
Program Roll-up (PROG_RLLP_LVL)	Required. Required rollup used to define the FHWA Project Classifications of Preliminary Engineering (PE) and Preliminary Engineering for Structures (PES). Valid values are 1 – Code, 2 – Category, 3 – Group, 4 – Type, 5 – Class	(No Default)
Program Roll-up Value (PROG_RLLP_VALS)	Required. Required rollup values used to define the FHWA Project classification. Specify the value for PE first followed by the value for PES. Must have two values and be separated by colon. Suffix. Need to have 2 values.	(No Default)
Exclude Project Agreement Number (PROJ_AGMT_ERROR_CD)	Required Project Agreement Number that should be excluded from the selection of Budget transaction lines. This number is used to compare with the Project Agreement Number on the Program table. Enter valid code(s) as the value(s) are not validated.	BYPASS

Output

- FMIS Project Extract Table (R_FMIS_PROJ_EXT)

Problem Resolution

If this job does not finish successfully, there is no restarting. A new instance of the chain should be submitted after addressing the reason(s) for failure.

Job Return Codes

The following table shows the potential job return codes for this job.

Return Code	Condition
Successful (1)	Parameter validation is successful and applicable details created based on records in R_FMIS_PROJ_EXT for the FPN.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> Parameters are invalid. Error encountered building details for some FPNs Run time exceptions for unexpected situations When this job fails, subsequent jobs in the chain are set to <i>Inactive</i> .
Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to <i>Inactive</i> .
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to <i>Inactive</i> .

Sort Criteria

NA

Selection Criteria

This job selects all records from R_FMIS_PROJ_EXT_Q6 table.

FHWA FMIS Extract Process: Build GIS and Non-GIS records

Job Name	Build GIS and Non-GIS records
Recommended Frequency	As needed
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This job facilitates in building Location Details under a Project Header in the Project Extract (R_FMIS_PROJ_EXT) table. The job identifies whether location details for the Project need to be sent out as GIS or Non-GIS depending on the setup in Program GIS Location (PGL). The job then updates various information to build records on FMIS_PROJ_EXT table.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> The job begins by validating parameters. When a required parameter is not specified the following messages are issued: "<Parameter Name> must not be empty" and "Batch parameters validation failed." When Program Rollup Level is not between 1 and 5 the following message is issued: "<Parameter Name> must be between 1 and 5" When multiple Program Rollup Values are not separated by a colon the following message is issued: "<Parameter Name> must be in format VALUE1:VALUE2:VALUE3" When parameters are validated successfully the following message is issued: "Job Parameters validated successfully"
2. Write Project Location records	<ul style="list-style-type: none"> In this step the job run series of queries to the location information and also determines whether it is GIS and Non-GIS data along with additional details like Preliminary Engineering (PES), Preliminary Engineering for Structure (PES) and Construction (CON), Structure, Route information. The job then then inserts the Location records with GIS and Non-GIS data to R_FMIS_PROJ_EXT under the Project Header.

Input

Please refer to the chain job.

Batch Parameters

The following are the delivered parameter values, which may have been updated through Batch Setup to meet local needs.

Parameter	Description	Default Value
FHWA Budget Department Code (FHWA_DEPT_CD)	Required Department Code used to select Budget transaction lines. Enter a valid code as this value is not validated.	(No Default)
State ID Number (FHWA_STATE_ID)	Required State ID used to update the State Id field at the Location Level (GIS or non-GIS). Typically, the Recipient ID and State ID are the same. Enter a valid value, as this value is not validated.	(No Default)
Program Roll-up (PROG_RLLP_LVL)	Required. Required rollup used to define the FHWA Project	(No Default)

	Classifications of Preliminary Engineering (PE) and Preliminary Engineering for Structures (PES). Valid values are 1 – Code, 2 – Category, 3 – Group, 4 – Type, and 5 – Class.	
Program Roll-up Value (PROG_RLLP_VALS)	Required: Required rollup values used to define the FHWA Project classification. Specify the value for PE first followed by the value for PES then the value for CON. Must have three values and be separated by colon (:). Cannot have more than 3 values.	(No Default)
Exclude Project Agreement Number (PROJ_AGMT_ERROR_CD)	Required Project Agreement Number that should be excluded from the selection of Budget transaction lines. This number is used to compare with the Project Agreement Number on the Program table. Enter valid code(s) as the value(s) are not validated.	BYPASS

Output

- FMIS Project Extract Table (R_FMIS_PROJ_EXT)

Problem Resolution

The following table shows the potential job return codes for this job.

Job Return Codes

If this job does not finish successfully, there is no restarting. A new instance of the chain should be submitted after addressing the reason(s) for failure.

Return Code	Condition
Successful (1)	Parameter validation is successful and all applicable GIS and Non-GIS data is built based on detail records in R_FMIS_PROJ_EXT.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for unexpected situations. When this job fails, subsequent jobs in the chain are set to <i>Inactive</i> .
Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to <i>Inactive</i> .

Return Code	Condition
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to <i>Inactive</i> .

Sort Criteria

NA

Selection Criteria

This job selects Detail entries for R_FMIS_PROJ_EXT and GIS Location data for several permutations of PE, PES, and CON, as well as whether the data has route IDs, structures, and so forth.

FHWA FMIS Extract Process: Generate FHWA FMIS Extract File

Job Name	Generate FHWA FMIS Extract File
Recommended Frequency	As needed
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This job generates the XML file for transmission to FHWA and marks the processed records as executed on the FMIS_PROJ_EXT table. The job also generates a Text file which will be used for applying approvals to the Project Budget transactions.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> The job begins by validating parameters. When a required parameter is not specified the following message is issued: "Parameter <Name> is required" "Batch parameters validation failed." Transaction Phase code and Approval Level should have numeric values if not, the following message is issued: "Parameter <Name> should be numeric" If the number of characters in User Defined field parameter exceeds 4000 the following message is issued: "Parameter <Name> is empty or has a length > 4000 characters" The job will not use User defined field parameter if the value is not according to the format containing ',' and ':' Example: :FIELD_NM,FIELD_VAL:FLD2_NM,FLD2_VAL: When Parameters are validated successfully the following

Process Steps	Messages
	<p>message is issued:</p> <p>“Job Parameters validated successfully”</p>
<p>2. FMIS file creation</p>	<ul style="list-style-type: none"> • In this step the job determines whether there are records to write to XML file, by taking a count of count of records in R_FMIS_PROJ_EXT where Project Status Indicator is P and DET_REC_NO is 0 and LOC_REC_NO. If there are no records the following message will be issued: “No data to be sent to FMIS – Exiting” • When there are project records, the job generates the XML with a file name of “FIN_FMIS_<Recipient ID>_<timestamp:yyyymmdd_hhmmss>.xml • The file is placed under ARCHEXPORT parameter location. • If file path is invalid then the job will issue the message “Unexpected error while preparing XML file format”. • If file path is valid then the process will start writing Header, Detail and Location data from R_FMIS_PROJ_EXT to the XML file after issuing the message: “XML file successfully initialized” • For each Header record the process issues the following message: <ul style="list-style-type: none"> ○ “Writing a new Header record” • At the end of the process the following messages are logged: <ul style="list-style-type: none"> ○ “Total <n> header records added to XML.” ○ “XML file is successfully generated” • If there are any issues the job will issue the error: “Error in generating XML file”. • For the records that are processed successfully and written to the XML file the job updates status Indicator to ‘E’ on R_FMIS_PROJ_EXT where the indicator was previously P and will issue the following message: “<n> updated to status ‘E’ in R_FMIS_PROJ_EXT table. • As a last step, the job copies the file from ARCHEXPORT to AMSEXPORT and logs the following message if there are any errors: “Unexpected errors when copying the file content from the source to the destination” “Error in Archive XML file”
<p>3. Generate Approval Parameter File for workflow</p>	<ul style="list-style-type: none"> • The process identifies the budget transaction by looking up to BG_DOC_HDR and Worklist tables. • The process generates a .TXT file for approving the Project Budget Transactions with details like User ID, Approval Level, Approver ID, Approval Role, Apply Overrides from job parameter defaulting an Action Code of DOCAPPROVE and fetching other values like Transaction Code,

Process Steps	Messages
	<p>Department, ID and Version number from processed records.</p> <ul style="list-style-type: none"> In case of failure the process logs the following message: "Job Processing Failed"

Input

Please refer to the chain job.

Batch Parameters

The following are the delivered parameter values, which may have been updated through Batch Setup to meet local needs.

Parameter	Description	Default Value
Export File Location (AMSEXPORT)	Required location where the XML File will be placed for transmission to FHWA.	\$\$AMSR00T\$\$/ExportImport
AMSPARM Directory (AMSPARM)	Required file location where the parameter file for Budget Transaction approval will be written.	\$\$AMSPARMS\$\$
Approval Override Indicator (APPLY_OVERRIDES)	Required Approval Overrides Indicator that will be populated in the parameter file for SMU Transaction Workflow Approval. Valid values are true or false.	False
Approval User ID (APRV_ID)	Required Approval User ID that will be populated in the parameter file for SMU Transaction Workflow Approval. Please specify a valid User ID since this will not be validated.	(No Default)
Approval Level (APRV_LVL)	Required Approval Level that will be populated in the parameter file for	(No Default)

	SMU Transaction Workflow Approval. Valid values are between 1-15.	
Approval Role (APRV_ROLE)	Required Approval Role that will be populated in the parameter file for SMU Transaction Workflow Approval. Please specify a valid Approval Role ID since this will not be validated.	(No Default)
Archive Export File Location (ARCHEXPORT)	Required to specify the Archive File Location where FHWA related XML File will initially be created.	\$\$AMSEXPORT\$\$/FMIS/Arch_Out_Files_To_FMIS
Chain Job ID (CHAIN_JOB_ID)	Protected Field to suffix the Chain Job ID to file names of the generated XML files.	\$\$@CHAINJOBID\$\$
Budget Transaction Phase Code (DOC_PHASE_CD)	Required Phase of the Budget transaction to be selected. Must be numeric. Valid values are 2 (Pending) and 3 (Submitted).	2
FHWA Budget Transaction Code (FHWA_DOC_CD)	Required Transaction Code used to select Budget transactions. Enter a valid code as this value is not validated.	(No Default)
FHWA Budget Department Code (FHWA_DOC_DEPT_CD)	Required Department Code used to select Budget transaction lines. Enter a valid code as this value is not validated.	(No Default)
FHWA Recipient ID (FHWA_RCPT_ID)	Required Recipient ID used to select records from	(No Default)

	Project Header table. It is also as part of the file naming convention and updates a field in the FMIS file called RecipientId.	
FHWA State ID (FHWA_STATE_ID)	Required State ID in FHWA used to update the StateId at the Location level (GIS or non-GIS) records.	(No Default)
FHWA User Defined Fields (FHWA_USR_DEF_FLDS)	Required list of fields that will be used in conjunction with the FMIS Project Extract table to build the FMIS extract XML file that is sent to FHWA. It is a two-dimensional array; stored in a linear parameter as a comma-separated list of the defined data elements, which themselves are differentiated by another separator, for instance a colon ":". The length cannot exceed 255 characters. Please refer to the Cost Accounting User Guide for examples.	(No Default)
User ID (USER_ID)	Required User ID that will be populated in the parameter XML file for SMU Transaction Workflow Approval.	(No Default)

Output

- FMIS Project Extract Table (R_FMIS_PROJ_EXT)
- XML file for transmission to FMIS

- Approval of Project Budget Transactions

Problem Resolution

If this job does not finish successfully, there is no restarting. A new instance of the chain should be submitted after addressing the reason(s) for failure.

Note: If FMIS Extract process creates a 0 byte XML file due to issues during the Generate FHWA FMIS Extract File such file should be manually deleted from the file location before it is sent out to FHWA either manually or through interfaces.

Job Return Codes

The following table shows the potential job return codes for this job.

Return Code	Condition
Successful (1)	All project data was added in the XML file correctly.
Failed (12)	<p>The job fails under the following conditions:</p> <ul style="list-style-type: none"> • Parameters are invalid. • Error in generating XML output file • Error in updating R_FMIS_PROJ_EXT table records to 'E' (Exported) • Error in archiving XML file • Run time exceptions for unexpected situations <p>When this job fails, subsequent jobs in the chain are set to <i>Inactive</i>.</p>
Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to <i>Inactive</i> .
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to <i>Inactive</i> .

Sort Criteria

NA

Selection Criteria

This job selects applicable entries from R_FMIS_PROJ_EXT to be entered in the XML file.

FHWA FMIS Extract Process: Apply Workflow Approval for Budget Transaction

Job Name	Apply Workflow Approval for Budget Transaction
Recommended Frequency	As needed
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This job approves the Budget Transactions through System Maintenance Utility using the parameter TXT file created by the previous.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> The process begins by validating the batch parameter. When the parameter is not specified the following message is issued: "Parameter <Name> is empty" "One or more job parameters are empty." The parameter is valid or invalid depending on the Validation. If the parameter is invalid, the following message will be issued: "<filename> file does not exist at specified location".
2. Process Approvals	<ul style="list-style-type: none"> The job selects budget transactions and applies approval using the Parameter file.

Input

Please refer to the chain job.

Batch Parameters

The following are the delivered parameter values, which may have been updated through Batch Setup to meet local needs.

Parameter	Description	Default Value
Parameter File Name (PARAM_FILE)	Parameter file to Load Transactions	\$\$AMSPARM\$\$/L oadADParm.txt

Output

- Approval of Project Budget Transactions

Problem Resolution

If this job does not finish successfully, there is no restarting. A new instance of the chain should be submitted after addressing the reason(s) for failure.

Job Return Codes

The following table shows the potential job return codes for this job.

Return Code	Condition
Successful (1)	Parameter validation is successful and at least one record was selected and processed. Alternatively: All transactions in the XML file were loaded successfully.
Warning (4)	No eligible records found in selection. This could be because of

Return Code	Condition
	the following reasons: <ul style="list-style-type: none"> • No new records since last run • No records matched selection criteria
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for unexpected situations.} When this job fails, subsequent jobs in the chain are set to <i>Inactive</i> .
Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to <i>Inactive</i> .
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to <i>Inactive</i> .

Sort Criteria

NA

Selection Criteria

This job selects GIS Location data for several permutations of PE, PES, and CON, as well as whether the data has route IDs, structures, and so forth.

2.1.11 FMIS Load Process

Chain Job Name	FMIS Load Process
Recommended Frequency	As needed
Single Instance Required	Yes
Can be restarted?	No
Reports generated	None

Overview

The State Department of Transportation (DoT) sends Project details to Federal Highway Authority (FHWA) for approval of project requests and create obligation request for the project amount. The request can be sent in an XML format through Electronic Data Sharing (EDS) to FHWA's Financial Management and Information System (FMIS). For each project, the XML contains header, detail, and location details. FHWA in-turn communicates its response to the project request through another XML format, which has most of the details originally sent by the DoT.

The FHWA FMIS Extract process facilitates in generating the XML file needed for sending Project Requests to FHWA in an electronic format. The process starts by selecting the new project budget transactions, which are pending for approval. The process using the project budget details queries for additional project data from numerous set up pages to build details needed like Project details, Funding details, Geospatial Information System (GIS) on the XML, which can create obligation requests in FMIS. The process subsequently moves the Project Budget Transactions in workflow by applying the required approval.

The FMIS Load Process reads the data from XML file sent by FHWA and parses the data to create a parameter file to approve the obligation requests, Budget Transactions, created for Projects in addition to loading/updating the data from file to the back-end tables. The process generates a report summarizing the project information.

This chain has the following jobs:

- [Preprocess XML File](#)
- [Load Input File](#)
- [Apply Workflow Approval for Budget Transaction](#)

Note: Even though the above jobs in the chain can be run individually by disabling other jobs, it is recommended to always run the entire chain.

Major Input

- XML file from FMIS
- Budget Transaction Header (BG_DOC_HDR)

Major Output

- FMIS Project Location - Non-GIS (R_FMIS_PROJ_LOC_NON_GIS)

- FMIS Project Location - GIS (R_FMIS_PROJ_LOC_GIS)
- Project Detail User Specific Fields (R_FMIS_PROJ_DET_USR_FLDS)
- FMIS Project Detail (R_FMIS_PROJ_DET)
- FMIS Project Header (R_FMIS_PROJ_HDR)
- R_FMIS_PROJ_REL_PROJ
- R_FMIS_PROJ_RECIP_PROG
- Project Header User Specific Fields (R_FMIS_PROJ_HDR_USR_FLDS)
- Project Groups (R_FMIS_PROJ_GRP)
- Program (PROG / R_PROG)
- Program Phase (PHPRG / R_PHASE_PROG)
- Approval of Project Budget Transactions

Chain / Job Return Code

The following table shows the potential return codes for the FMIS Load Process. Note that the Chain job will end with the highest return code across all of the jobs.

Return Code	Condition
Successful (1)	All of the jobs end successfully.
Warning (4)	One of the jobs in the chain ends with a Return Code of Warning.
Non-Fatal Error (8)	One of the jobs in the chain ends with a Return Code of Non-Fatal Error.
Failed (12)	One of the jobs in the chain ends with a Return Code of Failed.
Terminated (16)	One of the jobs in the chain ends with a Return Code of Terminated.
System Failure (20)	One of the jobs in the chain ends with a Return Code of System Failure.

Problem Resolution

This job cannot be restarted. If the job fails for some reason, then the reason for failure should be verified and the new job should be scheduled. The new job is scheduled irrespective of its previous jobs that failed.

FMIS Load Process: Preprocess XML File

Job Name	Preprocess XML File
Recommended Frequency	See FMIS Load Process Information
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This job reads the data from the XML file sent of FHWA, initializes variables, parses data from the XML file and generates new XML file for loading data to FHWA data tables.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> The job begins by validating parameters. When a required parameter is not specified, the following messages get logged: "Parameter <Name> is empty" "One or more job parameters are empty." Parameters are valid or invalid depending on the validation. If the parameter is invalid, the invalid value will be displayed in the log. Job Parameters validated successfully.
2. Selection of XML files	<ul style="list-style-type: none"> The job selects eligible FHWA XML files for processing. If the selection returns 0 files, the job will end and following message will be issued: "No W41 files found". Each file that is selected will be listed as processing in the job log.
3. Generate XML files for upload	<ul style="list-style-type: none"> Job processes eligible FHWA XML files and generates XML files with pps prefix. The following message is issued in the job log for each FHWA FMIS file that is generated: "Initialized Preprocessed Output file" Run Ended

Input

- XML file from FHWA

Batch Parameters

The following are the delivered parameter values, which may have been updated through Batch Setup to meet local needs.

Parameter	Description	Default Value
Import File Location	Required location of FMIS W41 W43 xml	\$\$AMSEXPORT\$\$

(AMSIMPORT)	files.	
Archive Import File Location (ARCHIMPORT)	Required file location for archived FMIS W41 W43 files.	\$\$AMSEXPORT\$\$
Chain Job ID (CHAIN_JOB_ID)	Required Chain Job ID of the current job used in the name of the processed FMIS W41 W43 files.	\$\$@CHAINJOBID@\$\$
Primary Import File Name Prefix (IMPORT_FILE_NM_PRE1)	Required prefix of FMIS W41 W43 files. The job looks for this prefix first in the name of the file.	(No Default)
Secondary Import File Name Prefix (IMPORT_FILE_NM_PRE2)	Required secondary prefix of FMIS W41 W43 files. The job looks for this prefix after the Primary Import File Name Prefix.	(No Default)
Progression Counter Size (PROG_CTR_SZ)	Required field. During processing, the job writes messages to the log to report on its progress based on the number of records already processed. When this parameter is specified, the value controls the interval at which these progression messages are written to the job log. A specified value should be a positive integer.	5000
Number of Records Flushed to File (NUM_RCRDS_FLUSH_TO_FILE)	Number of records written from the buffer to the xml file.	1000

Output

- XMLs file for data load to FMIS tables

Problem Resolution

If this job does not finish successfully, there is no restarting. A new instance of the chain should be submitted after addressing the reason(s) for failure.

Job Return Codes

The following table shows the potential job return codes for this job.

Return Code	Condition
Successful (1)	The job ends as successful when all the input XML files are processed successfully.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> Parameters are invalid. If FMIS_CODE_SET_ID parameter doesn't match APPCTRL set up. Run time exceptions for unexpected situations. When this job fails, subsequent jobs in the chain are set to <i>Inactive</i> .
Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to <i>Inactive</i> .
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to <i>Inactive</i> .

Sort Criteria

None

Selection Criteria

This job will select XML files with the prefix matching, which is entered on the Primary and Secondary Import File Name Prefix parameters.

FMIS Load Process: Load Input File

Job Name	Load Input File
Recommended Frequency	See FMIS Load Process Information
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This job loads the XML file to update data to FHWA data tables. The process also generates a TXT file to approve Project Budget transactions and updates Program (PROG) records.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> The process begins by validating parameters. When a required parameter is not specified, the following messages get logged: "Parameter <Name> is empty" "One or more job parameters are empty." Parameters are valid or invalid depending on the Validation. If

Process Steps	Messages
	<p>the parameter is invalid, the invalid value will be displayed in the log.</p> <ul style="list-style-type: none"> • Job Parameters validated successfully.
2. Selection of XML files	<ul style="list-style-type: none"> • The job begins selecting eligible records and issues the following message: “About to process XML Inbound Files” • If the selection returns 0 records, then the following message will be issued: “No files in directory”. • Each file and project that is selected will be listed as processing in the job log: “Processing file: <<Filename>>”.
3. Processing of files	<ul style="list-style-type: none"> • When the file is successfully processed, for the projects selected and imported the following log messages are displayed: “Processing Project #: <<Project number <<count>> “XML File successfully processed for <<count>> projects” “Data for <<count>> projects will be imported”. • A message is logged for each Input XML file that is successfully generated and archived: “Processing of Input XML File - <<Filename>> successful” “<<Filename>> archived successfully” “Completed Archiving Input XML File – <<Filename>>” • A message is issued when all input files are successfully processed: “All input files processed successfully”.
4. Generate TXT file for Approval	<ul style="list-style-type: none"> • For each SMU parameter file that is generated the job issues the following message: “Initialized Output XML file” • When the job begins generating the output file, the following message is issued: “Creating Approval File” • Run Ended

Input

- XML file from job 1 with pps prefix

Batch Parameters

The following are the delivered parameter values which may have been updated through Batch Setup to meet local needs.

Parameter	Description	Default Value
Import File Location	Required	\$\$AMSROOT\$\$/ExportImport

(AMSIMPORT)	location where input XML files are retrieved.	
Parameter directory (AMSPARM)	Required file location where the parameter file will be written.	\$\$AMSROOT\$\$/Parms
Approval Override Indicator. (APPLY_OVERRIDES)	Required. Stipulates whether override will be applied on transaction import and submit. Valid values are <i>True</i> or <i>False</i> .	False
Approval User ID (APRV_ID)	Required. Approval User ID that will be populated in the parameter file for SMU Transaction Workflow Approval.	(No Default)
Approval Role (APRV_ROLE)	Required. Approval Role that will be populated in the parameter file for SMU Transaction Workflow Approval.	(No Default)
Archive Import File Location (ARCHIMPORT).	Required file location for processed FMIS files.	\$\$AMSROOT\$\$/ExportImport
Chain Job ID (CHAIN_JOB_ID)	Required Chain Job ID used to determine which files to select for processing.	\$\$@CHAINJOBID\$\$
FHWA Transaction Code	Required	(No Default)

(FHWA_DOC_CD)	Transaction Code used to select transactions.	
FHWA Transaction Department Code (FHWA_DOC_DEPT_CD)	Required Department Code used to select transactions.	(No Default)
FHWA Transaction Unit Code (FHWA_DOC_UNIT_CD)	Required Transaction Unit used to select Budget transactions. Enter a valid code as it is not validated.	(No Default)
Primary Import File Name Prefix (IMPORT_FILE_NM_PRE1)	A required prefix of the inbound FMIS file name.	(No Default)
Secondary Import File Name Prefix (IMPORT_FILE_NM_PRE2)	Required secondary prefix of the inbound FMIS file name.	(No Default)
User ID (USER_ID)	The User ID that will be populated in the parameter file for SMU Transaction Workflow Approval. The value should be a valid User ID as this parameter is not validated.	(No Default)
Progression Counter Size (PROG_CTR_SZ)	Required. During processing, the job writes messages to the log to report on its progress based on the number of records already	5000

	<p>processed. When this parameter is specified, the value controls the interval at which these progression messages are written to the job log. A specified value should be a positive integer.</p>	
<p>SMU Catalog ID (SMU_CTLG_ID)</p>	<p>Required Catalog ID of the SMU job that is spawned as the child processes. A specified value should be a positive integer.</p>	<p>(No Default)</p>
<p>Sleep Interval in Seconds (SECONDS_SLEEP_INTERVAL)</p>	<p>Required polling frequency (in seconds) for internal controller thread for checking child processes. A specified value should be a positive integer.</p>	<p>(No Default)</p>
<p>Log Status Interval (SECONDS_LOG_STATUS_INTERVAL)</p>	<p>Required logging frequency (in seconds) for controller thread reporting status of child threads to the system log. The specified value should be a positive integer.</p>	<p>(No Default)</p>

Commit Block Size (SMU_JOB_COMMIT_BLOCK)	Required field that controls how many records are committed by the application at one time. The size should be compatible with technical capabilities and performance guidelines.	100
Number of times to Sleep (SMU_JOB_NUM_TIMES_SLEEP)	Required field that specifies the number of times the job waits for the internally spawned SMU job to complete. Each time is equal to the Sleep Interval in Seconds. A specified value should be a positive integer.	(No Default)

Output

- FMIS Project Location - Non-GIS (R_FMIS_PROJ_LOC_NON_GIS)
- FMIS Project Location - GIS (R_FMIS_PROJ_LOC_GIS)
- R_FMIS_PROJ_DET_USR_FLDS
- FMIS Project Details (R_FMIS_PROJ_DET)
- FMIS Project Header (R_FMIS_PROJ_HDR)
- R_FMIS_PROJ_REL_PROJ
- R_FMIS_PROJ_RECIP_PROG
- FMIS Project Header User Defined Fields (R_FMIS_PROJ_HDR_USR_FLDS)
- FMIS Project Groups (R_FMIS_PROJ_GRP)
- Program (PROG / R_PROG)
- Program Phase (PHPRG / R_PHASE_PROG)
- TXT file for Approval of Project Budget Transactions

Problem Resolution

If this job does not finish successfully, there is no restarting. A new instance of the chain should be submitted after addressing the reason(s) for failure.

Job Return Codes

The following table shows the potential job return codes for this job.

Return Code	Condition
Successful (1)	The job ends as successful when all the input files are processed successfully.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for unexpected situations. • Problem encountered while connecting Database When this job fails, subsequent jobs in the chain are set to <i>Inactive</i> .
Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to <i>Inactive</i> .
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to <i>Inactive</i> .

Sort Criteria

None

Selection Criteria

This job will select XML files with a pps prefix that were generated in the first chain job step.

FMIS Load Process: Apply Workflow Approval for BGPHR Transactions

Job Name	Apply Workflow Approval for BGPHR Transactions
Recommended Frequency	See FMIS Load Process Information
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This job approves the Budget transactions through System Maintenance Utility using the parameter TXT file created by the previous.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> The process begins by validating the batch parameter. When the parameter is not specified, the following message is issued: "Parameter <Name> is empty" "One or more job parameters are empty." The parameter is valid or invalid depending on the Validation. If the parameter is invalid, the following message will be issued: "<filename> file does not exist at specified location"
2. Selection of XML files	<ul style="list-style-type: none"> The job begins selecting eligible records and issues the following message: "About to process XML Inbound Files" If the selection returns 0 records, then the following message will be issued: "No files in directory". Each file and project that is selected will be listed as processing in the job log: "Processing file: <<Filename>>"
3. Process Approvals	The job selects Budget transactions and applies approval using the Parameter file.

Input

- TXT file from second job

Batch Parameters

The following are the delivered parameter values which may have been updated through Batch Setup to meet local needs.

Parameter	Description	Default Value
Parameter File Name (PARAM_FILE)	Parameter file to Load Transactions	\$\$AMSPARM\$\$/LoadADParm.txt

Output

- Approval of Project Budget Transactions

Problem Resolution

If this job does not finish successfully, there is no restarting. A new instance of the chain should be submitted after addressing the reason(s) for failure.

Job Return Codes

The following table shows the potential job return codes for this job.

Return Code	Condition
Successful (1)	Parameter validation successful and at least one record was selected and processed. Alternatively: All transactions in the XML file were approved successfully.
Warning (4)	No eligible records found in selection. This could be because of the following reasons: <ul style="list-style-type: none"> • No new records since last run • No records matched selection criteria
Non-Fatal Error (8)	Not all records could be successfully processed.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for unexpected situations. When this job fails, subsequent jobs in the chain are set to <i>Inactive</i> .
Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to <i>Inactive</i> .
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to <i>Inactive</i> .

Sort Criteria

None

Selection Criteria

This job will select XML files with a pps prefix that were generated in the first chain job step.

2.1.12 Internal Costing Chain

Chain or Job Name	Internal Costing Chain
Recommended Frequency	The Internal Costing Chain can be run on demand.
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	<p>Yes</p> <ol style="list-style-type: none"> 1. The Internal Cost Update job generates two reports: <ol style="list-style-type: none"> I. Standard Cost Update Exception Report II. Standard Cost Update Statistics Report 2. The Charge Transaction XML Generation job generates two reports: <ol style="list-style-type: none"> I. CHDocXMLGenerator report II. Exception Report 3. The CH Exception Report job generates one report: <ul style="list-style-type: none"> • CHDocExcpRpt report

Overview

The Internal Costing Chain is a chain job in CGI Advantage Financial located in the Cost Accounting / Chain job folder. This chain job is a group of jobs that work together to calculate standard and fixed costs for various usage types, and generate Charge Transaction (CH) transactions to record those costs in the Advantage system. This process will extract the labor, mileage, equipment and materials testing usage data collected on ICT transactions from the Internal Costing Journal (ICJ) table and use the data to generate the CH transactions.

The chain job can be run in either detailed mode or in summary mode depending on the user's requirement. The Internal Costing chain has the following jobs (each of the jobs listed below is described in subsequent sections):

1. Internal Cost Update
2. Charge Transaction XML Generation
3. CH Upload
4. CH Submit
5. CHExcpRpt

The acceptable job return code configuration depends on the business requirement. For example, if the requirement is that the subsequent jobs in the chain should continue only if the job ends with a Return Code of *Successful*, the Acceptable job return codes for all of the jobs should be set to *Successful*. If for some jobs in the chain, a Non Fatal error is an Acceptable job return code, then that can also be configured. These configurations can be done in the Job setup page.

For Baseline configuration, the CH Upload job will be kicked off when the CH XML Generation job ends with a Return Code of *Successful* or *Non Fatal Error*. If the CH XML Generation job ends with a Return Code of *Warning*, the rest of the jobs will be set to *Inactive*. The CH Submit job will be kicked off when the CH Upload job ends with a return code of *Successful*. The CH Exception report job will be kicked off when the CH Submit job ends with a return code of *Successful*.

If any of the jobs in the chain ends with a return code of *Failed*, *Terminated* or *System Failure*, all of the subsequent jobs will be set to *Inactive*.

Major Input

- Internal Cost Parameter (R_INT_CST_PARM) table
- Internal Cost Journal (R_INT_CST_JRNL) table
- Special Accounts (R_INT_CST_SPEC) table
- System Wide Option (R_INT_CST_SOPT) table
- Internal Cost Rate (R_INT_CST_RT) table
- Internal Vendor Accounting (R_INT_VEND_ACTG) table
- Employee Information (R_EMP_INFO) table
- Equipment ID (R_EQUIP) table
- Employee Benefits Multiplier (R_EMPL_BEN_MULTI) table
- Internal Costing Overtime Multipliers (R_INT_CST_OT_MULTI) table

Major Output

- Charge Transactions
- Internal Cost Update Exception Report
- Internal Cost Update Statistics Report
- Charge Transaction XML Generator Report
- Charge Transaction XML Generator Exception Report
- Charge Exception Report
- Internal Cost Journal (R_INT_CST_JRNL) table

Chain / Job Return Code

The following table shows the potential return codes for the Internal Costing chain job. Note that the Chain job will end with the highest return code across all of the jobs.

Return Code	Condition
Successful (1)	All of the jobs end successfully.
Warning (4)	One or more of the jobs in the chain ends with a Return Code of <i>Warning</i>
Non Fatal Error (8)	One or more of the jobs in the chain ends with a Return Code of <i>Non Fatal Error</i> .
Failed (12)	One of the jobs in the chain ends with a Return Code of <i>Failed</i> .
Terminated (16)	One of the jobs in the chain ends with a Return Code of <i>Terminated</i> .

System Failure (20)	One of the jobs in the chain ends with a Return Code of <i>System Failure</i> .
---------------------	---

Problem Resolution

All of the jobs in the chain do not have the restart capability. Only jobs driven by SysManUtility have the restart capability. Please refer to the individual jobs for details regarding the specific job processes and problem resolution.

Internal Costing Chain: Internal Cost Update job

Job Name	Internal Cost Update Job
Recommended Frequency	This job can be run daily as part of the nightly cycle or on demand.
Single Instance Required	Yes.
Can be restarted?	No.
Reports Generated	Yes.

Overview

The Internal Cost Update process selects records on the Internal Costing Journal (ICJ) table to calculate costs and store those values on ICJ for the Charge Transaction XML Generation Process. When updating internal costs, this process selects those ICJ records that meet the selection criteria and for which internal costs have not previously been computed.

For every record that has been processed and for which the standard cost fields are updated, the Internal Cost Process Date field is updated on ICJ with the Application Control Date of the batch process. If the ICJ record has been processed through the Internal Cost Update process, no modifications can be performed to it using an ICT transaction.

If rates have been entered incorrectly and the standard (or fixed) cost amounts need to be calculated again, the user may enter an adjusting ICT entry (negative amounts are allowed) to reverse the original entry and then enter a new ICT transaction using the correct rate.

Calculation of Labor & Employee Benefit Costs

The Standard Costing process uses the Regular Labor/Usage Units on the ICJ record and the Salary rate for the Employee ID (if the employee rate is specified on EMPID) or else the matching Employee Class from the Internal Cost Rate (INCR) table to compute the Standard Salary. Overtime salary is calculated by multiplying the Overtime Labor Units, the Salary rate, and the Overhead Multiplier Rate from the Internal Costing Overhead Multiplier (ICOM), if the Overtime Multiplier Code is populated on the ICJ record. The rate is associated with the Overtime Multiplier Code on the Internal Cost Journal (ICJ).

Standard Salary Calculation

Prior to this enhancement, Salary Cost was computed by multiplying the number of Regular Labor/Usage Units by the Salary Rate for the associated Employee Class. The Employee Class was already present on ICJ. However, the order of processing is now changed as follows:

- Internal Costing job will first perform a lookup to EMPID to determine if a non-zero Salary Rate has been defined for the employee.

- Once one or more Employee ID records are found, the following processing logic will take place:
 - If non-zero rates are defined at the Employee ID level, those rates will be used instead of the INCR Employee Class Rates.
 - Effective dating of the employee records means that the Salary may be computed using rates from multiple records on EMPID. The Usage From and To dates from ICJ will be compared to the effective dates on EMPID.
- The INCR rates will still apply for Standard Salary in the following cases:
 - Where the rate defined for the first matching record on EMPID is zero
 - Where the Validate Employee ID on EMPID flag is unchecked on System Options (SOPT).

The logic is outlined in more detail in the “Detailed Processing Steps” section.

Overtime Salary Calculation

The logic for Overtime Salary Calculation is very similar to that described above for Standard Salary. The main difference is the use of the Overtime Multiplier Rate.

- Instead of Regular/Labor Usage Units, Overtime Units is used in the calculation. This is not a change.
- Salary Rates will be determined using the same logic as described above for Standard Salary.
- However, if an Overtime Multiplier Code has been defined on the ICJ record, then a lookup to ICOM is performed to retrieve the Overtime Multiplier Rate. Each Salary Rate used in the calculation of Overtime Salary is multiplied by the Overtime Multiplier Rate.

Employee Benefits

For labor costs, up to five employee benefit multipliers may be defined for each usage class. Rates may be defined on INCR where those rates are FY-based. Alternatively, the employee benefits multipliers may be effective dated through a separate table “Employee Benefits Multiplier (EMPBM)”. The Get Rates from Employee Benefits Multiplier flag on the SOPT table controls the usage of the Employee Benefits Multiplier (EMPBM) table. If EMPBM is not used, then the INCR rates are used instead.

Ten Employee Benefit Costs need to be calculated as part of the enhanced labor costing mechanism. Employee Benefits 1-5 need to be computed for Standard Salary and Employee Benefits 1-5 also need to be computed for Overtime Salary. Not all benefits may apply. To calculate the Standard Employee Benefits, the process takes the Standard Salary value on the ICJ record and multiplies that amount by all of the appropriate Employee Benefits Multiplier rates from the Employee Benefits Multiplier (EMPBM) table or the Internal Cost Rate (INCR) table. Similar logic applies for the Overtime Employee benefits, except the previously calculated Overtime cost is used instead of Standard Salary in the calculation. As the EMPBM page is effective dated, it is possible that multiple rates may apply for the same benefit for the same ICJ record. The logic for effective dating is similar to that described above for the initial costing of Standard Salary and Overtime Salary. However, as the effective dates for salary rates may differ from employee benefits, the lookup for employee benefit rates is to EMPBM. Note that Standard Salary and Overtime Salary must be computed before the 10 Employee Benefit values can be computed.

The actual Employee Benefits that will be calculated depend on the following settings:

- The Employee Benefits Multiplier (1, 2, or All) was indicated during entry of the ICT transaction and is stored on the ICJ record. This setting may exclude one or more benefits from the calculations.
- Benefits multipliers can apply to Standard Salary only, Overtime only, or to both Standard and Overtime. The application of benefit multipliers is defined by Employee Class on INCR.
- Benefits Multipliers can apply to Inter COA transactions, Intra COA transactions or all transactions.

The Detailed Processing Steps will cover the computations in more detail.

The new Total Cost field will store the total of all amount fields. Employee records will not update the Standard Cost or Fixed Cost fields. For Labor, the Total Cost will be the sum of Standard Salary, Overtime Salary and the 10 Employee Benefits.

Fixed and Standard Rates for Equipment Costing

The Internal Cost process calculates the cost for Equipment records from the ICJ table by multiplying the 'Regular Labor/Usage Units' value on ICJ table by the 'Equipment Usage Rate' from either the associated Equipment record on the Equipment table or the associated Equipment Class record on the Internal Cost Rate table. If the Rate Category field on the Internal Cost Journal is set to *Standard* for the record being processed, then the calculated cost for the Equipment record is populated in the Standard Cost field on the ICJ table. If the Rate Category field on the Internal Cost Journal is set to *Fixed* for the record being processed, then the Fixed Cost field on the ICJ table will be populated with the calculated cost. An optional Equipment Surcharge cost may also be calculated. Rates for the surcharge, both fixed and standard, may be established on the Equipment or Internal Cost Rate tables.

If an Equipment or Internal Cost Rate table entry cannot be found for the Equipment Class, the record will be left uncOSTed and appear on the exception report. The user needs to enter the rate on the Internal Cost Rate table. Once this is done, the record will be costed correctly in the next costing run.

Materials Testing

The costs for the Materials Testing records are calculated by multiplying the 'Materials Testing Rate' value from Internal Cost Rate table by the 'Regular Labor/Usage Units' value on the ICJ table. This amount is stored in the Standard Cost field on the ICJ record.

If an Internal Cost Rate table entry cannot be found for the Materials Testing Class, the record will be left un-costed and appear on the Exception report. Rates need to be established on the Internal Cost Rate (INCR) table. Once this is done, the record will be processed correctly in the next costing run.

Mileage Costing

Mileage costs will be calculated for Mileage records on ICJ by multiplying the rate entered in the Mileage Rate field on the Internal Cost tab of the System Options (SOPT) table by the 'Regular Labor/Usage Units' value on the ICJ table.

If a Systems Option (SOPT) table entry cannot be found for the Fiscal Year, the record will not be processed and will appear on the exception report. The user needs to enter the mileage rate on the System Options table. Once this is done, the record will be costed correctly in the next costing run.

Steps in Running this Process

1. Select ICJ records
 - a. The process will select records from the Internal Costing Journal (ICJ) table where the Internal Cost Process Date is null.
 - b. Additional ICP parameters can be applied as record selection criteria.
 - i. If the Usage From is specified on the selected ICP record, the ICJ records that have a Usage From date greater than or equal to the ICP Usage From date will be selected for processing. Or, if the Usage From on the ICP record is blank, this field does not impact selection.
 - ii. If the Usage To is specified on the selected ICP record, the ICJ records that has a Usage To date less than or equal to the ICP Usage To date will be selected for processing. Or, if the Usage To on the ICP record is blank, this field does not impact selection.
 - iii. If the Selection Accounting Period, Selection Fiscal Year, or Selection Budget Fiscal Year is specified on the selected ICP record, the ICJ records with a matching Accounting Period, Fiscal Year, and/or Budget Fiscal Year will be selected for processing. Or, if these fields are blank on the ICP record, this field does not impact selection.
 - iv. If the Selection Department is specified on the selected ICP record, the ICJ records with a matching Department will be selected for processing. Or, if the Department on the ICP record is blank, this field does not impact selection. Note that it is the Department field under Detail Accounting on ICJ that is compared to the ICP Selection Department, and not the Usage Class Department or the Equipment/Employee Home Department located under Usage Information.
 - c. If no records are selected, the process will end and a record will be written into the Exception Report to indicate that no records were selected for processing, and continue to subsequent jobs within the chain job. The job will end with a return code of *Warning*.
2. Data Sorting
 - a. If records are selected for processing, then the selected records will be ordered by Fiscal Year, Usage Type, Department, Rate Category, Usage Class Department, and Usage Class (Equipment Class, Employee Class or Materials Testing Class).
3. Rate and Other Data Retrieval from INCR and SOPT
 - a. If the Usage Group is Labor, Materials Testing, Equipment (ID Required), or Equipment (ID Optional) then look up the Employee Class, Equipment Class or Materials Testing Class on the Internal Cost Rate table for the specified Fiscal Year, Rate Category, and Usage Class Department . The ICJ record's Usage Class Department is used instead of Department for the INCR lookup.
 - i. Perform the lookup on the Internal Cost Rate table first with the Fiscal Year, Rate Category, Usage Class, and Usage Class Department from the ICJ record. If the record is found:
 - If the Usage Group on ICJ is *Labor*, get the Salary Rate, IVAD Vendor Code, all five of the Employee Benefits Multipliers, and each Multiplier's Standard/Overtime Indicator and Inter/Intra-COA Indicator.

- If the Usage Group is *Equipment (ID Required)* or *Equipment (ID Optional)*, get the Equipment Usage Rate and Equipment Surcharge Rate.
 - If the Usage Group is *Materials Testing*, get the Materials Testing Rate.
- b. If the Usage Group is Mileage, look up the Mileage Rate on the System Options (SOPT) table for the ICJ record's Fiscal Year. There will be only one mileage rate applicable to all ICJ records per Fiscal Year.
- i. Use the mileage rate saved on the SOPT table for the record associated with the Fiscal Year of the ICJ record.

4. Calculate Costs

a. Usage Group = Labor

Salary, Overtime and Employee Benefit costs are calculated for this Usage Group. The base Salary and Overtime cost must be calculated before the associated Employee Benefit costs can be calculated. The first step in the process is to calculate Standard Salary.

Standard Salary Calculations

Regular/Labor Usage Units from the ICJ record is multiplied by the appropriate Salary rate. The appropriate Salary rate is derived either from the EMPID or INCR table. Lookup to EMPID is based on the combination of Employee ID and Equipment/Employee Home Department (the latter is only included in the lookup if Permit Duplicate Equipment/Employee IDs Across Departments is checked on SOPT) from the ICJ record. The lookup to INCR was described in Step 3 above.

- i. When the Employee ID table is not used or the rate(s) found for the employee on EMPID are set to zero, then the Salary Rate for the Employee Class on INCR is used.
- ii. When a single matching record with no Effective Start or End Date exists on EMPID, then that record's Salary Rate is used, unless it is zero. Otherwise, the applicable INCR rate will be used.
- iii. When more than one matching EMPID record is found, but the Usage From and To Dates from the ICJ record align with a single EMPID record's Effective From and Effective To Dates, then that record's Salary Rate is used, unless it is zero. Otherwise, the applicable INCR rate will be used. This logic applies in the following scenarios:
 - If the ICJ Usage From Date is greater than or equal to the EMPID Effective Start Date and ICJ Usage To Date is less than or equal to EMPID Effective End Date.
 - If the ICJ Usage From Date is greater than or equal to the EMPID Effective Start Date and EMPID Effective End Date is blank.
 - If the ICJ Usage To Date is less than or equal to the EMPID Effective End Date and EMPID Effective Start Date is blank.
- iv. When more than one matching EMPID record is found, and the ICJ Usage From and To Dates dictate that the Salary Rates from multiple EMPID records must be used. This occurs when the ICJ Usage From Date aligns with one EMPID record but the ICJ Usage To Date aligns with an EMPID record for a later date range. Where this is the case, the

rates from all applicable EMPID records are used, and salary calculation is pro-rated based on the number of usage days applicable for each EMPID record. Partial Standard Salary amounts are calculated for each applicable EMPID record. Once all partial amounts have been calculated, those amounts are added to generate the final Standard Salary amount. The processing logic is as follows:

- First, determine the total number of days between the ICJ Usage From and Usage To dates. Call this “Total Usage Days”. This is calculated as (Usage To Date minus Usage From Date) + 1.
- Calculate the number of days for each applicable EMPID record. This is called “Days at Current Rate”.
 - For the earliest applicable EMPID record, Days at Current Rate = (EMPID Effective End Date minus ICJ Usage From Date) + 1.
 - Where the entire range of an EMPID record falls within the Usage Date range of the ICJ record, Days at Current Rate = (EMPID Effective End Date minus EMPID Effective Start Date) + 1
 - For the latest applicable EMPID record, Days at Current Rate = (ICJ Usage To Date minus EMPID Effective Start Date) + 1
- Each Partial Standard Salary amount is calculated using the following formula: EMPID Salary Rate x Regular Labor Usage Units x (Days at Current Rate/ Total Usage Days)

Overtime Calculations

Overtime Cost Calculation is similar to Standard Salary. However, there are some important differences for any Overtime Cost calculations:

- i. Instead of Regular/Labor Usage Units, Overtime Units is used in the equation.
- ii. Salary rate is retrieved and calculated in the same way as described previously for the Standard Salary calculation. When there are multiple applicable EMPID records for a single ICJ record due to effective dating of rates, partial Overtime Cost calculations are necessary using the same methodology described for Standard Salary.
- iii. Finally, an Overtime Multiplier is applied after (i) has been multiplied by (ii). The Overtime Multiplier is retrieved from ICOM using the Overtime Multiplier code from the ICJ record.

Employee Benefit Calculations

Once the Standard Salary and Overtime Cost have been calculated, the associated Employee Benefit costs can then be calculated. The source of the rates is based on the setting for **Get Rates from Employee Benefits Multiplier** on SOPT. If this is checked, rates are retrieved from EMPBM. If unchecked, rates are retrieved from INCR. The former is effective-dated, resulting in the possibility that multiple rates may apply in the calculation of a single employee benefit. The latter is FY-based, and only a single rate will ever apply in the calculation of an employee benefit amount. However, before rates are applied, several other factors must be considered before a benefit amount is calculated.

- i. Benefit Multiplier Type: This value is either defaulted or entered on the ICT transaction and then populated on ICJ.
 - *Apply All Multipliers*: The process will attempt to calculate the Standard/Overtime Employee Benefit amounts for Employee Multipliers 1-5.
 - *Benefits Multiplier 1*: Only Standard/Overtime Employee Benefit 1 amounts will be calculated
 - *Benefits Multiplier 2*: Only Standard/Overtime Employee Benefit 2 amounts will be calculated
- ii. For each benefit multiplier, the INCR setting for the Standard/Overtime Indicator is considered.
 - If set to *Standard Salary Only*, then only the Standard Employee Benefit amount will be calculated for the applicable Benefit Multipliers as described in (i) above.
 - If set to *Overtime Only*, then only Overtime Employee Benefit amount will be calculated for the applicable Benefit Multipliers as described in (i) above.
 - If set to *Standard and Overtime*, then both the Standard and Overtime Employee Benefit amounts are calculated for the applicable Benefit Multipliers as described in (i) above.
- iii. Finally, the Inter/Intra-COA indicator for each benefit multiplier is considered. The determination of whether an ICJ record is considered Inter or Intra-COA is based on a comparison between buyer and seller accounting lines. The exact basis for comparison is defined in the Inter-COA Definition field on IVAD. When there is a difference between the buyer and seller for the combination of COA elements defined on IVAD, then the ICJ record is considered Inter-COA Usage. Otherwise, the record will be considered Intra-COA Usage. The COA elements for the buyer accounting line are found on the ICJ record and the COA elements for the seller accounting line are found on either IVAD or EMPID. IVAD will always determine the level of comparison to be performed.
 - If set to *Applies to All Usage*, then the applicable Standard/Overtime Employee Benefit amount will be calculated as long as other considerations are met as described in (i) and (ii) above.
 - If set to *Applies Only to Inter-COA Usage*, then the applicable Standard/Overtime Employee Benefit amount will only be calculated in an Inter-COA scenario and as long as other considerations are met as described in (i) and (ii) above.
 - If set to *Applies Only to Intra-COA Usage*, then the applicable Standard/Overtime Employee Benefit amount will only be calculated in an Intra-COA scenario and as long as other considerations are met as described in (i) and (ii) above.

Once it is known that a Standard and/or Overtime Employee Benefit has to be calculated based on settings for (i), (ii) and (iii) described above, the actual calculation is as follows:

- i. When the INCR rate is to be applied, then Standard Salary/Overtime Employee Benefit Amount = Standard Salary/Overtime amount * applicable Employee Benefits Rate on INCR
 - ii. When the EMPBM rate(s) is/are to be applied, then the methodology for effective dating described for Standard Salary calculation will apply:
 - Where a single EMPBM record can be applied to the ICJ record's Usage Dates, the calculation is similar to that described above when INCR rates are used: Standard Salary/Overtime Employee Benefit Amount = Standard Salary/Overtime amount * applicable Employee Benefits Rate on EMPBM
 - Where ICJ Usage Dates do not align with a single EMPBM record and the date range results in rates from multiple EMPBM records being applied, then Partial Employee Benefit amounts must be calculated for each applicable EMPBM record. The logic for calculation of the partial amounts is similar to that described for calculation of partial salary amounts.
- b. Usage Groups = Equipment (ID Optional) and Equipment (ID Required)
 Three separate cost amounts may be calculated: Standard Cost, Fixed Cost and Equipment Surcharge. For a single ICJ record, either Standard Cost or Fixed Cost amount will be calculated, but not both. Where Rate Category = *Standard* for the ICJ record, the Standard Cost is calculated and where Rate Category = *Fixed*, the Fixed Cost is calculated. Equipment Surcharge Cost may apply to records to both Rate Categories.

Standard Cost

Regular/Labor Usage Units from the ICJ record is multiplied by the appropriate Equipment Rate. The appropriate Equipment rate is derived either from the EQID or INCR table. Lookup to EMPID is based on the combination of Employee ID and Equipment/Employee Home Department (the latter is only included in the lookup if Permit Duplicate Equipment/Employee IDs Across Departments is checked on SOPT) from the ICJ record. The lookup from the ICJ record to INCR is described in Step 3 above and does include Rate Category – for Standard Cost calculation, the matching INCR record will have a Rate Category of *Standard*.

- i. When the Equipment ID table is not used or the rate found for the employee on EQID for the first matching record is set to zero, then the Equipment Usage Rate for the Equipment Class on INCR is used.
- ii. When a single matching record with no Effective Start or End Date exists on EQID, then that record's Equipment Rate – Standard is used, unless it is zero. Otherwise, the applicable INCR rate will be used.
- iii. When more than one matching EQID record is found, but the Usage From and To Dates from the ICJ record align with a single EQID record's Effective From and Effective To Dates, then that record's Equipment Rate - Standard is used, unless it is zero. Otherwise, the applicable INCR rate will be used. This logic applies in the following scenarios:
 - If ICJ Usage From Date is greater than or equal to the EQID Effective Start Date and ICJ Usage To Date is less than or equal to EQID Effective End Date.
 - If ICJ Usage From Date is greater than or equal to the EQID Effective Start Date and EQID Effective End Date is blank.

- If ICJ Usage To Date is less than or equal to the EQID Effective End Date and EQID Effective Start Date is blank
- iv. When more than one matching EQID record is found, and the ICJ Usage From and To Dates dictate that the Equipment Rates from multiple EQID records must be used. This occurs when the ICJ Usage From Date aligns with one EQID record but the ICJ Usage To Date aligns with an EQID record for a later date range. Where this is the case, the rates from all applicable EQID records are used, and Standard Cost calculation is pro-rated based on the number of usage days applicable for each EQID record. Partial Standard Cost amounts are calculated for each applicable EQID record. Once all partial amounts have been calculated, those amounts are added to generate the final Standard Cost amount. The processing logic is as follows:
 - First, determine the total number of days between the ICJ Usage From and Usage To dates. Call this “Total Usage Days”. This is calculated as (Usage To Date minus Usage From Date) + 1.
 - Calculate the number of days for each applicable EQID record. This is called “Days at Current Rate”.
 - For the earliest applicable EQID record, Days at Current Rate = (EQID Effective End Date minus ICJ Usage From Date) + 1.
 - Where the entire range of an EQID record falls within the Usage Date range of the ICJ record, Days at Current Rate = (EQID Effective End Date minus EQID Effective Start Date) + 1
 - For the latest applicable EQID record, Days at Current Rate = (ICJ Usage To Date minus EQID Effective Start Date) + 1
 - Each Partial Standard Cost amount is calculated using the following formula: EQID Equipment Rate - Standard x Regular Labor Usage Units x (Days at Current Rate/ Total Usage Days)

Fixed Cost

The process for Fixed Cost calculations is very similar to that described above for Standard Cost. There are a couple of key differences. Where ICJ Rate Category = *Fixed*:

- i. Equipment Rate – Fixed is used instead of Equipment Rate – Standard on EQID
- ii. When performing the lookup to INCR as described in Step 3, then the lookup will match on Rate Category = *Fixed*.

Otherwise, the process for Fixed Cost calculation is the same as described above for Standard Cost

Equipment Surcharge

Equipment Surcharge rates may optionally be established on INCR and EQID. The calculation for Equipment Surcharge = ICJ Regular/Labor Usage Units * Applicable Surcharge Rate from EQID or INCR. Where effective dating on EQID is used, partial Surcharge amounts may need to be calculated for each applicable EQID record, using the same logic described for Standard Cost

- calculation. The difference between Equipment Surcharge and Standard/Fixed amount calculations is in the rate that is used:
- i. Where ICJ Rate Category = *Fixed* and the EQID rates are being used for Fixed Cost calculation, then the Fixed Rate Surcharge is used.
 - ii. Where ICJ Rate Category = *Standard* and the EQID rates are being used for Standard Cost calculation, then the Standard Rate Surcharge is used.
 - iii. When the INCR rate is to be used, it is the Equipment Surcharge Rate that is used instead of Equipment Usage Rate. Note that there are not separate Fixed and Standard Surcharge rates on INCR as each INCR record is specific to the Fixed or Standard Rate Category.
- c. Usage Group = Materials Testing
Only the Standard Cost amount is calculated. Standard Cost = INCR Materials Testing Rate x ICJ Regular Labor/Usage Units.
 - d. Usage Group – Mileage
Only the Standard Cost amount is calculated. Standard Cost = Mileage Rate on System Options table x Regular Labor/Usage Units on ICJ.
5. Complete Processing of ICJ record
 - a. Multiple error situations may arise while processing an ICJ record. Where this occurs, error statistics are accumulated and then written to the Exception report. Costing will not be completed for the record for the following reasons.
 - i. If at any point, the ICJ Usage From/To date range cannot find a matching Employee or Equipment ID record with the right effective dates
 - ii. If at any point, the ICJ Usage From/To date range cannot find a matching EMPBM record with the right effective dates
 - iii. Where INCR record is not found
 - iv. Where ICOM record is not found
 - v. SOPT record not found (for mileage rate)
 - b. Where no errors are encountered and once ICJ record is successfully processed, update the Internal Cost Process Date to the application date to indicate that the Internal Costing Process has been executed and to prevent the record from being selected in future processing runs.
 - c. Accumulate processing statistics to be written to report.
 - d. Proceed to item 6.
 6. The process will read next record in the ICJ table.
 - a. If another record does not exist:
 - i. End the job.
 - ii. Continue to the subsequent job in the chain.
 - b. If another record does exist
 - i. Proceed to #3 and continue processing.
 7. Perform the purge logic for Employee and Equipment ID records. Perform the Purge logic, only if the **Purge Outdated Employee and Equipment ID records?** parameter is “Y” (Yes).

If purging is yet to occur, select and purge records from the Employee ID and Equipment ID tables, if both of the following are true:

- a. Effective End Date is not null on EMPID or EQID table records AND
- b. Effective End Date < (Application Date – ‘Number of Days Old for Purging’ parameter).

As records are purged, the job will monitor how many records are purged and provide statistics in the job log indicating the number of records purged from each table.

The following table shows the various steps that the Internal Cost Update Job goes through and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> • Validating Batch Parameters • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. • Batch Parameter validation completed.
2. Selection of Records	<ul style="list-style-type: none"> • Selecting eligible records from ICJ based on the ICP records. • If the selection returns 0 records, then the following message will be issued: “No eligible record found”. • Selecting records from INCR based on the ICJ record. • If the selection returns 0 records, then the following message will be issued: “Record not found on Internal Rate Table” • Lookup the Mileage Rate on the SOPT table for the ICJ record’s Fiscal Year • If the selection returns 0 records, then the following message will be issued: ” Record not found on Internal Cost System Options Table”
3. Internal cost Calculation	<ul style="list-style-type: none"> • N/A

Restartability Information

If the job fails in any of the above steps, it should be rescheduled. There is no need to back out any updates when this job has failed.

Major Input

- Internal Cost Parameter (R_INT_CST_PARM) table
- Internal Cost Journal (R_INT_CST_JRNL) table
- System Options (R_INT_CST_SOPT) table
- Internal Cost rate (R_INT_CST_RT) table
- Employee Information (R_EMP_INFO)

- Employee Benefits Multiplier (R_EMP_BEN_MULTI)
- Internal Costing Overtime Multipliers (R_INT_CST_OT_MULTI)
- Equipment (R_EQUIP)

Batch Parameters

Parameter	Description	Default Value
(AMSEXPORT)	Required. Export Import Location for Internal Costing Chain Job	\$\$AMSROOT\$\$/ ExportImport
(AMSLOGS)	Required. Logs Location for Internal Costing Chain Job	\$\$AMSROOT\$\$/ Logs
(AMSPARM)	Required. Parameter Location at Internal Costing Chain Job	\$\$AMSROOT\$\$/ Parms
Client Name (CLIENT_NAME)	Optional. Client name for Report	No Default
Commit Block Size (COMMIT_BLK_SIZE)	Required.	10
Maximum Accounting Line (MAX_ACTG_LN)	Required. Please specify an even number. If Labor and Overtime is used along with all Multipliers please specify 24.	50
Number of Days Old for Purging (NM_OF_DYS_OLD_FOR_PURGE)	Optional. Number of Days Old for Purging.	Blank
Parameter File (PARAM_FILE_NM)	Required. Common Parameter File Name.	Parm_File.txt
Parameter ID (PARAM_ID)	Required. The Parameter ID of the ICP record.	No Default
Purge Outdated Employee and Equipment ID records (PURGE_DATA)	Optional. Purge Outdated Employee and Equipment ID records? (Y/N)	N
Select Block Size (SEL_BLK_SIZE)	Required.	1000

Custom Parameters (R_INT_CST_PARM)

Parameter	Description	Default Value
Parameter ID (PARAM_ID)	Required.	No Default

Usage From Date (USG_FRM)	Optional.	No Default
Usage To Date (USG_TO)	Optional.	No Default
Selection Fiscal Year (SEL_FY)	Optional.	No Default
Selection Department (DEPT_CD)	Optional.	No Default
Selection Budget Fiscal Year (SEL_BFY)	Optional.	No Default
Selection Accounting Period (SEL_APD)	Optional.	No Default
Summary Type (SUMM_TYP)	Required.	DETAIL
Transaction Code (DOC_CD)	Required.	CH
Transaction Department (DOC_DEPT_CD)	Required.	No Default
Transaction Unit (DOC_UNIT_CD)	Optional.	No Default
Transaction Prefix (DOC_PFX)	Optional.	No Default
Buyer Event Type (BUYR_EVNT_TYP)	Optional.	No Default
Output Fiscal Year (OTPT_FY)	Optional.	No Default
Output Budget Fiscal Year (OTPT_BFY)	Optional.	No Default
Output Accounting Period (OTPT_APD)	Optional.	No Default

Major Output

- Internal Cost Update Exception Report
- Internal Cost Update Statistics Report
- Internal Cost Journal (R_INT_CST_JRNL) table

Job Return code

The following table shows the potential job return codes for the Internal Cost Update in the Internal Costing Chain.

Return Code	Condition
Successful (1)	Successfully performed the Internal Cost Update for all records.
Warning (4)	No record found in the ICJ table. The Internal Costing job continues processing with rest of the jobs.
Non Fatal Error (8)	No record found in the IEQID, EMPID, EMPBM, ICOM, INCR or SOPT tables.
Failed (12)	<ul style="list-style-type: none"> • Parameters are not valid. • Parameter file name is not entered. • Restart failed because another instance of the Internal Costing chain has already been run successfully. • Runtime exceptions encountered for any unexpected situations. <p>When the job ends with a return code of <i>Failed</i>, subsequent jobs in the chain will be set to <i>Inactive</i>.</p>
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of <i>Terminated</i> , subsequent jobs in the chain will be set to <i>Inactive</i> .
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of <i>System Failure</i> , subsequent jobs in the chain will be set to <i>Inactive</i> .

Sort Sequence

- Transaction ID
- Transaction Code
- Transaction Department Code
- Transaction Accounting Line
- Sequence Number
- Record Date

Selection Criteria

- If the Usage From Date is specified on the selected ICP record, the ICJ records that have a Usage From date greater than or equal to the Usage From on the ICP record will be selected for processing.
- If the Usage To Date is specified on the selected ICP record, the ICJ records that has a Usage To date less than or equal to the Usage To on the ICP record will be selected for processing.

- If the Selection Accounting Period, Selection Fiscal Year, or Selection Budget Fiscal Year is specified on the selected ICP record, the ICJ records with a matching Accounting Period, Fiscal Year, and/or Budget Fiscal Year will be selected for processing.
- If the Selection Department is specified on the selected ICP record, the ICJ record with a matching Department will be selected for processing. Note that matching will occur on the field labeled Department on ICJ and not the Usage Class Department or Equipment/Employee Home Department.
- Internal Cost Process Date on ICJ is null.

Internal Costing Chain: Charge Transaction XML Generation

Job Name	Charge Transaction XML Generation
Recommended Frequency	See chain job
Can be restarted?	No
Reports Generated	Yes

Overview

Charge transactions are created for Internal Costing Journal records that match the selection criteria and for which the Internal Cost Process Date field is populated and the Internal Charge Generation Date is blank.

A single charge transaction can represent multiple employee IDs and multiple COA combinations. The level of detail in accounting lines is controlled by batch parameters. If detail is required, the Employee ID and Usage From/To dates will be entered in the Line Description field, along with a memo reference to the Internal Costing Transaction.

Transactions generated abide by the most restrictive of accounting line limits among a batch input parameter, the Transaction Component Requirements for maximum accounting line limits, and the System Options accounting line limit. The program will start a new charge transaction in the XML file created when the most restrictive of the three limits is reached or the Usage Type changes.

Building the “Buyer” Lines on the Charge Transaction

When building charge accounting lines for the buyer, the chart of accounts fields are populated with the values that are stored in the journal records. Object, Sub-Object, and Department Object may be derived from other sources, such as Internal Costing Overtime Multipliers, Internal Cost Rate, or Special Accounts, if not populated on the source journal record. The event types for the Buyer accounting lines will be parameter driven.

Building the “Seller” Lines on the Charge Transaction

If the Internal Cost Rate includes an Internal Vendor Accounting Data, then a seller accounting line will also be generated in addition to the buyer accounting line. The seller line will contain the Chart of Account (COA) information recorded for the entity that is being reimbursed for the labor, equipment, and materials testing lab usage. Information for the seller accounting line, including the Event Type and COA values will be inferred from Internal Vendor Accounting Data based on the Internal Vendor Code specified on Internal Cost Rate. There is also an option that will allow the seller’s accounting line to be derived from Employee (EMPID) instead. Revenue and Object override options for equipment surcharges are also available on the Internal Cost Rate.

Updates to the ICJ Record from the Charge Transaction XML Generation Process

The process also updates the Charge Generation Dates on journal records that have been selected and successfully processed. If the cost fields are populated with \$0.00 amounts, then no accounting lines will be created for these amounts, but the Charge Generation Date field is still updated. The fact that the Charge Generation Date is populated is no indication that the associated Charge transactions were accepted by the system. If rejected, the Charge transaction would have to be corrected manually. It should not be discarded, as that will not clear the date, so the record can be selected again.

Steps in running this process

1. Select journal records:
 - a. Records have been costed but have not yet produced transactions.
 - The ICJ Internal Charge Generation Date is null.
 - The ICJ Internal Cost Process Date is not null.
 - b. Records will be selected using Internal Cost Parameters selection criteria:
 - If the Usage From is specified on the parameter, the records that have a Usage From date greater than or equal to the Parameter Usage From date are selected for processing.
 - If the Usage To is specified on the parameter, the records that have a Usage To date less than or equal to the parameter Usage To date are selected for processing.
 - If the Selection Accounting Period, Selection Fiscal Year, or Selection Budget Fiscal Year is specified on the parameter, the records with a matching Accounting Period, Fiscal Year, and/or Budget Fiscal Year are selected for processing.
 - If the Selection Department is specified on the selected ICP record, the ICJ records with a matching Department will be selected for processing. Or, if the Department on the ICP record is blank, this field does not impact selection. Note that it is the Department field under Detail Accounting on ICJ that is compared to the ICP Selection Department, and not the Usage Class Department or the Equipment/Employee Home Department located under Usage Information
 - c. If no records are selected, the process will end and a record will be written into the Exception Report to indicate that no records have been selected for processing, and the process continues to the subsequent jobs in the chain. The job step will end with a return code of *Warning*. If records are selected, continue to Step 2.

2. Build detail cost records.

The detail cost records are the input for the charge buyer accounting lines that are generated in this step. A single buyer accounting line is created for each non-zero cost amount for each journal record. For labor records, that means that up to 12 detail cost records may be built for a single journal record (Standard Cost, Overtime Cost, 12 Employee Benefit Costs). For equipment records, up to 2 detail cost records may be created (Standard or Fixed Cost, Equipment Surcharge). Only 1 detail cost record is built for Materials Testing and Mileage records (Standard Cost). The detail cost record consists of the cost amount, the Internal Costing Transaction accounting line reference and the COA elements that used on the buyer accounting line. Regardless of the cost amount, there is some common logic that applies to the building of all detail cost records.

The journal record is initially used as the source for COA elements for all detail cost records – the record having being created using the COA elements from the source transaction accounting line. Once the core structure is in place, specific COA override and inference

options are invoked for individual cost amounts. Note that in all cases where Department Object is being derived from Internal cost Rate, the following logic applies:

- If the Usage Class Department is *ALL*, then no Department Object is populated for the rate and therefore no Department Object is derived.
- If the Usage Class Department is a specific code that does not match the journal department, then the Department Object from the rate is not used.
- If the Usage Class Department is a specific code matching the journal department, Department Object is not populated on the journal record and the Usage Class Department Object is populated on the rate, then the Usage Class Department Object from the rate is used on the buyer accounting line.

a. Standard Salary

If a value is populated on the journal record, that value is always used. Object, Sub-Object, and Department Object may be blank on the journal record. The order of lookup for those values is as follows:

- Where Object, Sub-Object, and Department Object are blank on the source journal record, the rate values are used, if populated.
- If no values exist for Object, Sub-Object, and Department Object on the journal record or rate, the Standard Salary values from Special Accounts are used.

b. Overtime Cost

Object, Sub-Object and Department Object may be blank on the journal record. The order of lookup for those values is then as follows:

- Where Object, Sub-Object, and Department Object are blank on the source journal record, the Internal Costing Overtime Multiplier values are used, if populated.
- Where Object, Sub-Object, and Department Object are blank on the source journal and multiplier record, the Overtime Object, Sub-Object and Department Object from the applicable rate record are used if populated.
- If no values exist for Object, Sub-Object, and Department Object on the journal, multiplier, or rate, the overtime values Special Accounts are used.

c. Employee Benefit Costs

The following logic applies to all 10 Employee Benefit Costs (Standard Employee Benefits 1-5 and Overtime Employee Benefits 1-5). Object, Sub-Object and Department Object may be blank on the journal record. The order of lookup for those values is as follows:

- Where Object, Sub-Object, and Department Object are blank on the journal record, the applicable Benefits Multiplier (1-5) Object, Sub-Object, and Department Object from the rate are used, if populated.
- If the applicable values do not exist for Object, Sub-Object, and Department Object on journal record or rate, the applicable Employee Benefits (1-5) Object and Sub-Object pair from Special Accounts are used.

d. Standard Cost

If a value is populated on the journal record, that value is always used. If blank, the order of lookup for those values is as follows:

- i) The Usage Class Object, Sub-Object and Department Object of the rate are used, if not blank.
- ii) If both journal and rate are blank, Special Accounts is used as follows:

- Where Usage Group = Equipment (ID Optional), the Equipment (ID Optional) Object and Sub-Object are used.
- Where Usage Group = Equipment (ID Required), the Equipment (ID Required) Object and Sub-Object are used.
- Where Usage Group = Mileage, the Mileage Object and Sub-Object are used
- Where Usage Group = Materials Testing, the Materials Testing Rate Object and Sub-Object are used.

e. Fixed Cost

Fixed Cost applies to equipment-related costs only. If a value is populated on the journal record, that value is always used. Object, Sub-Object, and Department Object may be blank on the journal record. The order of lookup for those values is as follows:

- The rate Usage Class Object, Sub-Object, and Department Object are used, if not blank.
- If both the journal and rate are blank, Special accounts is used as follows:
 - Where Usage Group = Equipment (ID Optional), the Equipment (ID Optional) Object and Sub-Object are used.
 - Where Usage Group = Equipment (ID Required), the Equipment (ID Required) Object and Sub-Object are used.

f. Equipment Surcharge

Equipment Surcharge applies to equipment-related costs only. Equipment Surcharge is the only cost record where values from the source journal record may be replaced. The order of precedence for COA lookup is as follows:

- Where an Equipment Surcharge Buyer IVAD record is specified on the applicable rate record, the COA values from that IVAD record are initially used.
- When not specified, the COA values from the journal record are initially used.
- If after setting COA values using either location above the Object, Sub-Object and Department Object values are still blank, then the Equipment Surcharge Buyer Object, Sub-Object, and Department Object from the applicable rate are used.

3. Build Detail Seller Accounting Lines

If IVAD Vendor Code is populated on the journal record, then a matching seller line is created for each buyer accounting line that is created. For example, if both the Standard Cost and Equipment Surcharge Cost are non-zero amounts, then two buyer accounting lines and two seller accounting lines will be created if the applicable IVAD Vendor Code is not blank. Note that if the IVAD Vendor Code is null, then no seller lines should be generated. Creation of a seller accounting line should occur immediately after the “buyer” line that it corresponds.

Build the Seller’s COA: Look to IVAD first. The referenced IVAD record’s COA elements are always used for the seller accounting line except in the following cases:

- Where Usage Group = Labor and if the Get COA from Employee ID indicator is true on the IVAD record, then the COA elements from Employee Information are used instead of those on IVAD. This applies for all labor-related seller accounting lines (Standard Salary & Employee Benefits).
- Two important points to note when the indicator is Yes:
 - If there are multiple applicable employee records based on multiple Usage Dates, the COA values to be used on the seller line are those from the earliest applicable Employee record

- If there are no COA fields populated on the employee record(s), then the IVAD record will be used. If at least one COA field on an employee record is populated:
 - When the Use Employee Only batch parameter is *True*, then the COA elements will be updated from employee only.
 - Where *False*, then the COA elements will be updated from the employee and IVAD. Employee is the priority so IVAD will only contribute COA where the IVAD COA field is blank.
 - Where Usage Group = Equipment (ID Optional) or Equipment (ID Required) and where the cost record relates to Equipment Surcharge; the Revenue Source, Sub Revenue Source, and Department Revenue values on the applicable rate record will override the corresponding field values on the IVAD record. If there are no values specified on rate, then the IVAD values are retained, if available.
 - Where Usage Group = Equipment (ID Optional) or Equipment (ID Required) and where the cost record relates to Equipment Surcharge; the Object, Sub Object, and Department Object values on the applicable rate record override the corresponding field values on the IVAD record. If there are no values specified on the rate, then the IVAD values are retained, if available.

a. Assign Event Type.

Two scenarios exist for the Seller Event Type:

- If the IVAD value for Single Event Type is populated, it is used for the seller accounting line.
- If the IVAD value for Inter-COA Seller Event Type is populated, then the following steps are used to determine the event type:
 - Inter-COA Definition = Fund Only, then compare only fund between buyer and seller.
 - Inter-COA Definition = Fund and Department, then compare those two between buyer and seller.
 - Inter-COA Definition= Fund, Department, and Appropriation, then compare those three between buyer and seller.
 - Inter-COA Definition = Appropriation Classification, then compare the value of the Appropriation Classification (APPR_CLSN) (derived from Appropriation (R_APPR) with the buyer line Appropriation Unit) with the Appropriation Classification value (see logic for buyer line lookup) of the seller line.
 - Inter-COA Definition = Fund and Department and Appropriation Classification, then compare the Fund, Department and Appropriation Classification values (see logic for derivation in above) of the buyer line with those of the seller line.
- Set the Event Type and blank out either the Object or Revenue Source fields:
 - If the Inter-COA comparison found that the COA fields match, then use the Intra-COA Event Type and move null values to the Revenue Source, Sub-Revenue Source, and Department Revenue Source of the generated seller line. Those values would have been assigned, when building the seller line accounting distribution in the previous step.
 - If the Inter-COA comparison found that the COA fields are different, then use the Inter-COA Event Type and move null values to the Object, Sub-Object, and Department Object of the generated seller

line. (Note: the buyer line will still use the Object fields). Those values would have been assigned, when building the seller line accounting distribution in the previous step.

- b. Perform common logic for all Detail Cost records:
 - c. If the ICP Output Accounting Period, Output Fiscal Year, and/or Output Budget Fiscal Year are populated use those values for the cost record. The Fiscal Year, Accounting Period, Budget Fiscal Year, and Description for the seller line should be populated with the same value as the buyer line. For any of these 3 fields that is blank, use the value from the ICJ record.
 - d. If the ICP summarization type is “Detailed”, set the Reference fields to memo-reference the ICT transaction and line that recorded the costs.
4. Create CH transaction Accounting Lines

At this stage, the logic for creation of the transactions will vary based on ICP **Summary Type**. The logic for both run modes is described below:

a. Detail Mode

- If the ICP Summary Type is Detail, set the Reference fields to memo-reference the ICT transaction and line that recorded the costs.
- Build Accounting Line Description:
 - If the Usage Group is not Materials Testing, the Buyer Accounting Line Description will contain the value of ‘ EMP/EQUIP ID: ‘+ <Employee/Equipment ID> + ‘Usage: ‘<Usage From> + ‘-‘+ <Usage To>+”ICJ Amt: “ + <three-letter ICJ Amount code (list below)> + “ ICJ Seq: “ + <SEQ_NO from ICJ>.. :
 - If the Usage Group is Materials Testing, the Accounting Line Description field will contain the following information: ‘Materials Testing Class: ‘+ <Materials Testing Class> + ‘Usage: ‘<Usage From> + ‘-‘+ <Usage To> +”ICJ Seq:”+<Seq from ICJ>

b. Summary Mode

- Buyer and Seller Accounting Lines are summarized as follows:
 - Output Transaction Department
 - Usage Type
 - Chart of Account elements
 - Accounting Period
 - Fiscal Year
 - Budget Fiscal Year
- If the **ICP Summary** Type is *Summary*, the Reference fields will not be populated.
- Build Accounting Line Description:
 - The Line Description for summarized records should be set to ‘Summarized Charge Transaction – Usage Type:’ + (Usage Type).

5. Build Charge Transaction Headers

- a. Build Charge Transactions by combining lines until a break is necessary. Charge transaction breaks should occur for any of the following reasons:
 - If the Output Transaction Department changes.
 - The Usage Type changes.

- If Usage Group = Equipment (ID Required) or Equipment (ID Optional), then if the Equipment ID changes. Note that summarized output “summarizes away” Equipment IDs, so they’re all null. Summary transactions will essentially ignore Equipment ID for breaking logic.
- If Usage Group = Labor, then if the Employee ID changes. This is only for Usage Group Labor, as Employee ID can be populated on other records but it should not be part of the breaking logic for any other Usage Group. Note that summarized output “summarizes away” Employee IDs, so they’re all null. Summary transactions will essentially ignore Employee ID for breaking logic.
- The number of accounting lines exceeds the maximum accounting line limit that is the lowest value of the batch parameter, the transaction component’s record on DCREQ (in Admin application), or SOPT.
- If the record that exceeds the limit is a seller line, then the buyer line that generated the seller line (the line that directly preceded the seller line) should also be moved to the new Charge transaction. The general rule is that related buyer & seller lines should appear on the same transaction.

b. Output Transaction Department

- The ICP Transaction Department value if the ICP Transaction Department value <> “ALL”.
- The ICJ Department if the ICP Transaction Department is = “ALL”.

c. Transaction ID

The Transaction ID on the Charge Transaction will be assigned via auto numbering.

- If the user specified the Transaction Prefix on ICP, then follow this process.
 - Read ADNT with the ICP prefix, the department code of the transaction, the fiscal year of the first record on the Charge, and the CH transaction code.
 - If a record cannot be found on ADNT using the actual department code, then re-attempt the ADNT lookup using the Department value ‘*****’.
- If Transaction Prefix parameter was left blank on ICP, then follow this process:
 - Read ADNT with prefix ‘*****’, the department code of the transaction, the fiscal year of the first record on the Charge, and the CH transaction code.

d. Transaction Description

- If the **Summary Type** is *Detail*, then populate the Transaction Description as follows.
 - If Usage Group = Labor, set the Transaction Description to “Usage Group Labor - Employee ID: <Employee ID>”.
 - If Usage Group = Equipment (ID Required) or Equipment (ID Optional) and Equipment ID is not null, set the Transaction Description to “Usage Group Equipment – Equipment ID: <Equipment ID>”.
 - If Usage Group = Equipment (ID Optional) and Equipment ID is null, set the Transaction Description to “Usage Group Equipment – Equipment ID not specified”.

- If Usage Group = Mileage, set the Transaction Description to “Mileage”
 - If Usage Group = Materials Testing, set the Transaction Description to “Materials Testing”
 - If the Summary Type is Summary, then populate the Transaction Description as follows: “Summary Output for <Usage Group> + “<Usage Group Description>”
6. Update the Internal Charge Generation Date to the application date for the matching ICJ records to indicate that the Generate Charge Transaction Process has been executed and to prevent the records from being selected in future processing runs.
 7. Continue to the subsequent job in the Job Chain.

Major Input

- Internal Cost Parameter (R_INT_CST_PARM) table
- Internal Cost Journal (R_INT_CST_JRNL) table
- Special Accounts (R_INT_CST_SPEC) table
- Internal Cost rate (R_INT_CST_RT) table
- Internal Vendor Accounting (R_INT_VEND_ACTG) table
- Employee Information (R_EMP_INFO)
- Internal Costing Overtime Multipliers (R_INT_CST_OT_MULTI)

Batch Parameters

Parameter	Description	Default Value
CH XML File Name (CH_XML_FILE)	The required name of the XML file created.	\$\$AMSEXPORT\$\$/CH_DOC.xml
Commit Block Size (COMMIT_BLOCK_SIZE)	A required performance parameter that controls saving information into the XML file.	10
Exception Report File Name (EXCEP_REP_FILE_NAME)	The required file name for data that is included in the exception report.	\$\$AMSEXPORT\$\$/EXCEP_CH.txt
Exception Report Indicator (EXCEP_REP_IND)	The required level of information in the exception report: (1- Detailed ; 2 - Failed Transactions; 3-Processed Transactions;4 - Failed Transaction Line)	4
Upload Parameter File Name (LOAD_PARM_FILE)	The required file name passed onto the later load step with instructions.	\$\$AMSPARM\$\$/CH Upload.txt
Common Parameter File Name (PARM_FILE)	The required file name of the parameter file created in the first job step with instructions for later job steps.	\$\$AMSPARM\$\$/PARM_FILE.txt

Parameter	Description	Default Value
Submit Parameter File Name (SUBMIT_PARM_FILE)	The required file name passed onto the later submit step with instructions.	\$\$AMSPARM\$\$/CH Submit.txt
Temporary Folder (TEMPFOLDER)	The required location that temporary files created and used by this job step are stored.	\$\$AMSTEMP\$\$
Use Employee Only (USE_EMP_ONLY)	A required parameter that is used when the Usage Type is Labor and the Get COA from Employee ID indication on Internal Vendor Accounting (IVAD) is <i>true</i> . When the parameter is <i>True</i> , only employee COA will be used. When <i>False</i> , the COA specified on IVAD and not EMPLID will supplement what is on EMPLID. Please see the description for " Build the Seller's COA " for more details on this parameter and related system setup.	True

Major Output

- CHDocXMLGenerator Report
- Exception Report
- CH_DOC.xml
- Internal Cost Journal (R_INT_CST_JRNL) table

Batch Return Codes

The following table shows the potential job return codes for the Charge Transaction Generation in the Internal Costing Chain.

Return Code	Condition
Successful (1)	Successfully performed the Charge Transaction XML generation for the selected records.
Warning (4)	No eligible record found in the ICJ table. The process terminates, the subsequent jobs in the chain will be set to inactive and the Internal Costing Job does not continue with the rest of the processes.
Non Fatal Error (8)	No record found in the INCR, SPEC, EMPID, ICOM or IVAD tables.
Failed (12)	<ul style="list-style-type: none"> • Parameters are not valid. • Parameter file name is not entered. • Exception file name is not entered.

Return Code	Condition
	<ul style="list-style-type: none"> • Submit file name is not entered. • Upload file name is not entered. • Restart failed because another instance of the Internal Costing chain has already been run successfully. • Runtime exceptions encountered for any unexpected situations. <p>When the job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.</p>
Terminated (16)	<p>This return code will be issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain will be set to <i>Inactive</i>.</p>
System Failure (20)	<p>This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to <i>Inactive</i>.</p>

Sort Sequence

The Charge Transaction XML Generation process will sort the detail cost records by the following fields:

- Cost Record Output Transaction Department
- Usage Type
- Usage Class
- Chart of Account elements
- Cost Record Output Accounting Period
- Cost Record Output Fiscal Year
- Cost Record Output Budget Fiscal Year
- Seller Accounting Line IVAD Vendor

Selection Criteria

The Charge Transaction XML Generation process will select records on the Internal Costing Journal (ICJ) table that meet all of the following criteria:

1. If the Usage From is specified on the selected ICP record, the ICJ records that have a Usage From date greater than or equal to the ICP Usage From date will be selected for processing. Or, if the Usage From on the ICP record is blank, this field does not impact selection.
2. If the Usage To is specified on the selected ICP record, the ICJ records that has a Usage To date less than or equal to the ICP Usage To date will be selected for processing. Or, if the Usage To on the ICP record is blank, this field does not impact selection.
3. If the Selection Accounting Period, Selection Fiscal Year, or Selection Budget Fiscal Year is specified on the selected ICP record, the ICJ records with a matching Accounting Period, Fiscal Year, and/or Budget Fiscal Year will be selected for processing. Or, if these fields are blank on the ICP record, this field does not impact selection.

4. If the Selection Department is specified on the selected ICP record, the ICJ records with a matching Department will be selected for processing. Or, if the Department on the ICP record is blank, this field does not impact selection. Note that it is the Department field under Detail Accounting on ICJ that is compared to the ICP Selection Department, and not the Usage Class Department or the Equipment/Employee Home Department located under Usage Information.
5. The system shall select the records for which the Internal Cost Process Date field on the ICJ table is not null, but the Internal Charge Generation Date is null.

Summarization Criteria

If the **Summary Type** of the ICP parameter is set to *Summary*, the process will summarize the detail cost records by the following fields:

- Cost Record Output Transaction Department
- Usage Type
- Chart of Account elements
- Accounting Period
- Fiscal Year
- Budget Fiscal Year

Internal Costing Chain: CH Upload Job

Job Name	Upload CH transactions
Recommended Frequency	This job can be run daily as part of the nightly cycle or on demand.
Parallel processing enabled	No. Parallel processing is not supported for this job.
Can the job be restarted?	Optionally, based on the Save Restart Information parameter.
Exception report produced	No. All exceptions are written to the log.

Overview

The CH Upload job loads the records from the XML File, generated by the Charge XML generation job, into the Transaction Catalog. This job uses the common utility to load the records into the Transaction Catalog. This job first validates the batch parameters. If the parameters are valid, then it loads the records into the Transaction Catalog. If the parameters are not valid, the job issues appropriate messages and ends with a status of Failed. Once the records are loaded into the Transaction Catalog, the summary information is written into the log as how many records were in the input file and how many records loaded successfully.

The job can be restarted if it fails, provided the Save Restart Information parameter is selected. If the failure occurred after the parameter validation, then the job should be restarted after resolving the errors. If the Save Restart Information parameter is not selected or if the restart is not the immediate option, then the new job can be rescheduled but before rescheduling the job, the transactions loaded by the failed job should either be processed or discarded so that they do not remain in the catalog.

Major Input

- CH Transaction XML file.

Batch Parameters

Parameter	Description	Default Value
Parameter file (PARAM_FILE)	Parameter file to Load Transactions	\$\$AMSPARM\$\$/CHUpload.txt
Transaction Status Code (DOC_STA_CD)	Transaction status code	2

Note: This PARAM_FILE only contains the following subset of SMU parameters.

Parameter	Default Value
Action Code (ACTN_CD)	162
Commit Block Size (COMMIT_BLOCK)	10
File Name To Be Imported (FILE_NM)	\$\$AMSEXPORTE\$\$/CH_DOC.xml
By Pass Auto Transaction Number (BYPAS_ADNT_FL)	True
Restart Flag (RESTART_FL)	True
Statistics (STATS)	True

Please refer to the “SMU Transaction Upload Job” run sheet in the *CGI Advantage Financial – Utilities Run Sheets* guide for the full list of SMU Transaction Upload batch parameters.

Major Output

- CH Transactions in draft version

Batch Return Codes

The following table shows the potential job return codes for the CH Upload job.

Return Code	Condition
Successful (1)	All of the records are loaded into the Transaction Catalog successfully or the input file is empty.
Warning (4)	This return code will be issued when some of the records failed to load where as all other records were loaded successfully.
Non Fatal Error (8)	None of the records get loaded into the Transaction Catalog.
Failed (12)	<ul style="list-style-type: none"> • Parameters are invalid • When the input file is not found in the specified directory • Restart failed because another instance of the Internal

Return Code	Condition
	<p>Costing chain has already been run successfully</p> <ul style="list-style-type: none"> Runtime exceptions encountered for any unexpected situations <p>When the job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.</p>
Terminated (16)	<p>This return code will be issued when the job is terminated by the user. When the job ends with a return code of <i>Terminated</i>, subsequent jobs in the chain will be set to <i>Inactive</i>.</p>
System Failure (20)	<p>This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of <i>System Failure</i>, subsequent jobs in the chain will be set to <i>Inactive</i>.</p>

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

If the job ends with a return code of Failed and above, the job can be restarted.

Internal Costing Chain: CH Submit Job

Job Name	CH Submitter
Recommended Frequency	This job can be run daily as part of the nightly cycle or on demand.
Single Instance Required	No
Can be restarted?	Yes
Report Generated	No. All of the exceptions are written to the error file.

Overview

This job submits the transactions listed in the input parameter file that was generated by the CH Transaction Upload Job. In order to run the CH Submit step in parallel, the Job Interaction Client (JIC) must be used (refer the *CGI Advantage System Administration Guide* for more information on JIC). Also, when running in parallel, the 'Process Assigned Jobs Only' flag must be set to FALSE in the Job Server Control for all job managers.

Major Input

- SMU job parameter file
- Draft CH Transactions in the catalog

Batch Parameters

Parameter	Description	Default Value
Submit Parameter File Name (PARM_FILE)	Parameter file to read	\$\$AMSPARM\$\$/CHSubmit.txt

Note: This PARM_FILE only contains the following subset of SMU parameters.

Parameter	Default Value
Action Code (ACTN_CD)	171
Exception Report File Name (EXCEP_REP_FILE_NM)	\$\$AMSEXPORTEXCEPT.txt
Exception Report Indicator (EXCEP_REP_IND)	4
Transaction Code (DOC_CD)	CH (Transaction Code entered in custom parameter)
Transaction Phase Code (DOC_PHASE_CD)	1
Transaction Status Code (DOC_STA_CD)	1
Transaction Department Code (DOC_DEPT_CD)	Transaction Department Code entered in custom parameter
Transaction Identifier (DOC_ID)	Transaction ID generated during execution of job
Commit Block Size (COMMIT_BLOCK)	10
Restart Flag (RESTART_FL)	True
Statistics (STATS)	True

Note: This job uses only a subset of the SMU submit job parameters. For a full list of available parameters for the SMU submit job, refer to the “SMU Transaction Submit Job” run sheet in the *CGI Advantage Financial – Utilities Run Sheets* guide.

Major Output

The Transactions created in the previous job will be processed to final or rejected.

Batch Return Codes

The following table shows the potential job return codes for the individual Submit CH job in the submitter.

Return Code	Condition
Successful (1)	All of the transactions generated in that run submitted successfully.
Warning (4)	Not Applicable for this job.
Non Fatal Error (8)	Not Applicable for this job.
Failed (12)	<ul style="list-style-type: none"> • Input parameter file is not found. • Restart failed because another instance of the Internal Costing Chain has already been run successfully. • Runtime exceptions encountered for any unexpected situations. <p>When the job ends with a return code of failed, subsequent jobs in the chain will be set to inactive.</p>
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of <i>Terminated</i> , subsequent jobs in the chain will be set to <i>Inactive</i> .
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of <i>System Failure</i> , subsequent jobs in the chain will be set to <i>Inactive</i> .

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

If the job ends with a return code of Failed and above, the job can be restarted.

Internal Costing Chain: CH Exception Report

Job Name	CH Exception Report
Recommended Frequency	Daily as part of the nightly cycle.
Single Instance Required	Yes
Can be restarted?	No
Report Generated	Yes. The job generates the CH Transaction Exception Report in PDF and html formats.

Overview

This job in the Internal Costing chain generates an exception report that lists all of the errors encountered when the CH transaction was submitted in the earlier step. This report will list those CH transactions created that did not reach the *Final* transaction phase. The report provides the details of Transaction Code, Transaction Department, and Transaction ID for each transaction that is not final.

Major Input

- Draft CH Transactions in the catalog with a Rejected status

Batch Parameters

Parameter	Description	Default Value
EXCEP_REP_FILE_NM	Exception Report File Name	\$\$AMSEXP ORT\$\$/EXC EP_CH.txt
PARAM_FILE_NM	Common Parameter File Name	\$\$AMSPAR M\$\$/Parm_F ile.txt

Major Output

- CH Exception report with the CH transaction Id along with the error messages.

Sort Sequence

N/A

Selection Criteria

N/A

2.1.13 Internal Overhead Process

Chain or Job Name	Internal Overhead Process
Recommended Frequency	This process can be run on demand, daily, weekly or monthly basis
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	No

Overview

Internal Overhead Process calculates the various overhead costs for those Internal Costing Journal (ICJ) records with an Overhead Fiscal year, using information setup on the Overhead Rate (OHRT) and Overhead Variation Code (OVARC) pages. This process must be run before the Internal Costing Chain is run in order to get that chain to create the accounting transactions for overhead.

Please note that if the process finds a record on the ICJ with an Overhead Fiscal Year where it cannot determine how to process the record because of missing data, it will skip that record, log what record was skipped and why in the job log, and finish processing with a Return Code of *Successful*. For this reason, always check the job log.

There are two processing steps:

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> • Run Started • Validating Batch Parameters <ul style="list-style-type: none"> • Batch parameters are listed with values supplied • Any parameter errors are issued. • Parameters are validated successfully <ul style="list-style-type: none"> • Parameter validation failed
2. Selection and Update of Records	<ul style="list-style-type: none"> • Total ## records are processed • Run Ended

Restartability

After parameter validation, if the job fails it can be restarted.

Major Input

- Internal Costing Journal (ICJ / R_INT_CST_JRNL)
- Overhead Rate (OHRT / R_OHRT)
- Overhead Variation Code (OVARC / R_VARC)
- Internal Overhead Process Parameters (IOPP / R_INT_OVD_PARM)
- Department (DEPT / R_DEPT)

- Task Order (TASKORD / R_TASK_ORD)
- Task (TASK / R_TASK)

Note: The default values listed are those delivered with the software. Actual values may vary based on your site's setup.

Parameter	Description	Default Value
Commit Block (COMMIT_BLK_SIZE)	A required performance parameter that defines the number of records committed in an instance.	1000
Parameter ID (PARAM_ID)	A required ID from the Internal Overhead Process Parameters page to control selection and processing.	(no default)
Progression Counter (PROG_CTR_SIZE)	A required parameter used to push updates to the job log as progress is made through journal records.	1000
Select Block (SEL_BLK_SIZE)	A required performance parameter that defines the number of journal records selected in one group for processing.	1000

Major Output

- Internal Costing Journal (ICJ / R_INT_CST_JRNL)

Return Codes

The following return codes are issued for the process.

Return Code	Condition
Successful (1)	When selected ICJ records were processed successfully.
Warning (4)	No ICJ records were found.
Non-Fatal Error (8)	This job does not use this return code.
Failed (12)	Failed parameter validations.
Terminated (16)	The job was terminated by the user.
System Failure (20)	The job was terminated because of database server or network issues.

Sort Criteria

Record Date (REC_DT), Transaction Code (DOC_CD), Transaction Department (DOC_DEPT_CD), Transaction ID (DOC_ID), Transaction Accounting Line (DOC_ACTG_LN_NO), and Sequence No (SEQ_NO)

Selection Criteria

- Department Code of IOPP ID

- Overhead Fiscal Year of IOPP ID
- Period of IOPP when specified
- Variation Code Type is a value besides *Other*
- Flag 1 is *No*
- Usage Type is *Labor, Interfaced Labor, Adjustment, or Retroactive*
- Overhead Process Date is blank when Recalculate Overhead is *False*; otherwise, when *True* the Overhead Process Date is not blank

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully and records selected and processed successfully.	N/A	N/A
Warning (4)	Job ended with a Warning <ul style="list-style-type: none"> • No ICJ records met the selection criteria as specified on the IOPP with Parameter ID:30 	Review IOPP setup and ICJ data to determine if a correction and subsequent run is necessary or not.	N/A
Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	Job failed due to parameter validation failure or any Runtime Exception.	In this step, the job can fail under the following two conditions. <ul style="list-style-type: none"> • Encounters any runtime exceptions • Parameter Validation Verify the parameters and restart the job. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	
	Failed while restarting the job since another	Recommendation: Schedule a new job.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	<p>instance of the job has already been run successfully.</p> <p>Sample Message: Cannot restart the job since another instance of this job has already been run successfully.</p>		
s	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	Job should be rescheduled after correcting the problem.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	Job should be rescheduled after correcting the problem.

2.1.14 Labor Additive

Chain or Job Name	Labor Additive
Recommended Frequency	On demand as needed
Single Instance Required	Yes
Can be restarted?	Refer to the Restart Information section
Reports generated	No

Overview

Departments arrive at a methodology to charge for fringe benefits to projects and grants based on interactions with the funding entities of each Project or Grant. One methodology commonly adopted is to charge an additive for fringes based on a negotiated percentage of eligible direct salary expenditures. In this model, employees working on Projects or Grants charge their time to the Project or Grant, while their fringes go to a separate accounting distribution, generally without the corresponding Project or Grant. Based on the percentage of additives agreed with funding entities, the Project or Grant then needs to be charged (agreed percentage of direct charge) an additive towards the fringes with the corresponding credits posted against the accounting distribution where fringes were originally charged.

The Cost Allocation process calculates and records these additives with setup added with the Labor Additive chain. When run in *Charge Back* mode, Cost Allocation creates transactions to record Labor Additive charges as cash expenditures to eligible projects and grants based on a defined percentage of direct salaries, with offsetting expenditure credits to employee standard fringe benefit account distributions in the same proportion as originally recorded.

The Labor Additive chain selects payroll expense transactions from accounting records and performs a look-up into the Payroll Expense Journal to identify the GTN Run Numbers and Employee IDs tied to the accounting records. With those two added details, the process reads the Fringe Journal to determine the unique accounting distributions where fringes posted and the proportion of each to the total. The process then updates Cost Allocation setup, loading these accounting distributions as bases with the proportion as allocation percentage

In summary, the process:

- Selects the eligible subset of accounting records from one or more pay periods
- Connects those to the source Payroll Expense Journal and Payroll Fringe Journal records
- Calculates the sum for each fringe chart of account (COA) string
- Defines the percentage of each string of the total
- Creates and loads cost allocation setup data to perform the charge back.

As the result of the Labor Additive process is to update an existing cost allocation structure setup for a *Charge Back*, that setup must be in place first. A pool record must exist in Pool Base Setup (PLBS) and Pool Base Distribution (PBDIST) pages for the specified Cost Allocation ID, Series ID, and Step ID.

The Labor Additive chain consists of the following job steps:

Labor Additive Financial Selection

This job selects payroll related cash expenditure transactions from the source journal for a specified time period, (typically a pay period but could be multiple periods). The process selects

using the Posted Date (PSTD_DT) journal field and not the Record Date. The journal must be used instead of a ledger not because of the date range, but because the transaction identification fields are required to match to payroll data in a later job step.

Each of the selection criteria are critical to the process loading cost allocation setup correctly, according to a labor additive agreement or a group of similar agreements:

- Posting Code – External cash expenditures are the only type of accounting activity selected. Balance sheet activity is not applicable. Accrued expenditures would only end with extra processing time. Internally funded items for revenues and expenditure credits are not applicable.
- Objects – As not all payroll costs are eligible in reimbursement agreements, selection here is critical to comply. Individual object codes can be entered, or one or more rollup codes when all objects within the rollups are eligible.
- Programs – The Labor Additive process only works with reimbursement billing that requires program or its rollup. The process does not support the use of other chart of accounts to represent projects and grants. This selection has to match the cost allocation setup, which also matches the additive agreement with the funding agency or entity.
- Department – Three of the five levels of program selection require department.

The job validates whether the Cost Allocation setup information provided, such as, Cost Allocation ID, Series ID, and Step ID, exists or not.

The process selects records meeting the selection criteria from the specified Journal and loads them to a temporary database table (LBR_ADTV_TMP1).

Job Steps	Messages
Temporary Table Truncation	<ul style="list-style-type: none"> • "Rows Count" deleted from LBR_ADTV_TMP1 • "Rows Count" deleted from LBR_ADTV_TMP2 • "Rows Count" deleted from LBR_ADTV_TMP3
Parameter Validation	<ul style="list-style-type: none"> • Parameters successfully validated.
Load Temporary Table 1	<ul style="list-style-type: none"> • LBR_ADTV_TMP1 table successfully populated with "Records Count " records.

Labor Additive HRM Selection

The job takes records processed from the first job from LBR_ADTV_TMP1 and matches them to the Payroll Expense Journal using Transaction ID, Accounting Line, and Transaction Code (PRXP -Expense and PREXC - Expense Correction). Matching records are then loaded to another temporary database table (LBR_ADTV_TMP2). The job performs a look-up into Payroll Fringe Journal by using the GTN Run Number, Employee ID from initial processing and updates previously loaded records with Fringe COA elements from the Payroll Fringe Journal. The job only selects records where a Journal Status Indicator (JRNL_STATUS_IND) is *Reconciled* or *Unreconciled*.

Job Steps	Messages
Parameter Validation	<ul style="list-style-type: none"> • Batch Parameter validation completed.
Load Temporary Table 2	<ul style="list-style-type: none"> • "Progression counter value" Payroll Expense Journal History Net Pay records loaded to LBR_ADTV_TMP2 table. • Payroll Expense Journal History Net Pay records

Job Steps	Messages
	<p>loaded to LBR_ADTV_TMP2 table. Total Records Added: "Total Records count"</p> <ul style="list-style-type: none"> • "Progression counter value" Payroll Expense Journal History Correction records loaded to LBR_ADTV_TMP2 table. • Payroll Expense Journal History Correction records loaded to LBR_ADTV_TMP2 table. Total Records Added: "Total Records count"
Update Temporary Table 2	<ul style="list-style-type: none"> • "Progression counter value" Payroll Fringe Journal History records updated to LBR_ADTV_TMP2 table • Payroll Fringe Journal History records updated to LBR_ADTV_TMP2 table. Total records updated: "Total Records count"

Labor Additive Base Calculation

The job takes records processed from the previous job from LBR_ADTV_TMP2 and loads them into another temporary database table (LBR_ADTV_TMP3). The job summarizes those records in the third table by unique chart of account string, summarizing the Contract Pay Amount (CNTRCT_PAY_AM). The process then calculates Allocation Percentage for each record by taking the proportion of Contract Pay Amount (CNTRCT_PAY_AM) of each unique accounting string over sum of Contract Pay Amount of all records.

The process then goes on to delete the Base records (PLBS_IND=2) from Pool Base Setup (PLBS / R_PLBS_SETP) and Pool Base Distribution (PBDIST / R_PLBS_DIST) tables for the Cost Allocation ID, Series ID, and Step ID specified in the job parameter.

Job Steps	Messages
Parameter Validation	<ul style="list-style-type: none"> • Parameters Successfully Validated.
Load Temporary Table 3	<ul style="list-style-type: none"> • "Progression counter value" Record(s) inserted to LBR_ADTV_TMP3. • Successfully loaded LBR_ADTV_TMP3 table with LBR_ADTV_TMP2 data for each unique grouping of fringe (FRNG) chart of accounts fields.
Calculate Allocation Percentages	<ul style="list-style-type: none"> • Base Percentage has been calculated and updated on LBR_ADTV_TMP3 table "Records Count" records were updated
Delete PLBS and PBDIST Records	<ul style="list-style-type: none"> • Deleting Cost Allocation Base records • Successfully deleted records from Financial Pool/Base Distribution table for Cost allocation ID " ID here", Series ID " ID here ", Step No " Number here " combination. • Successfully deleted records from Financial Pool/Base Setup table for Cost allocation ID "ID Mentioned ", Series ID " ID Mentioned ", Step No " Number here" • All processing completed

Labor Additive PLBS Prep

The job creates an XML file with a record for each row on LBR_ADTV_TMP3 for upload to Pool Base Setup. These rows are written as Bases (PLBS_IND = 2) with a Base Type of Fixed Percentage (BASE_TYPE = 1) on the XML File. The job sets the Allocation Percentage calculated from the previous step.

Prior to creation, if a file already exists, the job deletes the existing file and creates a new file. If there is a failure to load the file, ensure another chain is not scheduled until that problem is resolved loading the previous file. Run the chain again with the same settings.

Job Steps	Messages
Parameter Validation	<ul style="list-style-type: none"> Parameters Successfully Validated.
PLBS XML File Creation	<ul style="list-style-type: none"> Completed processing "Progression count" records. File "File name" Created.

Labor Additive PLBS Load

The job uses System Maintenance Utility (SMU) to load the Pool Base Setup (PLBS) records.

Labor Additive PBDIST Prep

The job creates an XML file with a record for each row on LBR_ADTV_TMP3 for upload to Pool Base Distribution. These rows are written as Bases (PLBS_IND = 2) with a Distribution Type of Accounting (DIST_TYP = 1) on the XML File. Further, for individual COAs where a value is not present, the system writes a wildcard value of '!'. Capturing this wildcard indicates that on the cost allocation transaction generated, the base accounting lines are blank for that COA. The job overrides the OBJ_CD and sets the Object code on Base lines from the Cost Allocation Credit Object Code batch parameter.

Prior to creation, if the file already exists, the job deletes the existing file and creates a new file. If there is a failure to load the file, ensure another chain is not scheduled until that problem is resolved loading the previous file. Run the chain again with the same settings.

Job Steps	Messages
Parameter Validation	<ul style="list-style-type: none"> Parameters Successfully Validated.
PBDIST XML File Creation	<ul style="list-style-type: none"> File " File name " Created. All work completed.

Labor Additive PBDIST Load

The job uses System Maintenance Utility (SMU) to load the Pool Base Distribution (PBDIST) records.

Labor Additive History Table

The job copies the records from the temporary database table (LBR_ADTV_TMP2) and stores the records in the permanent Labor Additive History (LBR_ADTV_HIST) table. The job additionally adds the Batch Date and Time (BATCH_DT_TM) value, inferring from the System Date.

Job Steps	Messages
Parameter Validation	<ul style="list-style-type: none"> No Parameters present
Load Labor Additive History	<ul style="list-style-type: none"> Calling to load the Labor Additive History table Added " Records Count " records into the

Job Steps	Messages
	LBR_ADTV_HIST • Successfully committed LBR_ADTV_TMP2 records to LBR_ADTV_HIST table. • Completed load of the Labor Additive History table

Pre-conditions/Considerations:

The Labor Additive model works with the charge back method of cost allocation and updates parts of Cost Allocation setup. Given this, the cost allocation setup should be done prior to running the Labor Additive chain. While setting up Cost Allocation care should be taken for the following:

- Charge Back should be selected while configuring the Cost Allocation Control (ALOC) record. There is no validation in the process to verify whether Cost Allocation is configured with Charge Back method.
- A pool record should exist in Pool Base Setup (PLBS) and Pool Base Distribution (PBDIST) pages for the specified Cost Allocation ID, Series ID, and Step ID. If there is no pool record, the Labor Additive PLBS and PBDIST Load jobs fails.
- When fringes are originally posted against an Object/Sub Object combination, the job only overrides the Object code, so this may result in a situation where Sub Object may not exist for the Object. In such cases, it is recommended to update the Pool Base Distribution Base records manually, prior to running the Cost Allocation process.
- On the Cost Allocation Step Control (STEP) page the Charge Back Object indicates which object code should be used for an additional charge. When left blank, the system uses the Object code from the Pool line that is used for calculating the additive that needs to be charged to the Project/Grant. It is recommended to use a different Object code for an additional charge, depending on the model chosen to charge it to the pool object code.

Restart Information

- If the job failed due to any reason, schedule a new job after correcting the errors that caused the job to fail.

Major Input

Tables

The following list of tables are primary input for the Labor Additive process. Not listed are those tables just used for parameter validation.

- Cost Accounting Journal (JCA / JRNL_CA) or Accounting Journal (JACTG / JRNL_ACTG)
- Payroll Expense Journal History (PAMPYDT / PYRL_EXPS_JRNL_HST)
- Payroll Fringe Journal History (PAMPYDT / PYRL_FRNG_JRNL_HST)

Batch Parameters

Labor Additive Financial Selection

Parameter	Description	Default Value
Source Journal (SRC_JRNL)	The required Journal/Ledger ID of the input journal containing accounting results from payroll activity.	2
Cost Allocation Credit Object Code (CST_ALC_CR_OBJ_CD)	The required object code for the credit (offset) cash expenditure accounting lines.	<blank>
Cost Allocation ID (CST_ALC_ID)	The required ID used in the creation of both output XML files.	<blank>
Cost Allocation Series ID (CSL_ALC_SRS_ID)	The required ID used in the creation of both output XML files.	<blank>
Cost Allocation Step ID (CSL_ALC_STP_ID)	The required Cost Allocation Step ID used in the creation of both output XML files.	<blank>
Object Level for Selection (OBJ_RLP_FLD)	The required level for the selection. Valid values are OBJ_CD, OTYP_CD, OCLS_CD, OCAT_CD, or OGRP_CD.	<blank>
Object Selection Codes (OBJ_RLP_CD)	The required code(s) for the Object Level for Selection. Separate multiple values with a comma.	<blank>
Department Selection Code (SEL_DEPT_CD)	A conditionally required selection parameter of a single Department code when the Program Level for Selection is PROG_CD, PTYP_CD, or PGRP_CD.	<blank>
Program Level for Selection (PGM_RLP_FLD)	The required level for the selection. Valid values are PROG_CD, PTYP_CD, PCLS_CD, PCAT_CD, PGRP_CD.	<blank>
Program Selection Codes (PGM_RLP_CD)	The required code(s) for the Program Level for Selection. Separate multiple values with a comma.	<blank>
Payroll Posting Code (PYRL_PST_CD)	The required external cash expenditure posting code(s) for selection.	D014
From Date (FROM_DT)	The required starting date for selection from the Posted Date journal field. Please enter as MM/DD/YYYY.	<blank>
To Date (TO_DT)	The required ending date for selection from the Posted Date journal field. Please enter as MM/DD/YYYY.	<blank>
Select Block Size (SELECT_BLOCK)	A required performance parameter used to define the number of journal records selected for update to temporary table 1. If left blank, 1000 defaults.	1000

Parameter	Description	Default Value
Commit Block Size (COMMIT_BLOCK_SIZE)	A required performance parameter used to define the number of records saved to the XML file in one update. If left blank, 1000 defaults.	1000
Progression Message Block Size (PROG_CTR_SZ)	A required parameter used when logging messages to the job log. If left blank, 5000 defaults.	5000

Labor Additive HRM Selection

Parameter	Description	Default Value
Payroll Expense Transaction Codes (PYRL_EXP_DOC)	The required transaction code for normal expenditures used for look-up on the Payroll Expense and Fringe Journals. Multiple value, comma separated, accepted.	PREXP
Payroll Expense Correction Transaction Codes (PYRL_EXP_COR_DOC)	The required transaction code for expenditure corrections that used for look-up on the Payroll Expense and Fringe Journals. Multiple value, comma separated, accepted.	PREXC
Select Block Size (SELECT_BLOCK_SIZE)	A required performance parameter used to define the number of temporary table 3 records selected for update to the XML file. If left blank, 1000 defaults.	1000
Commit Block Size (COMMIT_BLOCK_SIZE)	A required performance parameter used to define the number of records saved to the XML file in one update. If left blank, 1000 defaults.	1000
Progression Message Block Size (PROG_CTR_SZ)	A required parameter used when logging messages to the job log. If left blank, 5000 defaults.	5000

Labor Additive Base Calculation

Parameter	Description	Default Value
Cost Allocation ID CST_ALC_ID	The required Cost Allocation ID for XML output.	<blank>
Cost Allocation Series ID CSL_ALC_SRS_ID	The required Cost Allocation Series ID for XML output.	<blank>
Cost Allocation Step ID CSL_ALC_STP_ID	The required Cost Allocation Step ID for XML output.	<blank>
Select Block Size (SELECT_BLOCK)	A required performance parameter used to define the number of temporary table 3 records selected for update to the XML file.	1000

Parameter	Description	Default Value
	If left blank, 1000 defaults.	
Commit Block Size (COMMIT_SIZE)	A required performance parameter used to define the number of records saved to the XML file in one update. If left blank, 1000 defaults.	1000
Progression Message Block Size (PROG_CTR_SZ)	A required parameter used when logging messages to the job log. If left blank, 5000 defaults.	5000

Labor Additive PLBS Prep

Parameter	Description	Default Value
Cost Allocation ID CST_ALC_ID	The required Cost Allocation ID for XML output.	<blank>
Cost Allocation Series ID CSL_ALC_SRS_ID	The required Cost Allocation Series ID for XML output.	<blank>
Cost Allocation Step ID CSL_ALC_STP_ID	The required Cost Allocation Step ID for XML output.	<blank>
Select Block Size (SELECT_BLOCK_SIZE)	A required performance parameter used to define the number of temporary table 3 records selected for update to the XML file. If left blank, 1000 defaults.	1000
Progression Message Block Size (PROG_CTR_SZ)	A required parameter used when logging messages to the job log. If left blank, 5000 defaults.	5000
XML File Path FILE_LOC	The required location where the PLBS (Pool/Base) XML output is written.	\$\$AMSROOT\$\$/Export/Import

Labor Additive PLBS Load

Parameter	Description	Default Value
Action Code (ACTN_CD)	The required action for table import.	201
File Path and Name (FILE_NM)	The required path and file name from where XML is retrieved to load.	\$\$AMSROOT\$\$/Export/Import/LBR_ADTV_R_PLBS_SETP.xml
Commit Block Size COMMIT_BLOCK	A required performance parameter used to define the number of records saved to the table in one update. If left blank, 200 defaults.	200

Labor Additive PBDIST Prep

Parameter	Description	Default Value
Cost Allocation Credit Object Code	The required object code for the credit	<blank>

Parameter	Description	Default Value
(CST_ALC_CR_OBJ_CD)	(offset) cash expenditure accounting lines.	
Cost Allocation ID (CST_ALC_ID)	The required Cost Allocation ID for XML output.	<blank>
Cost Allocation Series ID (CSL_ALC_SRS_ID)	The required Cost Allocation Series ID for XML output.	<blank>
Cost Allocation Step ID (CSL_ALC_STP_ID)	The required Cost Allocation Step ID for XML output.	<blank>
Select Block Size (SELECT_BLOCK)	A required performance parameter used to define the number of temporary table 3 records selected for update to the XML file. If left blank, 1000 defaults.	1000
Commit Block Size (COMMIT_BLOCK_SIZE)	A required performance parameter used to define the number of records saved to the XML file in one update. If left blank, 200 defaults.	200
Progression Message Block Size (PROG_CTR_SZ)	A required parameter used when logging messages to the job log. If left blank, 500 defaults.	500
XML File Path FILE_LOC	The required location where the PLBS (Pool/Base) XML output is written.	\$\$AMSR00T\$\$/ExportImport

Labor Additive PBDIST Load

Parameter	Description	Default Value
Action Code	The required	201

Parameter	Description	Default Value
(ACTN_CD)	action for table import.	
File Path and Name (FILE_NM)	The required path and file name from where XML is retrieved to load.	\$\$AMSR00T\$\$/ExportImport/LBR_ADTV_R_PLBS_DIST.xml
Commit Block Size COMMIT_BLOCK	A required performance parameter used to define the number of records saved to the table in one update. If left blank, 200 defaults.	200

Labor Additive History Table

N/A

Major Output

- Labor Additive Financial Section
 - Temporary Table 1 (LBR_ADTV_TMP1)
- Labor Additive HRM Selection
 - Temporary Table 2 (LBR_ADTV_TMP2)
- Labor Additive Base Calculation
 - Temporary Table 3 (LBR_ADTV_TMP3)
- Labor Additive PLBS Prep
 - LBR_ADTV_R_PLBS_SETP.xml
- Labor Additive PLBS Load
 - Pool Base Setup (PLBS / R_PLBS_SETP)
- Labor Additive PBDIST Prep
 - LBR_ADTV_R_PLBS_DIST.xml
- Labor Additive PBDIST Load
 - Pool Base Distribution (PBDIST / R_PLBS_DIST)
- Labor Additive History Table
 - Labor Additive History (LBR_ADTV_HIST)

Job Return Codes

The following table shows the potential return codes for the individual jobs within the Labor Additive chain. Note that the Chain job ends with the highest return code across all of the jobs.

Labor Additive Financial Section

Return Code	Condition	Recommendation
Successful (1)	All validations passed, journal records selected, and temporary table updated.	N/A
Warning (4)	No journal records selected.	Review parameters to see if they were correct if records were expected.
Non-Fatal Error (8)	N/A	N/A
Failed (12)	Job failed under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for unexpected situations. • Records already exist on Labor Additive History table for the time period. 	Investigate the reason for the job failure before scheduling a new job.
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of database server or network issues.	Investigate the reason for the System Failure before scheduling a new job.

Labor Additive HRM Selection

Return Code	Condition	Recommendation
Successful (1)	All validations passed, journal records selected, and temporary table updated.	N/A
Warning (4)	The job does not end with this return code.	N/A
Non-Fatal Error (8)	N/A	N/A
Failed (12)	Job failed under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for unexpected situations. 	Investigate the reason for the job failure before scheduling a new job.
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of database server or network issues.	Investigate the reason for the System Failure before scheduling a new job.

Labor Additive Base Calculation

Return Code	Condition	Recommendation
Successful (1)	All validations passed, fringe records selected, calculations done, and temporary table updated.	N/A
Warning (4)	The job does not end with this return code.	N/A
Non-Fatal Error (8)	N/A	N/A
Failed (12)	Job failed under the following conditions: <ul style="list-style-type: none"> Parameters are invalid. Run time exceptions for unexpected situations. 	Investigate the reason for the job failure before scheduling a new job.
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of database server or network issues.	Investigate the reason for the System Failure before scheduling a new job.

Labor Additive PLBS Prep

Return Code	Condition	Recommendation
Successful (1)	All validations passed and XML file created.	N/A
Warning (4)	The job does not end with this return code.	N/A
Non-Fatal Error (8)	N/A	N/A
Failed (12)	Job failed under the following conditions: <ul style="list-style-type: none"> Parameters are invalid. Run time exceptions for unexpected situations. 	Investigate the reason for the job failure before scheduling a new job.
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of database server or network issues.	Investigate the reason for the System Failure before scheduling a new job.

Labor Additive PLBS Load

Return Code	Condition	Recommendation
Successful (1)	All validations passed and XML file loaded.	N/A
Warning (4)	The job does not end with this return code.	N/A
Non-Fatal Error (8)	N/A	N/A
Failed (12)	Job failed under the following conditions: <ul style="list-style-type: none"> Parameters are invalid. Missing pool record on PLBS or 	Investigate the reason for the job failure before scheduling a new job.

Return Code	Condition	Recommendation
	PBDIST. <ul style="list-style-type: none"> Run time exceptions for unexpected situations. 	
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of database server or network issues.	Investigate the reason for the System Failure before scheduling a new job.

Labor Additive PBDIST Prep

Return Code	Condition	Recommendation
Successful (1)	All validations passed and XML file created.	N/A
Warning (4)	The job does not end with this return code.	N/A
Non-Fatal Error (8)	N/A	N/A
Failed (12)	Job failed under the following conditions: <ul style="list-style-type: none"> Parameters are invalid. Run time exceptions for unexpected situations. 	Investigate the reason for the job failure before scheduling a new job.
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of database server or network issues.	Investigate the reason for the System Failure before scheduling a new job.

Labor Additive PBDIST Load

Return Code	Condition	Recommendation
Successful (1)	All validations passed and XML file loaded.	N/A
Warning (4)	The job does not end with this return code.	N/A
Non-Fatal Error (8)	N/A	N/A
Failed (12)	Job failed under the following conditions: <ul style="list-style-type: none"> Parameters are invalid. Missing pool record on PLBS or PBDIST. Run time exceptions for unexpected situations. 	Investigate the reason for the job failure before scheduling a new job.
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of	Investigate the reason for the System Failure before

Return Code	Condition	Recommendation
	database server or network issues.	scheduling a new job.

Labor Additive History Table

Return Code	Condition	Recommendation
Successful (1)	All validations passed and table updated.	N/A
Warning (4)	The job does not end with this return code.	N/A
Non-Fatal Error (8)	N/A	N/A
Failed (12)	Job failed under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for unexpected situations. 	Investigate the reason for the job failure before scheduling a new job.
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of database server or network issues.	Investigate the reason for the System Failure before scheduling a new job.

Sort Criteria

- Temporary Table 1 – Transaction Code, Transaction Department Code, Transaction ID, Version Number, and Accounting Line
- Temporary Table 2 – Employee id and GTN run number while updating records for Fringe
- Temporary Table 3 – N/A
- PLBS XML File – N/A
- PBDIST XML File – N/A
- Labor Additive History Table – N/A

Selection Criteria

- Cost Accounting Journal – Uses all selection parameters in Labor Additive FIN Selection job step.
- Fringe Expense Journal – Matches on Transaction Code, Transaction ID, and Accounting Line 1
- Payroll Expense Journal – Matches on Employee ID, GTN Run Number

2.1.15 Labor Additive Reversal

Chain or Job Name	Labor Additive Reversal
Recommended Frequency	On demand as needed
Single Instance Required	Yes
Can be restarted?	Refer to the Restart Information section
Reports generated	No

Overview

The Labor Additive Reversal job enables re-running the Labor Additive chain when the initial data load is incorrect. The job deletes Labor Additive History records for the specified period so the Labor Additive chain can be run again for the same timeframe.

Restart Information

- If the job failed due to any reason, schedule a new job after correcting the errors that caused the job to fail.

Major Input

Tables

- Labor Additive History (LBR_ADTV_HIST)

Batch Parameters

Parameter	Description	Default Value
From Date (FROM_DT)	The required starting Posted Date for deleting History records. Please enter as MM/DD/YYYY.	<blank>
To Date (TO_DT)	The required ending Posted Date for deleting History records. Please enter as MM/DD/YYYY.	<blank>

Major Output

- Labor Additive History (LBR_ADTV_HIST)

Job Return Codes

The following table shows the potential return codes for the Labor Additive Reversal job:

Return Code	Condition	Recommendation
Successful (1)	All validations passed, journal records selected, and temporary table updated.	N/A

Return Code	Condition	Recommendation
Warning (4)	No history records selected.	Review parameters to see if they are correct, if records were expected.
Non-Fatal Error (8)	N/A	N/A
Failed (12)	Job failed under the following conditions: <ul style="list-style-type: none"> Parameters are invalid. Run time exceptions for unexpected situations. 	Investigate the reason for the job failure before scheduling a new job.
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of database server or network issues.	Investigate the reason for the System Failure before scheduling a new job.

Sort Criteria

- Labor Additive History Table – N/A

Selection Criteria

- Labor Additive History Table – Posted date field.

2.1.16 Labor Cost Distribution Load to Cost Allocation

Chain Name	Labor Cost Distribution Load to Cost Allocation
Recommended Frequency	As needed when payroll data becomes available
Single Instance Required	Yes
Can be restarted?	See individual job steps
Reports generated	No

Overview

The Labor Cost Distribution Load to Cost Allocation chain is a group of jobs that work together to automatically modify various Cost Allocation configurations based on the latest time and labor distribution information available in Labor Cost Distribution History (referred to as LCDH, please note this is not a page code) produced from Advantage payroll processing. The process will perform updates to the following Cost Allocation configurations: Statistical Units (STAT), Pool/Base Setup (PLBS), and Pool/Base Distribution (PBDIST). This process allows the allocations costs on a department-wide basis (the most prevalent method) and also on a specified Chart of Account breakdown (element) basis, as defined in the COA Breakdown parameter.

The Labor Cost Distribution Load to Cost Allocation has the following jobs (each of the jobs listed below is described in subsequent sections):

1. Select LCDH Records
2. Build LCDHT Records
3. Build STAT XML
4. Build PLBS XML
5. Build PBDIST XML
6. Delete Cost Allocation Records
7. Load STAT XML
8. Load PLBS XML
9. Load PBDIST XML

Major Input

- Labor Cost Distribution History - HRM (LABR_COST_DIST_HST)
- Cost Allocation Control Setup (R_CSAL_CTRL_SETP/ ALOC)

Major Output

- Labor Cost Distribution History Temp - FIN (LABR_COST_DIST_HST_TEMP)
- Pool Base (R_PLBS_SETP)
- Pool Base Distribution (R_PLBS_DIST)

- Statistics (R_STAT_UNIT_SETP)

Chain Job Return code

The following table shows the potential return codes for the Labor Cost Distribution Load to Cost Allocation chain job. Note that the Chain job will end with the highest return code across all of the jobs.

Return Code	Condition
Successful (1)	All of the jobs end successfully
Warning (4)	One of the jobs in the chain ended as <i>Warning</i>
Non-Fatal Error (8)	One of the jobs in the chain ended as <i>Non-Fatal Error</i>
Failed (12)	One of the jobs in the chain ended as <i>Failed</i>
Terminated (16)	One of the jobs in the chain ended as <i>Terminated</i>
System Failure (20)	One of the jobs in the chain ended as <i>System Failure</i>

Problem Resolution

If any of the jobs in the chain failed due to application errors it is advisable to reschedule the job. Please refer to the individual jobs for details regarding the specific job processes and problem resolution.

Labor Cost Distribution Load to Cost Allocation: Select LCDH Records

Job Name	Select LCDH Records
Recommended Frequency	See chain section
Single Instance Required	No
Can be restarted?	No
Reports Generated	No.

Overview

The Select LCDH Records job selects records from Labor Cost Distribution History (LCDH) based on the following criteria:

- Department value on LCDH records is in the list of eligible Departments specified in the batch parameter (Department).
- LCDH records have at least one Event Type which is in the list of eligible Time Events specified in the batch parameter (Time Events).
- LCDH records where in the Month (MM) and Year (YYYY) of the Pay Period Start Date match with the batch parameter (Month and Year of Pay Period Start Date).

The following table shows the various steps that the Select LCDH Records Job goes through and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> Validating Batch Parameters Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value is displayed in the log. Batch Parameter validation completed.
2. Deletion of Records	<ul style="list-style-type: none"> Job deletes existing records from LABR_COST_DIST_HST_TEMP so that it can retrieve the latest data from HRM.
3. Selection of Records	<ul style="list-style-type: none"> Selection of record from Labor Cost Distribution History in Advantage HRM based on the values entered in the job parameters.

Input

- Labor Cost Distribution History – HRM (LABR_COST_DIST_HST_TEMP)
- Cost Allocation Control Setup (R_CSAL_CTRL_SETP)

Batch Parameters

Parameter	Description	Default Value
Allocation ID (ALOC_ID)	A required Cost Allocation ID used for cost allocation record selection.	(blank)
Export Location (AMSEXPORT)	A required location to write and retrieve the XML files created for STAT,PLBS, and PBDIST	\$\$AMSR00T\$\$ /ExportImport
Parameter File Location (AMSPARM)	A required location to write out a parameter file for later job steps to retrieve and use.	\$\$AMSR00T\$\$ /Parms
COA Breakdown (COA_BRKDWN)	A conditionally required parameter (if Mode of Allocation parameter is 2) to indicate the COA used for selection and grouping. Only one value is allowed. Valid values include: UNIT_CD, FUND_CD, APPR_CD, FUNC_CD, PROG_CD, PHASE_CD, and PPC_CD.	(blank)
Commit Block Size (COMMIT_BLOCK_SIZE)	A required performance parameter used to control the number of records committed in an update. If left blank, 1000 defaults.	1000
Selected Departments DEPT_CD	A required selection parameter used in selecting LCDH records. Care should be taken that any value or values (separated by commas) match the setup within the Allocation ID.	(blank)

Parameter	Description	Default Value
	There is no edit between the Department parameter and the Department field on Cost Allocation Control ID (ALOC).	
Mode of Allocation MODE_OF_ALLOC	A required indication of how the process will select and summarize Labor Cost Distribution History records. (1 = Department wide, 2 = COA breakdown)	(blank)
Pay Period End Date (PPRD_END_DT)	A required selection month and year (MM/YYYY) used in the selection of Labor Cost Distribution History records.	(blank)
Progression Counter (PROG_CTR_SIZE)	A required block size used when writing incremental progress messages in the job log. If left blank, 100 defaults.	100
Select Block Size SELECT_BLOCK	A required performance parameter used to control the number of records selected in a processing instance. If left blank, 100 defaults.	100
Step Number (STEP_NO)	A required Step Number from the STEP page for the Allocation ID parameter value. Only a single value is permitted. A later job will use this and the Allocation ID to select one or more Series.	(blank)
Time Events TIME_EVENT_CD	A required selection parameter of one or more Time Event Codes for LCDH record selection. Separate multiple values with a comma.	(blank)

Output

- Labor Cost Distribution History Temp - FIN (LABR_COST_DIST_HST_TEMP)
- LCDHParams.txt parameter file

Job Return code

The following table shows the potential job return codes for the Select LCDH Records job.

Return Code	Condition
Successful (1)	All of the selected payment records are processed successfully.
Warning (4)	No eligible records found. This could be because of the following reasons: <ul style="list-style-type: none"> • No LCDH records exist • No LCDH records match selection criteria - "No records

Return Code	Condition
	found on LCDH table which meets the selection criteria.”
Non-Fatal Error (8)	This job does not use this return code.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid • Failure while committing records • Run time exceptions for unexpected situations
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with the return code of <i>Terminated</i> , subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with the return code of <i>System Failure</i> , subsequent jobs in the chain are set to inactive.

Sort Sequence

The LCDH records pulled from HRM are sorted by the following fields, depending on which allocation mode was chosen (Department Wide or COA Breakdown):

Department Wide

- Minutes
- Budget Fiscal Year
- Department
- Unit
- Fund
- Sub Fund
- Appropriation

COA Breakdown

- Budget Fiscal Year
- Department
- Additional value listed/specified in the COA Breakdown job parameter

Selection Criteria

The Select LCDH Records job will retrieve records from Labor Cost Distribution History in Advantage HRM based on the job parameters entered:

- Selected Departments
- Time Event
- Pay Period End Date

Problem Resolution

There is no need to back out any updates if this job fails in any of the steps since the job only updates the temporary table (LABR_COST_DIST_HST_TEMP) which gets cleared when the job is rescheduled the next time. If the job, after writing the records to the temporary table, ends with a return code of *Failed*, *Terminated* or *System Failure* and another instance of the job has already been scheduled and run successfully, then this job should not be restarted; it should only be rescheduled.

The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	This step does not issue this return code.
Non-Fatal Error (8)	N/A	This step does not issue this return code.	This step does not issue this return code.
Failed (12)	Entered parameters are not valid. Sample Message: Department xxxx (xxxx being the value from the parameter) is not a valid Department.	Enter the correct Department and restart the job.	Job should be rescheduled after correcting the parameters.
	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Step 2: Deletion of Records:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	This step does not issue this return code.
Non-Fatal Error (8)	N/A	This step does not issue this return code.	This step does not issue this return code.
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Step 3: Selection of Records:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	The query to LCDH returns no records.	Verify that the job parameters entered match existing records in the HRM table.	Re-schedule a new job after verifying parameter values.
Non-Fatal Error (8)	N/A	This step does not issue this return code.	This step does not issue this return code.
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	

Possible Return Codes	Condition	Recommendation	Other Instructions
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Labor Cost Distribution Load to Cost Allocation: Build LCDHT Records

Job Name	Build LCDHT Records
Recommended Frequency	See chain section
Single Instance Required	No
Can be restarted?	No
Reports Generated	No.

Overview

The Build LCDHT Records job will take the records previously selected from LCDH and Sort/Summarize the records. The number of rows may be reduced as a result of the summarization, whereby records with matching identifying information are combined into a single record, adding up the minutes for all the matching records.

- The LABR_COST_DIST_HST_TEMP table (abbreviated henceforth as LCDHT) will have the following columns to be updated with values - Minutes, Budget Fiscal Year, Department, Unit, Fund, Appropriation Unit, Function, Program, Phase, Program Period, Cost Allocation Series ID.
- Once records are available on LCDHT, they will be summarized based on Department, COA Breakdown parameter value. While summarizing the records, the respective Minutes column values will be summed up for the lines which are summarized. Other columns (including other COA elements) will be ignored during this process. Summarized records will be sorted and arranged in ascending order based on Department value for Department-wide allocation. If the Mode of Allocation is set to “2”, then sorting will be based on Department and a specified COA breakdown value, as defined in the COA Breakdown job parameter.
- Summarized and sorted LCDHT records will be used to determine the Series ID value for each record. For department wide allocation, multiple Series IDs would be determined for LCDHT records. Each LCDHT record’s Department value will be the respective Series ID value. For a specified COA breakdown code basis allocation, Series ID will be determined

based on Allocation ID and the specified COA breakdown value. System will navigate to Cost Allocation Control Setup (ALOC) based on Allocation ID and pull down all the available Series IDs from Cost Allocation Series Setup (SRS). Each Series ID, Allocation ID and Step Number will be used to identify the respective Pool record on Pool Base Distribution (PBDIST). From the Pool record(s), the specified COA breakdown value will be derived for the given Allocation ID, Series ID and Step No. System will allocate the Series ID if the specified COA breakdown value of the Pool is matching LCDHT record. Series ID column value cannot be blank for any LCDHT record. If Series ID value is blank (null), then the system removes that record from LCDHT. The job will log an informational message in job logs for the missing record and continue processing.

Process Steps	Messages
1. Parameter- Validation	<ul style="list-style-type: none"> Validating Batch Parameters Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. Batch Parameter validation completed.
2. Deletion of Records	<ul style="list-style-type: none"> Job will delete the temporary records as it prepares to modify/re-insert the records into LCDHT
3. Summary/Sort of Records	<ul style="list-style-type: none"> Summarized Selection of LCDHT records based on similar identifier fields. Records are sorted by Department code and any COA values specified in the Select LCDH Record job's COA Breakdown parameter.

Input

- Labor Cost Distribution History Temp - FIN (LABR_COST_DIST_HST_TEMP)
- Pool Base Distribution (R_PLBS_DIST / PBDIST)
- Cost Allocation Series Setup (R_CSAL_SRCTR_SETP / SRS)
- LCDHParams.txt file

Batch Parameters

Parameter	Description	Default Value
AMSPARM	The required location from which the parameter file created in the first job will be retrieved.	\$\$AMSR00T\$\$ /Parms
PROG_CTR_SIZE	A required block size used when writing incremental progress messages in the job log. If left blank, 100 defaults.	100
SELECT_BLOCK	A required performance parameter used to control the number of records selected in a processing instance. If left blank, 100 defaults.	100

Output

- Labor Cost Distribution History Temp - FIN (LABR_COST_DIST_HST_TEMP)

Job Return code

The following table shows the potential job return codes for the Build LCDHT Records job.

Return Code	Condition
Successful (1)	All of the selected payment records are processed successfully.
Warning (4)	No records exist or could be found on LCDHT for processing.
Non-Fatal Error (8)	This job step does not use this return code.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid • Failure while committing records • Run time exceptions for unexpected situations
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with the Return Code of <i>Terminated</i> , subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with the Return Code of <i>System Failure</i> , subsequent jobs in the chain will be set to inactive.

Sort Sequence

Please see the sort sequence from the previous job step as it is the same with this job.

Selection Criteria

The Build LCDHT Records job will take all of the records found in LCDHT and summarize/sort records as needed before re-writing them to the table.

Problem Resolution

There is no need to back out any updates if this job fails in any of the steps since the job only updates the temporary table (LABR_COST_DIST_HST_TEMP) which gets cleared when the job is rescheduled the next time. If the job, after writing the records to the temporary table, ends with a return code of *Failed*, *Terminated* or *System Failure* and another instance of the job has already been scheduled and run successfully, then this job should not be restarted; it should only be rescheduled.

The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	This step does not issue this return code.
Non-Fatal Error (8)	N/A	This step does not issue this return code.	This step does not issue this return code.
Failed (12)	Entered Parameters are not valid Sample Message: Select Block should be a positive integer	Enter a valid positive integer for the Select Block size	Job should be rescheduled after correcting the parameters.
	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Step 2: Deletion of Records:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	This step does not issue this return code.
Non-Fatal Error (8)	N/A	This step does not issue this return code.	This step does not issue this return code.

Possible Return Codes	Condition	Recommendation	Other Instructions
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Step 3: Selection of Records:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	The query to LCDHT returns no records.	Verify that there is data first in the LCDHT table.	Run the Select LCDH Records job first to populate LCDHT with records from LCDH in HRM.
Non-Fatal Error (8)	N/A	This step does not issue this return code.	This step does not issue this return code.
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure	When the job is terminated because of database server or	The reason for the System Failure needs to be investigated. The	

Possible Return Codes	Condition	Recommendation	Other Instructions
(20)	network issues.	job can either be restarted or rescheduled.	

Labor Cost Distribution Load to Cost Allocation: Build STAT xml

Job Name	Build STAT xml
Recommended Frequency	See chain section
Single Instance Required	No
Can be restarted?	No
Reports Generated	No

Overview

The Build STAT xml job takes the records previously sorted/summarized from LCDH and summarizes them based on what will be new base records on Pool/Base setup into the LABR_COST_DIST_HST_TEMP table (abbreviated onward as LCDHT) and resulting records are then used to create an XML file for Statistical Unit (R_STAT_UNIT_SETP / STAT). Any records written to LCDHT that are not associated with a Series ID are deleted before that summarization. Summarization is done on Department, Unit, Fund, Appropriation Unit, Function, Program, Phase, Program Period and ordered by the Cost Allocation Series ID. That Series ID will become the Statistical Group. The total number of summarized minutes will become the Statistical Unit.

Process Steps	Messages
1. Parameter- Validation	Parameters supplied are listed. <ul style="list-style-type: none"> Validating batch parameters If a parameter is invalid, the invalid value will be displayed in the log with the error. <ul style="list-style-type: none"> Parameter validation complete.
2. Deletion of Records	<ul style="list-style-type: none"> Executing deletion of records from LCDHT the table where Series ID is null. ## records were deleted from LCDHT
3. Selection	<ul style="list-style-type: none"> Executing query on LCDH Temp table Query execution on LCDH Temp table completed.
4. Summarization	This step does not update the job log.
5. File Creation	<ul style="list-style-type: none"> STAT XML file created successfully.

Input

- Labor Cost Distribution History - FIN (LABR_COST_DIST_HST)
- Cost Allocation Series Setup (R_CSAL_SRCTR_SETUP / SRS)

Batch Parameters

Parameter	Description	Default Value
AMSPARM	The required location from which the parameter file created in the first job will be retrieved.	\$\$AMSR00T\$\$ /Parms
PROG_CTR_SIZE	A required block size used when writing incremental progress messages in the job log. If left blank, 100 defaults.	100
SELECT_BLOCK	A required performance parameter used to control the number of records selected in a processing instance. If left blank, 100 defaults.	100

Output

- Labor Cost Distribution History Temp - FIN (LABR_COST_DIST_HST_TEMP)
- STAT.xml

Job Return code

The following table shows the potential job return codes for the Build STAT xml.job.

Return Code	Condition
Successful (1)	All of the selected payment records are processed successfully.
Warning (4)	Could not find LCDHT records to create STAT XML file.
Non-Fatal Error (8)	This job step does not use this return code.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid • Failure while committing records • Run time exceptions for unexpected situations
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with the Return Code of <i>Terminated</i> , subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with the Return Code of <i>System Failure</i> , subsequent jobs in the chain will be set to inactive.

Sort Sequence

The LCDH records pulled from previous job are sorted by Series ID.

Selection Criteria

The Build STAT xml job will take all of the records found in LCDHT and update the xml as needed.

Problem Resolution

There is no need to back out any updates if this job fails in any of the steps. If the job, after writing the records to the temporary table, ends with a return code of *Failed*, *Terminated* or *System Failure*, then a new instance of the chain should be scheduled starting with this 3rd job step. Restarting is not an option.

The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	
Warning (4)	N/A	This step does not issue this return code.	
Non-Fatal Error (8)	N/A	This step does not issue this return code.	
Failed (12)	Entered Parameters are not valid Sample Message: Select Block should be a positive integer	Enter a valid positive integer for the Select Block size	Job should be rescheduled after correcting the parameters.
	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before scheduling another instance of the chain that starts with this job step.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling another instance of the chain that starts with this job step.	

Possible Return Codes	Condition	Recommendation	Other Instructions
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before scheduling another instance of the chain that starts with this job step.	

Step 2: Deletion of Records:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	
Warning (4)	N/A	This step does not issue this return code.	
Non-Fatal Error (8)	N/A	This step does not issue this return code.	
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Step 3: Selection of Records:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Warning (4)	N/A	This step does not issue this return code.	
Non-Fatal Error (8)	N/A	This step does not issue this return code.	
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Step 4: Summarization:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	
Non-Fatal Error (8)	N/A	This step does not issue this return code.	
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure	When the job is terminated because of database server or	The reason for the System Failure needs to be investigated. The	

Possible Return Codes	Condition	Recommendation	Other Instructions
(20)	network issues.	job can either be restarted or rescheduled.	

Step 5: File creation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	
Non-Fatal Error (8)	N/A	This step does not issue this return code.	
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Labor Cost Distribution Load to Cost Allocation: Build PLBS xml

Job Name	Build PLBS xml
Recommended Frequency	See chain section
Single Instance Required	No
Can be restarted?	No
Reports Generated	No

Overview

The Build PLBS xml job takes the records previously sorted/summarized from LCDH and summarizes them based on what will be new base records on Pool/Base setup into the LABR_COST_DIST_HST_TEMP table (abbreviated onward as LCDHT) and resulting records are then used to create an XML file for Pool Base Setup (R_PLBS_SETP / PLBS).

Those PLBS records are then created based on the following columns to be updated with values – Allocation ID, Step No, Pool Base Indicator, Pool Base Sequence Number, Statistical Record ID Cost Allocation Series ID. The resulting records are then used to create an XML file.

Process Steps	Parameters supplied are listed.
1. Parameter Validation	Parameters supplied are listed. <ul style="list-style-type: none"> Validating batch parameters If a parameter is invalid, the invalid value will be displayed in the log with the error. <ul style="list-style-type: none"> Parameter validation complete.
2. Selection of Records	<ul style="list-style-type: none"> Executing query on LCDH Temp table Query execution on LCDH Temp table completed.
3. File Creation	<ul style="list-style-type: none"> PLBS XML file created successfully.

Input

- Labor Cost Distribution History - FIN (LABR_COST_DIST_HST)
- Cost Allocation Series Setup (R_CSAL_SRCTR_SETP / SRS)

Batch Parameters

Parameter	Description	Default Value
AMSEXPORT	The required location from which the parameter file created in the first job will be retrieved.	\$\$AMSR00T\$\$ /ExportImport
AMSPARM	The required location from which the parameter file created in the first job will be retrieved.	\$\$AMSR00T\$\$ /Parms
PROG_CTR_SIZE	A required block size used when writing incremental progress messages in the job log. If left blank, 100 defaults.	100
SELECT_BLOCK	A required performance parameter used to control the number of records selected in a processing instance. If left blank, 100 defaults.	100

Output

- Labor Cost Distribution History Temp - FIN (LABR_COST_DIST_HST_TEMP)
- PLBS.xml

Job Return code

The following table shows the potential job return codes for the Build PLBS xml job.

Return Code	Condition
Successful (1)	All of the selected payment records are processed successfully.
Warning (4)	Could not find LCDHT records to create PLBS XML file.
Non-Fatal Error (8)	This job step does not use this return code.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid • Failure while committing records • Run time exceptions for unexpected situations
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with the Return Code of <i>Terminated</i> , subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with the Return Code of <i>System Failure</i> , subsequent jobs in the chain will be set to inactive.

Sort Sequence

The LCDHT records pulled from the Build LCDHT Records job.

Selection Criteria

The Build PLBS xml job will take all of the records found in LCDHT and update the xml as needed.

Problem Resolution

There is no need to back out any updates if this job fails in any of the steps. If the job, after writing the records to the temporary table, ends with a return code of *Failed*, *Terminated* or *System Failure*, then a new instance of the chain should be scheduled starting with this 4th job step. Restarting is not an option.

The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	

Possible Return Codes	Condition	Recommendation	Other Instructions
Warning (4)	N/A	This step does not issue this return code.	
Non-Fatal Error (8)	N/A	This step does not issue this return code.	
Failed (12)	Entered Parameters are not valid Sample Message: Select Block should be a positive integer	Enter a valid positive integer for the Select Block size	Job should be rescheduled after correcting the parameters.
	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before scheduling another instance of the chain that starts with this job step.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling another instance of the chain that starts with this job step.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before scheduling another instance of the chain that starts with this job step.	

Step 2: Selection of Records:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	
Warning (4)	N/A	This step does not issue this return code.	
Non-Fatal Error (8)	N/A	This step does not issue this return code.	

Possible Return Codes	Condition	Recommendation	Other Instructions
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Step 3: File creation:

Possible Return Codes		Condition	Recommendation	Other Instructions
Successful (1)		Successful	N/A	
Warning (4)		N/A	This step does not issue this return code.	
Non-Fatal Error (8)		N/A	This step does not issue this return code.	
Failed (12)		Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)		Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)		When the job is terminated because of database server or network	The reason for the System Failure needs to be investigated. The job can either be restarted or	

Possible Return Codes		Condition	Recommendation	Other Instructions
		issues.	rescheduled.	

Labor Cost Distribution Load to Cost Allocation: Build PBDIST xml

Job Name	Build PBDIST xml
Recommended Frequency	See chain section
Single Instance Required	No
Can be restarted?	No
Reports Generated	No.

Overview

The Build PBDIST xml job takes the records previously sorted/summarized from LCDH and summarizes them based on what will be new base records on Pool/Base setup into the LABR_COST_DIST_HST_TEMP table (abbreviated onward as LCDHT) and resulting records are then used to create an XML file for Pool Base Distribution (R_PBDIST_DIST / PBDIST).

Process Steps	Messages
1. Parameter Validation	Parameters supplied are listed. <ul style="list-style-type: none"> Validating batch parameters If a parameter is invalid, the invalid value will be displayed in the log with the error. <ul style="list-style-type: none"> Parameter validation complete.
2. Selection of Records	<ul style="list-style-type: none"> Executing query on LCDH Temp table Query execution on LCDH Temp table completed.
3. File Creation	<ul style="list-style-type: none"> PLBS XML file created successfully.

Input

- Labor Cost Distribution History - FIN (LABR_COST_DIST_HST)
- Pool Base Distribution (R_PBDIST_DIST / PBDIST)

Batch Parameters

Parameter	Description	Default Value
AMSPARM	The required location from which the parameter	\$\$AMSROOT\$\$

Parameter	Description	Default Value
	file created in the first job will be retrieved.	/Parms
PROG_CTR_SIZE	A required block size used when writing incremental progress messages in the job log. If left blank, 100 defaults.	100
SELECT_BLOCK	A required performance parameter used to control the number of records selected in a processing instance. If left blank, 100 defaults.	100

Output

- Labor Cost Distribution History Temp - FIN (LABR_COST_DIST_HST_TEMP)
- PBDIST.xml

Job Return code

The following table shows the potential job return codes for the Build PBDIST xml job.

Return Code	Condition
Successful (1)	All of the selected payment records are processed successfully.
Warning (4)	Could not find LCDHT records to create PBDIST XML file.
Non-Fatal Error (8)	This job step does not use this return code.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid • Failure while committing records • Run time exceptions for unexpected situations
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with the Return Code of <i>Terminated</i> , subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with the Return Code of <i>System Failure</i> , subsequent jobs in the chain will be set to inactive.

Sort Sequence

The LCDH records pulled from Build LCDHT Records job.

Selection Criteria

The Build PBDIST xml job will take all of the records found in LCDHT and update the xml as needed.

Problem Resolution

There is no need to back out any updates if this job fails in any of the steps. If the job, after writing the records to the temporary table, ends with a return code of Failed, Terminated or System Failure then a new instance of the chain should be scheduled starting with this 5th job step. Restarting is not an option.

The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	
Warning (4)	N/A	This step does not issue this return code.	
Non-Fatal Error (8)	N/A	This step does not issue this return code.	
Failed (12)	Entered Parameters are not valid Sample Message: Select Block should be a positive integer	Enter a valid positive integer for the Select Block size	Job should be rescheduled after correcting the parameters.
	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before scheduling another instance of the chain that starts with this job step.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling another instance of the chain that starts with this job step.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before scheduling another instance of the chain that starts with this job step.	

Step 2: Summarization:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	
Warning (4)	N/A	This step does not issue this return code.	
Non-Fatal Error (8)	N/A	This step does not issue this return code.	
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Step 3: File Creation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	
Warning (4)	N/A	This step does not issue this return code.	
Non-Fatal Error (8)	N/A	This step does not issue this return code.	
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted	

Possible Return Codes	Condition	Recommendation	Other Instructions
		or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Labor Cost Distribution Load to Cost Allocation: Delete records from PBDIST PLBS and STAT table

Job Name	Delete records from PBDIST PLBS and STAT table
Recommended Frequency	See chain section
Single Instance Required	No
Can be restarted?	No
Reports Generated	No.

Overview

The Delete records from PBDIST PLBS and STAT table job takes the records previously used in building the xml of STAT, PLBS, PBDIST and LCDHT records and purges them based on their respective parameters.

Process Steps	Messages
Parameter Validation	Parameters supplied are listed. <ul style="list-style-type: none"> Validating batch parameters If a parameter is invalid, the invalid value will be displayed in the log with the error. <ul style="list-style-type: none"> Parameter validation complete.
Selection of Records	<ul style="list-style-type: none"> About to find records inside of Labor Cost Dist History Temp table About to walk through records inside of Labor Cost Dist History Temp table
Deletion of Records	<ul style="list-style-type: none"> About to purge records from R_PLBS_DIST Completed purging records from R_PLBS_DIST: number purged # About to purge records from PLBS Setup Purged records from PLBS Setup: number purged #

Process Steps	Messages
	<ul style="list-style-type: none"> About to purge records from Stat Unit Setup Purged records from Stat Unit Setup: number purged #

Input

- Labor Cost Distribution History Temp - FIN (LABR_COST_DIST_HST_TEMP)
- Pool Base Distribution (R_PBDIST_DIST / PBDIST)
- Statistics (R_STAT_UNIT_SETUP/STAT)
- Pool Base Setup(R_PLBS_SETUP/PLBS)

Batch Parameters

Parameter	Description	Default Value
AMSEXPORT	The required location from which the parameter file created in the first job will be retrieved.	\$\$AMSROOT\$\$ /ExportImport
AMSPARM	The required location from which the parameter file created in the first job will be retrieved.	\$\$AMSROOT\$\$ /Parms
SELECT_BLOCK	A required performance parameter used to control the number of records selected in a processing instance. If left blank, 100 defaults.	100

Output

- Labor Cost Distribution History Temp - FIN (LABR_COST_DIST_HST_TEMP)
- Pool Base Distribution (R_PBDIST_DIST / PBDIST)
- Statistics (R_STAT_UNIT_SETUP/STAT)
- Pool Base Setup (R_PLBS_SETUP/PLBS)

Job Return code

The following table shows the potential job return codes for the Delete records from PBDIST PLBS and Stat table job.

Return Code	Condition
Successful (1)	All of the selected payment records are processed successfully.
Warning (4)	N/A
Non-Fatal Error (8)	This job step does not use this return code.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> Parameters are invalid Failure while committing records

Return Code	Condition
	<ul style="list-style-type: none"> Run time exceptions for unexpected situations
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with the Return Code of <i>Terminated</i> , subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with the Return Code of <i>System Failure</i> , subsequent jobs in the chain will be set to inactive.

Sort Sequence

None

Selection Criteria

The Delete records from PBDIST PLBS and STAT table job will take all of the records found in LCDHT and purge all the records from LCDHT, PBDIST, PLBS and STAT.

Problem Resolution

There is no need to back out any updates if this job fails in any of the steps. If the job, after writing the records to the temporary table, ends with a return code of *Failed*, *Terminated* or *System Failure* then a new instance of the chain should be scheduled starting with this 6th job step. Restarting is not an option.

The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	
Warning (4)	N/A	This step does not issue this return code.	
Non-Fatal Error (8)	N/A	This step does not issue this return code.	
Failed (12)	Entered parameters are not valid. Sample Message: Select Block should be a positive integer	Enter a valid positive integer for the Select Block size	Job should be rescheduled after correcting the parameters.
	Failed because of runtime exceptions for an	The reason for the failure needs to be	

Possible Return Codes	Condition	Recommendation	Other Instructions
	unexpected situation.	investigated before scheduling another instance of the chain that starts with this job step.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling another instance of the chain that starts with this job step.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before scheduling another instance of the chain that starts with this job step.	

Step 2: Selection of Records:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	Warning while selecting if no records available while purging in the LCDHT, PBDIST, PLBS and STAT table.	Check logs.	Run the Build STAT xml, Build PLBS xml, and Build PBDISTxml jobs first to create respective XML.
Non-Fatal Error (8)	N/A	This step does not issue this return code.	
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted	

Possible Return Codes	Condition	Recommendation	Other Instructions
		or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Step 3: Deletion of Records:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	
Warning (4)	N/A	This step does not issue this return code.	
Non-Fatal Error (8)	N/A	This step does not issue this return code.	
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

The last three job steps of this chain job use SMU to load the XML files created by the previous job steps within the same chain. For general errors and recommendations, refer to the SMU Transaction Upload run sheet in the *CGI Advantage Financial – Utilities Run Sheet Guide*.

Labor Cost Distribution Load to Cost Allocation: Load STAT XML

The job uses System Maintenance Utility (SMU) to load the Statistics (STAT) records.

Labor Cost Distribution Load to Cost Allocation: Load PLBS XML

The job uses System Maintenance Utility (SMU) to load the Pool Base Setup (PLBS) records.

Labor Cost Distribution Load to Cost Allocation: Load PBDIST XML

The job uses System Maintenance Utility (SMU) to load the Pool Base Distribution (PBDIST) records.

2.1.17 Maximo Other Cost Batch Job

Job Name	Maximo Other Cost Batch Job
Recommended Frequency	The job can be run daily as part of the nightly cycle or on demand.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

The Maximo Other Cost process in Advantage Financial is used to track costs associated with Maximo Work Orders that can accumulate in CGI Advantage and have not been recorded in Maximo by either purchasing or disbursement transactions. These “other costs” need to be transmitted to Maximo, so that the Work Order totals in both systems will be accurate.

The following transaction categories are places where the “other costs” can be recorded:

Transaction Category	Transaction Category Name	Selection Logic	Notes
ABS	Accounting Based Spending	Based on Posting code Exclude entries that originate from Maximo.	Maximo entries are generated for some inventory transactions.
AP	Accounts Payable	PRCI only	CECI transactions are not selected.
AR	Accounts Receivable	Select Vendor Refund entries	
CA	Cost Accounting	Based on Posting code	
FA	Fixed Asset	Based on Posting code – should be depreciation only	
INT	Internal	IET/IETA	
INV	Inventory/Stock	Based on Posting code	Used to pay for items issued from Advantage Warehouses for an OC transaction type.
JV	Journal Voucher	Based on Posting code	

Process Steps

The following table shows the various steps that the Maximo Other Cost Job goes through and the messages issued at each step:

Process Steps	Messages
1. Selection of Records	<ul style="list-style-type: none"> • Run Started • Date Run: <i>date</i> • If selection of records from ACTGINT table returns 0 records then the following message is issued: "No records found in ACTGINT Table" • Source Journal table read: JRNL_ACTG • If the selection of records from the Journal returns 0 records, then the following message is issued: "No records fetched for Ledger Table". • Number of Records read: 'n'
2. Update WOINT and WOINT_DT tables	<ul style="list-style-type: none"> • Number of Records written to Work Order Integration Table: 'n' • At the end, the following message is issued: Job Status completed

Major Input

Data from the following tables:

- Accounting Integration Parameters (ACTGINT)
- Accounting journal (JRNL_ACTG)
- Work Order Integration Detail (WOINT_DT)
- Task Order (R_TASK_ORD)

Batch Parameters

None

Major Output

- Work Order Integration Detail (WOINT_DT)
- Work Order Integration Summary (WOINT)

Job Return Code

The following table shows the potential job return codes for the job:

Return Code	Condition
Successful (1)	WOINT and WOINT_DT tables updated successfully
Warning (4)	N/A

Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following condition <ul style="list-style-type: none"> • Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

None

Selection Criteria

- Selects the Last Run Date (LAST_RUN_DT) from Work Order Integration Detail (WOINT_DT) for the last record on the table.
- The batch job will perform the following based on the entries on the Accounting Integration Parameters table (ACTGINT).
- Reads those Journal tables that are defined as the Source table in the Ledger Table field (LDGR_TBL) where the Integration Type = "Service".
- For a Journal (Source) table, only those accounting records are selected where there is a match for Transaction Code and Posting Code as defined for that source table and Run time Date Stamp is greater than the Last Run Date selected in the above step.
- If any of the records meet the above selection criteria, the batch job determines the following:
 - Transaction has a phase = Final
 - Using the Task Order on the accounting line of the selected new, modification, or cancellation record, the batch job retrieves the Source System and an External Site ID from the Task Order (R_TASK_ORD) table. If the Source System = "Maximo" and the External Site ID is populated, then the batch job writes the record to the Work Order Integration Detail (WOINT_DT) table; or
 - If the Transaction Function = Modification and Task Order does not have Source System = "Maximo", use the Transaction Code, Transaction Department, and Transaction ID to determine if the entry was previously transmitted to Maximo by reading the Work Order Integration Summary (WOINT) table. If an entry is found, then the batch job writes the record to the Work Order Integration Detail (WOINT_DT) table.

Problem Resolution

The following table shows the possible return codes and recommendations for each processing step.

Step 1: Selection of records

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Records are successfully selected from the Journal	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non Fatal Error (8)	N/A	N/A	N/A
Failed (12)	Job failed due to Fatal conditions	In this step, the job can fail under the following condition: Encounters runtime exceptions If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and reschedule a new job.	Schedule a new job
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. A new job can be scheduled	Schedule a new job
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. A new job can be scheduled.	Schedule a new job

Step 2: Updating the records

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Records are successfully updated to WOINT and WOINT_DT tables	N/A	N/A
Warning (4)	N/A	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Non Fatal Error (8)	N/A	N/A	N/A
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	<p>In this step, the job can fail under the following condition:</p> <p>If committing of this transaction fails.</p> <p>Failure reason needs to be investigated before scheduling a new job.</p>	Failed because of runtime exceptions for an unexpected situation.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. A new job can be scheduled	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. A new job can be scheduled.	N/A

2.1.18 Overhead Rate Process

Description

The purpose of the Overhead Rate process is to generate transactions that will record indirect costs for Cost Accounting entities. Those indirect costs may subsequently be eligible for reimbursement by various funding agencies. The Overhead Rate Method is one of two methods that can be used to record indirect costs. The other method is Cost Allocation. This run sheet only describes the Overhead Rate Method.

The Overhead Method applies user-defined overhead rate percentages to direct costs that have already been attributed to Cost Accounting entities (Major Programs). All accounting activity related to Cost Accounting entities is stored in the Cost Accounting Journal (JRNL_CA). This Journal in conjunction with the Journal Log (JRNL_LOG) is the basis for records that are to be selected for overhead rate application. Not all direct costs on the Cost Accounting Journal are eligible for selection by the Overhead Rate Process.

Steps in Running this Process

The Overhead Rate Process performs four key functions that must also take place in the following sequence:

1. Selection of Records from Cost Accounting Journal
2. Calculation of Overhead Amount for selected records
3. Generation of Charge Transactions to record the Overhead amounts calculated in the previous step.
4. Reversal of Charge Transactions previously generated

Selection of Records: Records will be selected from the Cost Accounting journal based on the last time the Overhead Rate Process was last successfully run to completion rather than based on the Fiscal Year From/To, Accounting Period From/To, and Budget Fiscal Year fields entered on the Overhead Rate Parameters table, thus removing any time constraints from the process. The Overhead Rate Process will then be capable of being run as frequently as needed. The Posting Code table defines whether or not Cost Accounting Journal records are eligible for Overhead Rate Process selection. If a journal record's posting code is eligible, then it may be submitted for further eligibility tests before it can be finally selected for overhead rate application. Typically, the posting codes associated with cash expenditures will be defined as eligible for overhead. Finally, the subset of journal records is narrowed down further based on the Major Program. If the Major Program entity defined on the journal record has been recorded as eligible for overhead, then that journal record can be included in the final subset of entries that will have an overhead rate applied to record indirect costs.

Calculation of Overhead Amount: This step takes the final set of selected journal records from the previous step and applies a user-defined overhead rate. A default rate is defined by the administering government entity for each Major Program. This rate is applied unless exception rates have been defined on the Overhead rate Exception table for specified Chart of Account combinations. The output is a set of accounting lines that are ready to be recorded on Charge Transactions. This step may also redirect the original COA values of a selected record to some new values, based on values defined on the Major Program. One further function that may also be performed here and that is dependent on the custom parameter record is the ability of the process to combine selected records with identical combinations of Chart of Account values.

Generation of Charge Transactions: This step creates Charge Transaction accounting lines in XML format from the accounting lines generated in the previous step.

Transaction breaks are generated in the following manner:

First it checks for CH_DOC_ACTG line limit on the Transaction Component Requirements table:

- If it contains some value:
- It checks on SOPT setting for No. of Accounting lines field:
 - If this field is empty then Max no. of accounting lines considered for CH transaction are equal to CH_DOC_ACTG line limit on the Transaction Component Requirements table.
 - If this field is not empty then Max no. of accounting lines considered for CH transaction are equal to minimum of SOPT values and CH_DOC_ACTG line limit on the Transaction Component Requirements table.
- If CH_DOC_ACTG line limit on the Transaction Component Requirements table is empty then only Max no. of accounting lines considered for CH transaction are equal to SOPT value.

The generated XML file can then be loaded to the transaction catalog. From there, the Charge Transactions can then be submitted. Both of the latter functions, the loading and submission of the Charge Transactions can be accomplished by chaining the Overhead Rate Process to the System Maintenance Utility process. The System Maintenance Utility will be run twice in the chain. The first run will load the Charge Transactions to the Transaction Catalog, the second run will attempt to submit those previously loaded transactions.

Reversal of Prior Transactions: This step should only be executed if the transactions generated in the previous step are deemed to be invalid. This step is accomplished in a separate running of the process. As a modification to the Overhead rate process, only records processed in the prior run can be selected for reversal in the next run. In other words, the Last Run Flag of the parameter record must be set for the records to be selected for reversal. Transactions that were successfully submitted will be cancelled, and any generated transactions that failed to submit in the previous step will be discarded from the transaction catalog. The recording of the range of generated transactions from the previous step in the Journal Log identifies the range of transactions that is to be reversed in this step.

Run Type and Run Mode

The Overhead Rate Process employs a Custom Parameter table as input to the program. There are two key fields, Run Type and Run Mode, on each parameter record that will determine which combination of the four functions described above will be performed by the program. It is the combination of Run Type and Run Mode that determines the combination of functions to be performed.

Run Type = Initial, Run Mode = Report Only:

This combination results in the execution of the first two functions, record selection and calculation of the overhead amount for each selected record. No Charge Transaction lines are generated, but a view of the Charge Transaction accounting lines that would have been generated if another run mode had been chosen is available in a Report Journal.

Run Type = Initial, Run Mode = Generate Transactions Only:

This combination can only be executed after the previous combination has been executed for the same custom parameter record. This combination will only perform step 3: the generation of Charge Transaction accounting lines. The input for this step is the Report Journal output from the

first combination of Run Type and Run Mode above. To load and submit the generated accounting lines, the Overhead Rate Process needs to be chained to the System Maintenance Utility process.

Run Type = Initial, Run Mode = Report and Generate Transactions:

This combination is a hybrid of the first two: it performs the first three steps described in the functional flow above. Linking the process to the System Maintenance Utility again provides the opportunity to load and submit the generated Charge Transaction accounting lines.

Run Type = Reversal, Run Mode = Report and Generate Transactions:

This combination performs the fourth function described above, that is, the reversal of previously generated transactions. Draft Charge transactions will be discarded and Cancellation transaction versions will be submitted for Charge transactions in Final state.

When to Run

Since the Overhead Rte Process is independent of the accounting period, the process can be run as often as required. The process will select records based on the last entry in the Journal Log and accordingly search for new unprocessed records from the Cost Accounting journal. If a reversal is required for any reason, then the Overhead Process can be run in Reversal mode using the Parameter record with the Last Run flag checked in the Process History section.

Run Sequence

There is only one job specifically designed for Overhead, and that is the Overhead Rate Process job itself. So that the final intentions of the process are accomplished, the process should consist of a chain involving the System Maintenance Utility so that Charge Transactions transactions to record indirect costs can be loaded and submitted on the Transaction Catalog. But the System Maintenance Utility will not always be part of the “entire” Overhead Process. Again, the jobs to be run depend upon the combination of Run Type and Run Mode:

Run Mode	Run Type	Job Name (In Sequential Order)
Report Only	Initial	Overhead Rate Process
Generate Transactions Only	Initial	Overhead Rate Process System Maintenance Utility (for loading of transactions) System Maintenance Utility (for submitting of transactions)
Report and Generate Transactions	Initial	Overhead Rate Process System Maintenance Utility (for loading of transactions) System Maintenance Utility (for submitting of transactions)
Report and Generate Transactions	Reversal	Overhead Rate Process NOTE: Although Reversals result in the loading and submitting cancellation transactions, the Overhead Rate Process does not need to be chained to the

Run Mode	Run Type	Job Name (In Sequential Order)
		System Maintenance Utility for this purpose. Instead, the Overhead Rate Process calls a similar routine employed by the System Maintenance Utility job that accomplishes this goal without running the System Maintenance Utility itself.

Parameters

Batch Parameters

Description (Caption)	Parameter Name	Default Value
OVD_RT_PAR_ID	Selected Parameter Id	0

Custom Parameters

If you are running the job in the Financial application, you can click on the **Custom Parameter** link and select a parameter record. If you are running the job in the Administration application, enter a valid Parameter ID from the Custom Parameter table from the Financial application.

Data Object Name R_OVD_RT_PAR

Description (Caption)	Parameter (Attribute) Name	Data Type/Length	Default Value
Input			
Exporting Location at Overhead Job	AMSEXPORT (** Refer to Note: Assumptions for SWBP on page no. 5)	Text	-
Parameter Location at Overhead Job	AMSPARM (** Refer to Note: Assumptions for SWBP on page no. 5)	Text	-
Overhead Rate Parameter Id	OVD_RT_PAR_ID	Number	-
Run Mode	RUN_MOD	Number	-
Next Run Type	NXT_RUN_TYP	Number	-
Retain BFY-FY-APD	Retain_BFY_FY_AP D	Number	1
Summarize	SUM	Number	-
Output			
Transaction Record Date	DOC_REC_DT	Date/Time	-
Fiscal Year	FY	Number	-
Accounting Period	PER	Number	-

Description (Caption)	Parameter (Attribute) Name	Data Type/Length	Default Value
Budget Fiscal Year	BFY	Number	-
Description	DSCR	Text	-
Prefix	PFX	Text	-
Transaction Code	DOC_CD	Text	-
Transaction Category	DOC_CAT	Text	CA
Transaction Type	DOC_TYP	Text	CH
Transaction Department Code	DOC_DEPT_CD	Text	-
Event Type	EVNT_TYP_ID	Text	-
Transaction unit Code	DOC_UNIT_CD	Text	-
History			
Previous Run Date	PREV_RUN_DT	Date/Time	-
Previous Run Type	PREV_RUN_TYP	Number	-
Previous Run Mode	PREV_RUN_MOD	Number	-
Initial Run Date	INIT_RUN_DT	Date/Time	-
Initial Unique Identifier	Unid	Number	-
Reversal Unique Identifier	Rev_Unid	Number	-
Beginning Record	START_REC_NO	Number	-
Ending Record	END_REC_NO	Number	-
Last Run	LST_RUN_FL	Number	0

Load Charge

Description (Caption)	Parameter (Attribute) Name	Data Type/Length	Default Value
Input			
Filename to be loaded:	FILE_NM	Text	\$\$\$AMSEXPOR \$\$/ChargeDocu ment.xml

Submit Charge

Description (Caption)	Parameter (Attribute) Name	Data Type/Length	Default Value
Input			
Exception Report File	EXCEP_REP_FILE	Text	\$\$\$AMSLOGS\$\$/

Name:	_NM		OvhdExcepRep.txt
Parameter file name	PARM_FILE	Text	\$\$AMSPARM\$\$/ OVCH_PARM.txt

Major Input

Run Mode	Run Type	Input
Report Only	Initial	Overhead Rate Parameter table (R_OVD_RT_PAR) Posting Code table (R_PSCD) Major Program table (R_MJR_PROG) Overhead Rate Exception table (R_OVD_RT_EXCP) Cost Accounting Journal (JRNL_CA) Journal Log (JRNL_LOG)
Generate Transactions Only	Initial	Overhead Rate Parameter table (R_OVD_RT_PAR) Report Journal (R_RPT_JRNL_OP)
Report and Generate Transactions	Initial	Overhead Rate Parameter table (R_OVD_RT_PAR) Posting Code table (R_PSCD) Major Program table (R_MJR_PROG) Overhead Rate Exception table (R_OVD_RT_EXCP) Cost Accounting Journal (JRNL_CA) Journal Log (JRNL_LOG) Report Journal (R_RPT_JRNL_OP)
Report and Generate Transactions	Reversal	Overhead Rate Parameter table (R_OVD_RT_PAR) Cost Accounting Journal (JRNL_CA) Journal Log (JRNL_LOG)

Selection Criteria

Run Mode	Run Type	Inputs
Report Only	Initial	Select all Cost Accounting Journal (JRNL_CA) records which fall in the range of the last processed record in the prior run and all new transactions posted in the Cost Accounting journal (JRNL_CA) since then. Select only those records whose posting codes (R_PSCD) are eligible for overhead. Select only those records whose Major Program (R_MJR_PROG) is eligible for Overhead.
Generate Transactions Only	Initial	Select all records from Report Journal (R_RPT_JRNL_OP)

Run Mode	Run Type	Inputs
Report and Generate Transactions	Initial	Select all Cost Accounting Journal (JRNL_CA) records which fall in the range of the last processed record in the prior run and all new transactions posted in the Cost Accounting journal (JRNL_CA) since then. Select only those records whose posting codes (R_PSCD) are eligible for overhead. Select only those records whose Major Program (R_MJR_PROG) is eligible for Overhead.
Report and Generate Transactions	Reversal	Select all transactions in the range identified in the Journal Log (JRNL_LOG) for when the process was previously run to generate overhead transactions.

Outputs

Process Output is dependent on the combination of Run Mode and Run Type.

Run Mode	Run Type	Outputs
Report Only	Initial	*Report Journal (R_RPT_JRNL_OP) Journal Log (JRNL_LOG) Overhead Parameter Rate table (R_OVD_RT_PAR)
Generate Transactions Only	Initial	Journal Log (JRNL_LOG) Overhead Parameter Rate table (R_OVD_RT_PAR) Charge Transaction.XML (The Overhead Process when chained to subsequent runnings of the System Maintenance Utility will result in the submission of Charge Transactions on the Transaction Catalog.)
Report and Generate Transactions	Initial	*Report Journal (R_RPT_JRNL_OP) Journal Log (JRNL_LOG) Overhead Parameter Rate table (R_OVD_RT_PAR) Charge Transaction.XML file (The Overhead Process when chained to subsequent runnings of the System Maintenance Utility will result in the submission of Charge Transactions on the Transaction Catalog.)
Report and Generate Transactions	Reversal	Journal Log (JRNL_LOG) Overhead Parameter Rate table (R_OVD_RT_PAR) Discarded/Cancelled Charge Transaction Lines

*Note: The R_RPT_JRNL_OP table is first truncated and then updated with the Overhead records generated by the process, based on the job parameters.

NOTE:

Run Mode = Report Only. May only be used with Run Type = Initial:

Report Journal is updated with the accounting lines that would have been generated in XML format if the process was run in one of the other two modes

Journal Log: A record is created on the Journal Log to transaction running of the process

Overhead rate parameter table: The Parameter record used as input is updated with important history detail that is important for future runs of the process using the same parameter.

Run Mode = Generate Transactions Only. May only ever be used with Run Type = Initial

XML for charge transactions file is created in the default Import/Export file location directory

Journal Log: A record is created on the Journal Log to transaction running of the process

Run Mode = Report and Generate Transactions, Run Type = Initial

Report Journal is updated with the accounting lines that are then generated in XML format

Journal Log: A record is created on the Journal Log to transaction running of the process

Overhead Rate Parameter Table: The Parameter record used as input is updated with important history detail that is important for future runs of the process using the same parameter.

XML for charge transactions file is created in the default Import/Export file location

Run Mode = Report and Generate Transactions, Run Type = Reversal

Journal Log: A record is created on the Journal Log to transaction running of the process.

Overhead Rate Parameter table: The Parameter record used as input is updated with important history detail.

Discarded/Cancelled Charge Transaction Lines: The Charge Transactions generated when the process was run initial run type are either discarded or cancellation draft created, depending on the Transaction Phase on Transaction Catalog.

Sort Criteria

Records selected from the Cost Accounting Journal are sorted in the following order (this is only done when run type = initial and run mode = Report Only or Report and Generate Transactions):

- Dept
- Major Program
- Program
- Program Period
- Activity
- Object

If either Activity or Object is Null, sorting is by the most detailed records first, that is, those that are not null. This is to facilitate the application of the correct overhead exception rate.

Problem Resolution

If the process fails due to any reason then perform the corrective actions:

- Check the Log file for any errors that may have occurred while the program was running.

- Check whether an XML file of the name XML for charge transactions is created in the D:\AMSADV30\RTFiles\ directory for today's date and time. This will only be done if the process is run in a mode that is intended to generate transactions.
- Delete the XML file if it exists.
- Re-run process starting from first job step.
- This batch program produces new Advantage transactions, which are subject to the line limit functionality constraints. Sites should ensure that they run this job with parameters set to ensure that the created transactions are within the line limit controls.
- For performance reasons, the Identify and Archive Stale Gaps job should be run periodically to remove gaps in journal records being tracked and processed by this program. Please refer to the "Identify and Archive Stale Gaps" run sheet in the *CGI Advantage Financial – General Accounting Run Sheets* for more details on scheduling and recommended parameters.

2.1.19 Program Asset Generation

Chain Job Name	Program Asset Generation
Recommended Frequency	Please see the When to Run section that follows
Can be Restarted?	There are notes on each job in the chain about restarting if a step fails.
Reports Generated	Please see individual job steps

Description

Program Asset Generation is the process to automate the management and recognition of capital balances with regard to construction in progress. The balance often used is construction in progress (CIP), but the process can also be used to capture remediation costs before disposing of an asset into a new asset component.

The process will generate Fixed Asset transactions that increase, change the Fixed Asset Type, and cancel Fixed Asset Components. The chain consists of several job steps:

- Update FACPER (Construction Expenditure Reconciliation)
- Update FACPEA (Construction Expenditure Accumulation)
- Generate FI (Fixed Asset Increase/Decrease transactions)
- Generate FX (Fixed Asset Type Change transactions)
- Generate FC (Fixed Asset Cancellation transactions)

Please note that the chart of accounts (COA) used to track and account for construction and acquisition efforts are passed through this process to the accounting records found on the Fixed Asset Registry (FARACTG) page. While that is often needed for very important COA, there are others that are not critical to the asset record. There is a parameter in the Create FI Transactions job step that can be used to reduce the variety of COA elements retained in the process. Not all COA can be excluded, but it is recommended that those not necessary for historical asset reporting be eliminated to reduce the number of Accounting records. That COA detail will always remain as a function of recording the original expenditures.

Job Step 1: Update FACPER:

1. Process gap records in the Cost Accounting Journal from previous run.
2. Process Cost Accounting Journal records from the last journal record processed.
3. Inserts records into the Construction Expenditure Reconciliation (FACPER) and Construction Expenditure Reconciliation Details (FACPERD).

Job Step 2: Update FACPER:

1. Reviews records from FACPER with a blank Accumulation date.
2. Insert records into the Construction Expenditure Accumulation (FACPEA):

Job Step 3: Generate FI Transactions:

1. Generate FI transactions for records from FACPEA with Pending Increase Amount not equal to \$0.00.
2. Upload the FI xml file.
3. Submit the FI transactions.

Job Step 3: Generate FX Transactions:

1. Generate FX transactions for records from FACPAD with a Final In-Service Date less than or equal to Current Run Date and Final indication as *false*.
2. Uploading the FX xml file
3. Submit the FX transactions.

Job Step 3: Generate FC Transactions:

1. Generate FC transactions for records from the FACPAD with the Cancel indication as *true*. set to true and Cancelled Flag set to false.
2. Upload the FC xml file.
3. Submit the FC transactions.

When to Run

Although it can be run on demand, this process should be run on a regular frequency to not only keep volume manageable but also so that Construction in Progress balances are as currently as possible. The trade off for a higher frequency is a higher volume of transactions. The frequency of this process should be more frequent than the archiving of gaps from the Cost Accounting Journal.

Input

Update FACPER

- Cost Accounting Journal (JCA / JRNL_CA)
- Journal Log (JLOG / JRNL_LOG)
- Construction Allocation (FACPA / R_FACPA)
- Construction Allocation Details (FACPAD / R_FACPAD)
- Posting Code (PSCD / R_PSCD)
- Object (OBJ / R_OBJ)
- Activity (ACTV / R_ACTV)

Update FACPEA

- Construction Expenditure Reconciliation FACPER / (R_FACPERT)
- Construction Expenditure Reconciliation Details (FACPERD / R_FACPERD)

Generate FI Transactions

- Construction Expenditure Accumulation (FACPEA / R_FACPEA)

- Automated Transaction Number (ADNT / AUTO_DOC_NO)

Generate FX Transactions

- Construction Allocation Details (FACPAD / R_FACPAD)
- Fixed Asset Registry Header (FARHDR / R_FAR_HDR)
- Automated Transaction Number (ADNT / AUTO_DOC_NO)

Generate FC Transactions

- Construction Allocation Details (FACPAD / R_FACPAD)
- Automated Transaction Number (ADNT / AUTO_DOC_NO)

Output

Update FACPER

- Construction Allocation (FACPA / R_FACPA)
- Construction Expenditure Reconciliation FACPER / (R_FACPERT)
- Construction Expenditure Reconciliation Details (FACPERD / R_FACPERD)
- Journal Log (JLOG / JRNL_LOG) table

Update FACPEA

- Construction Expenditure Reconciliation FACPER / (R_FACPERT)
- Construction Expenditure Accumulation (FACPEA / R_FACPEA)

Generate FI Transactions

- Fixed Asset Increase/Decrease (FI) transactions
- Construction Expenditure Reconciliation FACPER / (R_FACPERT)

Generate FX Transactions

- Fixed Asset Type Change (FX) transactions
- Construction Allocation Details (FACPAD / R_FACPAD)

Generate FC process

- Fixed Asset Cancellation (FC) transactions
- Construction Allocation Details (FACPAD / R_FACPAD)

Parameters

The Program Asset Generation process is comprised of 11 batch jobs that are chained together. Each step in the process is described below.

Job	Parameter	Description	Default Value
Update FACPER	Global Bypass Review (BYPASS_REVIEW_FL)	A required input parameter to indicate only the selection of FACPEA records marked as reviewed. A value of Y (yes) will result in all FACPEA records being selected regardless if they have been reviewed or not.	N
	Run Date (RUN_DATE)	A required selection date used determine the Fiscal Year when selecting eligible Object codes and Activity codes. Enter as mm/dd/yyyy. The current date is used if no value is entered.	No Default
	Prefetch Count MAX_PREFETCH_COUNT	A required performance parameter that defines the maximum number of records that will be selected for processing in an instance.	1000
	Commit Block Size (COMMIT_BLOCK)	A required performance parameter that defines how many records are committed by the process in an instance. 100 defaults if left blank.	100
	Journal Log Prefetch Count (JLOG_PREFETCH_COUNT)	A required performance parameter that defines the number of JLOG records to be fetched and processed for Gap records, 100 defaults if left blank.	500
Update FACPEA	Run Date (RUN_DATE)	A required input parameter of a date to use as the Accumulation Date on FACPER. Enter as mm/dd/yyyy. The current date is used if no value is entered.	No Default
Generate FI Transaction	Transaction Prefix (DOC_PREFIX)	A required output parameter of the prefix used for transaction numbering.	No Default

Job	Parameter	Description	Default Value
	Transaction Code (DOC_CD)	A required output parameter of the transaction code to be created to increase the CIP value of an asset.	FI
	Department (DOC_DEPT_CD)	A required output parameter used for security and automatic numbering of the transactions created to increase the CIP value of an asset.	No Default
	Unit (DOC_UNIT_CD)	An optional output parameter used for security purposes of the transactions created to increase the CIP value of an asset.	No Default
	Transaction Status (DOC_STA_CD)	A required output parameter of the transaction status used when transactions are loaded. A value of 2 (Ready) will result in the subsequent job step submitting the transactions. This is the recommended setting. A value of 1 (Held) will result in the transactions being loaded but not submitted. It would require manual intervention to then submit the transactions.	No Default
	Maximum Number of Commodity Lines (MAX_COMP_NO)	A required output parameter that defines the maximum the number of component lines in a generated transaction before creating a subsequent transaction. Ensure this is in line with any Transaction Component Requirements (DCREQ) record for Maximum Number of Lines.	50

Job	Parameter	Description	Default Value
	Maximum Number of Accounting Lines (MAX_ACCT_NO)	A required output parameter that defines the maximum the number of accounting lines in a generated transaction before creating a subsequent transaction. Ensure this is in line with any Transaction Component Requirements (DCREQ) record for Maximum Number of Lines.	50
	Exporting Location (AMSEXPORT)	The required location to which the XML file will be written.	\$\$AMSR00T\$ \$/ExportImport
	Parameter Location (AMSPARM)	The required location to which the TXT file of parameters passed to subsequent jobs steps will be written.	\$\$AMSR00T\$ \$/Parms
	Exclude COA List (EXCL_COA_LST)	An optional parameter for a listing of one or more COA codes, by data attribute name such as TASK_CD, to be excluded when grouping records to create FI accounting lines. This parameter is used to control the volume of accounting lines from containing what was necessary for construction tracking but not necessary to retain with the asset or asset component. Note: FUND_CD, OBJ_CD, DEPT_CD, UNIT_CD, APPR_CD, ACTV_CD, PROG_CD, MJR_PROG_CD, PHASE_CD, and PPC_CD cannot be specified.	No Default
Load FI	Parameter File (PARM_FILE)	The required location and file name created in the previous job step for loading FI transactions.	\$\$AMSPARM\$ \$/FILoadParm.txt
Submit FI	Parameter File (PARM_FILE)	The required location and file name created in the previous job step for submitting FI transactions.	\$\$AMSPARM\$ \$/FISubmitParm.txt

Job	Parameter	Description	Default Value
Generate FX Transactions	Transaction Prefix (DOC_PREFIX)	A required output parameter of the prefix used for transaction numbering.	No Default
	Transaction Code (DOC_CD)	A required output parameter of the transaction code to be created to change the fixed asset type of an asset.	FX
	Department (DOC_DEPT_CD)	A required output parameter used for security and automatic numbering of the transactions created to change the fixed asset type of an asset.	No Default
	Unit (DOC_UNIT_CD)	An optional output parameter used for security purposes of the transactions created to change the fixed asset type of an asset.	No Default
	Transaction Status (DOC_STA_CD)	A required output parameter of the transaction status used when transactions are loaded. A value of 2 (Ready) will result in the subsequent job step submitting the transactions. This is the recommended setting. A value of 1 (Held) will result in the transactions being loaded but not submitted. It would require manual intervention to then submit the transactions.	No Default
	Maximum Number of Commodity Lines (MAX_COMP_NO)	A required output parameter that defines the maximum the number of component lines in a generated transaction before creating a subsequent transaction. Ensure this is in line with any Transaction Component Requirements (DCREQ) record for Maximum Number of Lines.	50

Job	Parameter	Description	Default Value
	Exporting Location (AMSEXPORT)	The required location to which the XML file will be written.	\$\$AMROOT\$ \$/ExportImport
	Parameter Location (AMSPARM)	The required location to which the TXT file of parameters passed to subsequent jobs steps will be written.	\$\$AMROOT\$ \$/Parms
Load FX	Parameter File (PARM_FILE)	The required location and file name created in the previous job step for loading FX transactions.	\$\$AMSPARM\$ \$/FXLoadParm. txt
Submit FX	Parameter File (PARM_FILE)	The required location and file name created in the previous job step for submitting FX transactions.	\$\$AMSPARM\$ \$/FXSubmitPar m.txt
Generate FC Transaction	Transaction Prefix (DOC_PREFIX)	A required output parameter of the prefix used for transaction numbering.	No Default
	Transaction Code (DOC_CD)	A required output parameter of the transaction code to be created to cancel an asset.	FC
	Department (DOC_DEPT_CD)	A required output parameter used for security and automatic numbering of the transactions created to cancel an asset.	No Default
	Unit (DOC_UNIT_CD)	An optional output parameter used for security purposes of the transactions created to cancel an asset.	No Default

Job	Parameter	Description	Default Value
	Transaction Status (DOC_STA_CD)	A required output parameter of the transaction status used when transactions are loaded. A value of 2 (Ready) will result in the subsequent job step submitting the transactions. This is the recommended setting. A value of 1 (Held) will result in the transactions being loaded but not submitted. It would require manual intervention to then submit the transactions.	No Default
	Maximum Number of Commodity Lines (MAX_COMP_NO)	A required output parameter that defines the maximum the number of component lines in a generated transaction before creating a subsequent transaction. Ensure this is in line with any Transaction Component Requirements (DCREQ) record for Maximum Number of Lines.	50
	Exporting Location (AMSEXPOR)	The required location to which the XML file will be written.	-
	Parameter Location (AMSPARM)	The required location to which the TXT file of parameters passed to subsequent jobs steps will be written.	-
Load FC	Parameter File (PARAM_FILE)	The required location and file name created in the previous job step for loading FC transactions.	\$\$AMSPARM\$ \$/FCLoadParm.txt
Submit FC	Parameter File (PARAM_FILE)	The required location and file name created in the previous job step for submitting FC transactions.	\$\$AMSPARM\$ \$/FCSubmitParm.txt

Sort Sequence

None

Selection Criteria

Update FACPAR

- Selects records from Journal Log with Status of GAP FOUND and Process ID of UPDFACPER. The Cost Accounting Journal records in those gaps are selected, if any, and are analyzed for selection. A record is selected for processing in step 3 if it meets all of these selection criteria:
 - Matches the COA of an active Construction Allocation (FACPA) record
 - Posting Code is marked *true* for the Fixed Asset Construction Program Eligible
 - The Object and Activity are blank or have the Construction Program Eligible indication as *true*.
- Select Cost Accounting Journal records from the last record processed using the same 3 selection criteria after obtaining the last record processed from Journal Log.
- For each journal record selected, one record is inserted into FACPER. The process selects all child records from Construction Allocation Detail (FACPAD) for each Active allocation record and inserts records into FACPERD with Line Amounts set as per the allocation percentage on the Construction Allocation Detail record. The Reviewed indication on the FACPER record should be set as per the Global Bypass Review parameter value and the Bypass Review indication on the Construction Allocation Total record. The Allocation Total record is then updated with the last Journal record processed.

Update FACPAD

- Selects Reviewed FACPER records where Accumulate date is blank. The job updates FACPERD with an Accumulated date. This prevents users from modifying a record on the FACPERD once it has been accumulated on the FACPEA. The next time the job runs, it does not select records that have been previously accumulated on FACPEA.
- For each of the selected FACPER, the corresponding child records from FACPERD will be selected. If a FACPEA record already exists with the Fixed Asset Number, Commodity Number, Budget Fiscal Year, Fiscal Year and all Chart of Account elements, then the process will add the incremental amount to the Pending Increase amount. If no matching record exists, a new record will be inserted with the same combination and will have a Pending Increase amount set to the incremental amount.

Generate FI process

- Pending Increase Amount not equal to zero
- For records selected update the FACPEA to add the Pending Increase Amount to the Cumulative Increase Amount and set Pending Increase Amount to \$0.00.

Generate FX process

- Final In Service Date is less than or equal to run date
- Final is *no*
- Cancelled is *no*
- For records selected to create FX transactions, the process sets that Final indication to *true*.

Generate FC process

- Cancel is *yes*
- Cancelled is *no*
- For records selected to create FC transactions, the process updates the Cancelled indication to *true*.

Problem Resolution

The whole process is implemented with checkpoints. If the process fails for any reason (such as the network is down or the server is down), then the process restarts from the last point where it stopped.

For performance reasons, the Identify and Archive Stale Gaps job should be run periodically to remove gaps in journal records being tracked and processed by this program. Please refer to the “Identify and Archive Stale Gaps” run sheet in the *CGI Advantage Financial – General Accounting Run Sheets* guide for more details on scheduling and recommended parameters.

2.1.20 Reclassification Process

Description

The Cost Accounting reclassification functionality is divided internally into the following processes:

1. Normal Reclassification
2. Automatic Overflow Recapture

Normal Reclassification - During the life of a Cost Accounting entity (for example, project, grant, or job), the financial structure of the reimbursement funding and cost eligibility will require modification. These modifications are due to a variety of factors to include: changes in the agreement amounts with an external funding source, changes in the eligibility of certain types of reimbursable expenditures, or setup errors of a cost accounting program.

Automatic Overflow Recapture - Changes to funding or eligibility may free up monies, which can be applied against transactions that were previously not eligible for reimbursement due to exceeding the funding limitations. These amounts would have been applied to the “overflow” priority in a Funding Profile and, with the new funding rules established, may now be eligible for reimbursement.

It is important to note these two processes are independent of each other and must be executed separately. In addition, when the reclassification process is executed in the normal reclassification mode or automatic overflow recapture mode, the reclassification process identifies the original transactions required to be reclassified and creates the appropriate adjusting transactions to affect the changes. The reclassification process can be done against cash expenditures, charges, or internal transactions.

Run Modes

The majority of the cost accounting offline processes can be submitted in three different “run modes” to provide the user community and the system administrators additional flexibility when requesting complex and intricate offline processes. The following are the descriptions of the run modes available in CGI Advantage Financial:

Report Only - The purpose of this “run mode” is to provide the capability for offline processes to be submitted without the final/true database updates occurring. The Report-Only mode provides a “what-if” analysis tool to the users so they can see what potential data impacts will eventually occur by an offline process without having the permanent portions of the database updated. For example, if the main output of a cost accounting offline process is to generate and submit a significant number of new transactions, such as, Charge Transactions. The Report-Only run mode can be utilized to produce a detailed report (and update temporary tables) containing the intended output (for example, Charge Transaction data) without actually creating the massive number of Charge Transactions. In this example, the users would be able to review the “what-if” reports and/or temporary tables to ensure the Charge Transactions contain the expected results prior to actually creating and submitting the transactions. The Report-Only run mode provides the users and system administrators the capability to make the necessary adjustments to the Advantage business rules found in the cost accounting set-up tables prior to the actual database updates being applied by the offline process results. In other words, if the “what-if” report analysis determines that there are changes required to the Advantage set-up data then the users would be able to make the corrections and then re-run the offline process in Report-Only run

mode again to check the set-up data corrections before updating the database. The Report-Only run mode can be run as many times as necessary until the users determine that the “what-if” results are as expected.

Update - The purpose of this “run mode” is to perform the updates to the database with the results of the previous run in Report-Only “run mode”. In other words, if the previous run of a particular offline process was in the Report-Only “run mode” and after analyzing the expected results the users decided that the results were correct then this run mode will perform all necessary database updates. It is important to note that the Update run mode does not re-process the data, its sole responsibility is to take the data processed during the previous Report-Only run mode (that is, residing in the temporary tables) and then perform the appropriate database updates to the “permanent” (that is, not temporary) tables. Consequently, the Update run mode cannot be run unless the previous run mode was Report-Only because the Update run mode does not re-process and/or re-calculate the data.

Report & Update - The purpose of this “run mode” is to combine the first two “run modes” (that is, Report-Only and Update) into one job submission step. It allows the users the capability to update the database without having to review the “what-if” reports. For example, if the users are certain the cost accounting set-up data is correct, then they may want to utilize this all-in-one “run mode” to save time from having to perform the two-step submission process (that is, first the Report-Only and then the Update).

Technical Flow – Normal Reclassification

This process is executed when there are changes in the agreement amounts with an external funding source, changes in the eligibility of certain types of reimbursable expenditures, or setup errors. Transactions selected for normal reclassification are identified by the user community through custom parameters entered into Reclassification Parameter Screen.

At the time record has been selected by the normal reclassification process the following steps are executed:

- a. Selected Records are written to the reclass report temp table (RCLS_RPT_TEMP)
- b. Following Reports are created from the reclass report temp table:
 - **Parameters Report** that lists the parameters for the current run.
 - **General Report** that lists all the records to be processed in the current run.
 - **Exception Report** that lists all the records that have been modified or cancelled beyond the date specified, if any, on the parameters screen. It also lists all the records for which funding Profile is inactive. Journal Vouchers that met selection criteria for a reclassification, which could not be modified because the transaction type prohibits that action, are listed for manual reclassification on a second journal voucher.
- c. A modification transaction is created in XML file format.
- d. All lines that have been selected be reclassified will have the “Reclass Indicator Flag” (RECLASS_IND_FL) incremented by one to trigger processing.
- e. The XML files are imported to the database using SysManUtil.
- f. The newly imported modification transactions are submitted using SysManUtil.
- g. If transaction(s) fails to submit, due to any reason, the offline process generates an exception report, which is then used by the “Reclass Cleanup” to remove the transactions that failed during the submission phase. In addition, the “Reclass Cleanup” process will generate a report of all failed transactions with error messages pertaining to their failure.

- In the report only mode steps, a to d are executed.
- In the update mode steps, e to g are executed.
- In the report and update mode, all the steps are executed.

It is important to note that for each transaction, which is reclassified via a “Modification” transaction, the front-end split logic does two things:

- **Back out** – The original reimbursement split information will be backed out using the LIFO method of accounting.
- **Re-process** – The front-end split logic will determine the funds available beginning at the lowest Funding Priority until it distributes the money.

The Funding Split Log (FSL) will include information about the original transaction that was split and the new transaction split after the Automatic Overflow Recapture process.

When to Run – Normal Reclassification

At any time

Run Sequence – Normal Reclassification

1. Reclassification Adjustments
2. System Maintenance Utility (for loading of Charge transactions)
3. System Maintenance Utility (for loading of AD transactions)
4. System Maintenance Utility (for loading of MD transactions)
5. System Maintenance Utility (for loading of IET transactions)
6. System Maintenance Utility (for loading of ITA transactions)
7. System Maintenance Utility (for loading of PYRL transactions)
8. System Maintenance Utility (for loading of CI transactions)
9. System Maintenance Utility (for loading of TR transactions)
10. System Maintenance Utility (for loading of OC transactions)
11. System Maintenance Utility (for loading of SN transactions)
12. System Maintenance Utility (for loading of CR transactions)
13. System Maintenance Utility (for submitting of Charge transactions)
14. System Maintenance Utility (for submitting of AD transactions)
15. System Maintenance Utility (for submitting of MD transactions)
16. System Maintenance Utility (for submitting of IET transactions)
17. System Maintenance Utility (for submitting of ITA transactions)
18. System Maintenance Utility (for submitting of PYRL transactions)
19. System Maintenance Utility (for submitting of CI transactions)
20. System Maintenance Utility (for submitting of TR transactions)

- 21. System Maintenance Utility (for submitting of OC transactions)
- 22. System Maintenance Utility (for submitting of SN transactions)
- 23. System Maintenance Utility (for submitting of CR transactions)
- 24. Reclassification Clean Up Process

Step 1 has to be run before any of the other steps are run.

Step 2 to 12 can be run in any order and any of these steps can be disabled also.

Step 13 to 23 can be run in any order and any of these steps can be disabled also.

Step 24 should be run only after at least one of the steps from 13 to 23 has been run.

Parameters – Normal Reclassification

Batch Parameters

COMMIT_BLOCK:- Specifies the Commit Block size. Valid value is any positive integer number. Process will commit the transaction after the number of the records specified in this parameter is processed. This parameter is defaulted to 100 if it is not specified or an invalid value is entered.

PARAM_ID:- Specifies the Parameter Id that is the link to the Custom Parameters listed below. Valid value is the appropriate positive integer number. This parameter is defaulted to 0 if it is not specified or an invalid value is entered.

Custom Parameters

If you are running the job in the Financial application, you can click on the **Custom Parameter** link and select a parameter record. If you are running the job in the Administration application, enter a valid Parameter ID from the Custom Parameter table from the Financial application.

SWBP Parameters:

Reclassification Process

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
AMSEXPORT	Exporting Location at Reclassification Process Job	Required	-	** Refer to Note: Assumptions for SWBP on page no. 7
AMSPARM	Parameter Location at Reclassification Process Job	Required	-	** Refer to Note: Assumptions for SWBP on page no. 7
EXCLUDE_DO C_CD	List of Transaction Codes to exclude	Optional	-	Enter the Transaction Code separated by a comma to exclude the Selection of the transaction by the Reclassification process.

Reclass - CH Uploading

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/C HRECLAS S.XML	-
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/C HRECLAS S.XML	-

Reclass - AD Uploading

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/A DRECLAS S.XML	-
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/A DRECLAS S.XML	-

Reclass - MD Uploading

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/M DRECLAS S.XML	-
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/M DRECLAS S.XML	-

Reclass - IET Uploading

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/IE TRECLAS	-

			S.XML	
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/IE TRECLAS S.XML	-

ITA Uploading

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/IT ARECLAS S.XML	-
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/IT ARECLAS S.XML	-

Reclass – PYRL Uploading

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/P YRLRECL ASS.XML	-
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/P YRLRECL ASS.XML	-

Reclass – CI Uploading

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/CI RECLASS. XML	-
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$\$MSEX PORT\$\$/P YRLRECL ASS.XML	-

Reclass – TR Uploading

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/T RRECLAS S.XML	-
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/T RRECLAS S.XML	-

Reclass – OC Uploading

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/O CRECLAS S.XML	-
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/O CRECLAS S.XML	-

Reclass – SN Uploading

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/S NRECLAS S.XML	-
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/S NRECLAS S.XML	-

Reclass – CR Uploading

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/C RRECLAS S.XML	-
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/C	-

			RRECLAS S.XML	
--	--	--	------------------	--

Reclass - Transaction Submission

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
PARM_FILE	Parameter File	Protected	\$\$AMSPARM\$\$/RCLS_PARM.txt	-

Data Object Name RCLS_PARM_HDR

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
EXEC MODE	Execution Mode	Required	Report Only	Valid Values are Report Only, Generate Transactions Only, and Report and Generate Transactions
AUTO OF IND	Automatic Overflow Indicator	Required	"N"	Select if executing the Automatic Overflow Recapture.
TRANS FM DT	Transaction From Date	Optional	Blank	Enter the date the reclassification process should begin.
TRANS TO DT	Transaction To Date	Optional	Blank	Enter the date the reclassification process should end.
RCLS PARM ID	Job ID	Required	Blank	Enter any numeric value.

Data Object Name RCLS_PARM_DTL

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
DEPT CD	Department	Required	Blank	Enter the Department Code wanting to reclassify.
MJR PROG CD	Major Program	Required	Blank	Enter the Major Program Code wanting to reclassify.
PROG CD	Program	Optional	Blank	Enter the Program Code wanting to reclassify.

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
PPC CD	Period Code	Optional	Blank	Enter the Period Code wanting to reclassify.
FPRFL CD	Funding Profile	Required	Blank	Enter the Funding Profile wanting to reclassify.
PHASE CD	Phase	Optional	Blank	Enter the Phase wanting to reclassify.

Any number of detail records can be inserted with different COA elements for one Header record.

Process can be run in Update mode only when the process has been run in the report only mode with the same Reclass Parameter ID specified in the custom parameters header record.

Inputs and Selection Criteria – Normal Reclassification

Run Mode	Input	Selection Criteria
Report Only	Reclassification Parameter Screens (RCLS_PARM_HDR and RCLS_PARM_DTL)	User defined Chart of Account Parameters

Run Mode	Input	Selection Criteria
	Funding Split Log (FSL)	<p>Transaction date is within the From / To range, if entered.</p> <p>Department matches parameter.</p> <p>The cost accounting elements are all “active” (Major Program, Program, Phase, PPC, Funding Profile).</p> <p>Transaction is not discarded and is the “final version”.</p> <p>Funding Profile is not expired (that is, the To date on the Funding Profile is less than the run date).</p> <p>Amount is not \$0.00.</p> <p>Not excluded by FES (that is, Exclude Flag on FSL).</p> <p>The Reclass Exclusion flags on Major Program, Program and Program Phase are “N”.</p> <p>The cost accounting elements (if entered) match those entered on the custom parameter.</p> <p>Note:</p> <p>If a record has been BES the date used for selection is the RCL_DOC_REC_DT not the transaction date.</p> <p>If a record is selected from FSL but it is not the “Final” version within the date range on FSL the “Final” version is the one used for the reclass (even if the dates fall outside of the range), an exception is written to the exception report.</p>
Generate Transactions Only	Reclassification Parameter Screens (RCLS_PARM_HDR and RCLS_PARM_DTL)	User-defined Chart of Account parameters
	XML generated in the report only mode.	
Report and Generate Transactions	Reclassification Parameter Screens (RCLS_PARM_HDR and RCLS_PARM_DTL)	User defined Chart of Account Parameters
	Funding Split Log (FSL)	Same selection criteria as above

Sort Criteria – Normal Reclassification

None

Outputs – Normal Reclassification

Run Mode	Outputs
Report Only	Reclass report temp table (RCLS_RPT_TEMP) Parameters Report Exception Report Report of all selected records reclassified No updates to the database take place
Generate Transaction Only	Modified AD, CH, IAT, IET, MD, PYRL, CI, TR, OC, SN and CR transactions in XML file format Processed AD, CH, IAT, IET, MD, PYRL, CI, TR, OC, SN and CR transactions by SysManUtil Report of all transactions that failed during submission
Report and Generate Transactions	Exception Report Report of all selected records reclassified Modified AD, CH, IAT, IET, MD, PYRL, CI, TR, OC, SN and CR transactions in XML file format Processed AD, CH, IAT, IET, MD, PYRL, CI, TR, OC, SN and CR transactions by SysManUtil Report of all transactions that failed during submission

Problem Resolution – Normal Reclassification

If the process fails due to any reason then perform the corrective actions:

- Check the Log file for any errors that may have occurred while the program was running
- Check whether an XML file of the name XML for charge transactions is created in the D:\AMSADV30\RTFiles\ directory for today’s date and time. This will only be done if the process is run in a mode that is intended to generate transactions.
- Delete the XML file if it exists.
- Re-run process starting from first job step.
- This batch program produces new Advantage transactions, which are subject to the line limit functionality constraints. Sites should ensure that they run this job with parameters set to ensure that the created transactions are within the line limit controls.

Technical Flow – Automatic Overflow Recapture

When the Automatic Overflow Recapture indicator is set to “Yes”, the reclassification process will look for any programs that have “Overflow” costs (that is, the overflow priority has billed balances). If there are any of these programs with “Overflow”, which also have had a funding change (that is, additional dollars added, a new priority added ahead of the “Overflow” priority – determined by a previous priority with available dollars), the reclassification process will automatically include these programs in the reclassification process. This process eliminates the need to identify all Programs with “Overflow” costs in the reclassification parameters.

It is important to note that the Automatic Overflow Recapture indicator only affects the input to the reclassification process. Once these “Overflow” Reclassifications are included in the input, they are processed just like any other Reclassification.

This Automatic Overflow Recapture process is executed when the following conditions exist:

- The Automatic Overflow Recapture indicator is set to “Yes.”
- This process can be executed when funds are spent from a temporary budget and funds become available in a real budget.
- A temporary budget is associated with a Funding Priority with an Overflow Indicator of ‘Y’. Online edits are in place to ensure that only the last Funding Priority within a Profile can be defined as Overflow.
- This process looks at the Funding Split Log (FSL) for Funding Priorities that are marked as Overflow, if it finds one it will see if there are any funds available in the other Priorities associated with the same Funding Profile.
- A line selected from FSL for Overflow Recapture will trigger a modification transaction with only the RECLASS_IND_FL changed – which will then trigger the reclassification via FES logic.

The Automatic Overflow Recapture Process can be executed in two modes:

- Globally (This accomplished when only the Department or Department and COA elements are entered as a user-defined parameter on the Reclassification Parameter Screen.)
- Based dates entered as user-defined parameter on the Reclassification Parameter Screen

The difference between these two modes is that when the Automatic Overflow Recapture process is executed “Globally” the process will select all the lines associated with priorities defined as “Overflow.” If either the cost accounting elements or To/From dates are entered on the custom parameter, only records for the specified cost accounting elements and or date range will be selected.

Below are other items to be considered when executing the Automatic Overflow Recapture process:

- When cost accounting elements as well as the dates are entered on the custom parameter, all transactions within the date range for the specified Chart of Account elements are selected.
- If only the cost accounting elements are entered on the custom parameter then all the transactions for the specified cost accounting elements are selected.
- If only the date is entered on the custom parameter without any Chart of Account elements, all the records within the specified date range are selected.
- The Department Code is mandatory on the custom parameter; it is considered a “Global” run if the date fields are not entered on the custom parameter.
- If the run is “Global” all the funding profiles that have the “exclude from Global Overflow” indicator set to “Yes” are excluded.
- The process when not executed as global sets the “Exclude from Global Overflow” indicator for funding profiles affected during the run. The flag will not be set for a Funding Profile if the modified transaction for that funding profile has failed to submit successfully.

At the time record has been selected by the Automatic Overflow Recapture process the following steps are executed:

- a. Selected Records are written to the reclass report temp table (RCLS_RPT_TEMP)
- b. Following Reports are created from the reclass report temp table:
 - Parameters Report that lists the parameters for the current run.
 - General report that lists all the records to be processed in the current run.
 - Exception Report that lists all the records that have been modified or cancelled beyond the date specified, if any, on the parameters screen. It also lists all the records for which funding Profile is inactive.
- c. A modification transaction is created in XML file format
- d. All lines that have been selected be reclassified will have the “Reclass Indicator Flag” (RECLASS_IND_FL) incremented by one to trigger processing.
- e. The XML files are imported to the database using SysManUtil.
- f. The newly imported modification transactions are submitted using SysManUtil.
- g. If transaction(s) fails to submit, due to any reason, the offline process generates an exception report, which is then used by the “Reclass Cleanup” to remove the entries that failed during the submission phase. In addition, the “Reclass Cleanup” process will generate a report of all failed transactions with error messages pertaining to their failure.
- h. If the run was not “Global” (that is, date fields are not populated) all profiles that were successfully reclassified will have the “Exclude from Global” indicator set to “Yes.”

In the report only mode steps, a to d are executed.

In the update mode steps, e to h are executed.

In the report and update mode, all the steps are executed.

When to Run – Automatic Overflow Recapture

At any time

Run Sequence – Automatic Overflow Recapture

1. Reclassification Adjustments
2. System Maintenance Utility (for loading of Charge transactions)
3. System Maintenance Utility (for loading of AD transactions)
4. System Maintenance Utility (for loading of MD transactions)
5. System Maintenance Utility (for loading of IET transactions)
6. System Maintenance Utility (for loading of ITA transactions)
7. System Maintenance Utility (for loading of PYRL transactions)
8. System Maintenance Utility (for loading of CI transactions)
9. System Maintenance Utility (for loading of TR transactions)
10. System Maintenance Utility (for loading of OC transactions)
11. System Maintenance Utility (for loading of SN transactions)
12. System Maintenance Utility (for loading of CR transactions)
13. System Maintenance Utility (for submitting of Charge transactions)

14. System Maintenance Utility (for submitting of AD transactions)
15. System Maintenance Utility (for submitting of MD transactions)
16. System Maintenance Utility (for submitting of IET transactions)
17. System Maintenance Utility (for submitting of ITA transactions)
18. System Maintenance Utility (for submitting of PYRL transactions)
19. System Maintenance Utility (for submitting of CI transactions)
20. System Maintenance Utility (for submitting of TR transactions)
21. System Maintenance Utility (for submitting of OC transactions)
22. System Maintenance Utility (for submitting of SN transactions)
23. System Maintenance Utility (for submitting of CR transactions)
24. Reclassification Clean Up Process

Step 1 has to be run before any of the other steps are run.

Step 2 to 12 can be run in any order and any of these steps can be disabled also.

Step 13 to 23 can be run in any order and any of these steps can be disabled also.

Step 24 should be run only after at least one of the steps from 13 to 23 has been run.

Parameters – Automatic Overflow Recapture

Batch Parameters

COMMIT_BLOCK:- Specifies the Commit Block size. Valid value is any positive integer number. Process will commit the transaction after the number of the records specified in this parameter is processed. This parameter is defaulted to 100 if it is not specified or an invalid value is entered.

PARM_ID:- Specifies the Parameter Id that is the link to the Custom Parameters listed below. Valid value is the appropriate positive integer number. This parameter is defaulted to 0 if it is not specified or an invalid value is entered.

Custom Parameters

Data Object Name RCLS_PARM_HDR

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
EXEC MODE	Execution Mode	Required	Report Only	Valid Values are Report Only, Generate Transactions Only, and Report and Generate Transactions
AUTO OF IND	Automatic Overflow Indicator	Required	"N"	Select if executing the Automatic Overflow Recapture
TRANS FM	Transaction From	Optional	Blank	Enter the date the reclassification

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
DT	Date			process should begin.
TRANS TO DT	Transaction To Date	Optional	Blank	Enter the date the reclassification process should end.
RCLS PARM ID	Job ID	Required	Blank	Enter any numeric value.

Data Object Name RCLS_PARM_DTL

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
DEPT CD	Department	Required	Blank	Enter the Department Code wanting to reclassify
MJR PROG CD	Major Program	Optional	Blank	Enter the Major Program Code wanting to reclassify
PROG CD	Program	Optional	Blank	Enter the Program Code wanting to reclassify
PPC CD	Period Code	Optional	Blank	Enter the Period Code wanting to reclassify
FPRFL CD	Funding Profile	Optional	Blank	Enter the Funding Profile wanting to reclassify
PHASE CD	Phase	Optional	Blank	Enter the Phase wanting to reclassify

Any number of detail records can be inserted with different COA elements for one Header record.

Process can be run in Update mode only when the process has been run in the report only mode with the same Reclass Parameter ID specified in the custom parameters header record.

Inputs and Selection Criteria – Automatic Overflow Recapture

Run Mode	Input	Selection Criteria
Report Only	Reclassification Parameter Screens (RCLS_PARM_HDR and RCLS_PARM_DTL)	User defined Chart of Account Parameters

Run Mode	Input	Selection Criteria
	Funding Split Log (FSL)	<p>Transaction date is within the From / To range, if entered.</p> <p>Department matches parameter.</p> <p>Funding Profile, Funding Priority, and Funding Line are marked as "Overflow".</p> <p>Transaction is not discarded and is the "final version".</p> <p>Funding Profile is not expired (that is, the To date on the Funding Profile is less than the run date).</p> <p>Amount is not \$0.00.</p> <p>Budget is available in another Funding Priority within the Funding Profile.</p> <p>Not excluded by FES (that is, Exclude Flag on FSL).</p> <p><u>For Global Runs</u></p> <p>The "Exclude from Global" indicator on the Funding Profile table is set to "No".</p> <p><u>For Specific Runs</u></p> <p>The cost accounting elements must match those entered on the custom parameter (that is, Reclassification Parameter Screen).</p> <p>Note:</p> <p>If a record has been BES the date used for selection is the RCL_DOC_REC_DT not the transaction date.</p> <p>If a record is selected from FSL but it is not the "Final" version within the date range on FSL the "Final" version is the one used for the reclass (even if the dates fall outside of the range), an exception is written to the exception report.</p>
Generate Transactions Only	Reclassification Parameter Screens (RCLS_PARM_HDR and RCLS_PARM_DTL)	User defined Chart of Account Parameters
	XML generated in the report only mode.	
Report and Generate Transactions	Reclassification Parameter Screens (RCLS_PARM_HDR and RCLS_PARM_DTL)	User defined Chart of Account Parameters
	Funding Split Log (FSL)	Same selection criteria as above

Sort Criteria – Automatic Overflow Recapture

None

Outputs – Automatic Overflow Recapture

Run Mode	Outputs
Report Only	Reclass report temp table (RCLS_RPT_TEMP) Parameters Report. Exception Report Report of all selected records reclassified No updates to the database take place
Generate Transaction Only	Modified AD, CH, IAT, IET, MD, PYRL, CI, TR, OC, SN and CR transactions in XML file format Processed AD, CH, IAT, IET, MD, PYRL, CI, TR, OC, SN and CR transactions by SysManUtil Report of all transactions that failed during submission
Report and Generate Transactions	Exception Report Report of all selected records reclassified Modified AD, CH, IAT, IET, MD, PYRL, CI, TR, OC, SN and CR transactions in XML file format Processed AD, CH, IAT, IET, MD, PYRL, CI, TR, OC, SN and CR transactions by SysManUtil Report of all transactions that failed during submission.

Problem Resolution – Automatic Overflow Recapture

If the process fails due to any reason then perform the corrective actions:

- Check the Log file for any errors that may have occurred while the program was running.
- Check whether an XML file of the name XML for charge transactions is created in the D:\AMSADV30\RTFiles\ directory for today’s date and time. This will only be done if the process is run in a mode that is intended to generate transactions.
- Delete the XML file if it exists.
- Re-run process starting from first job step.

This batch program produces new Advantage transactions, which are subject to the line limit functionality constraints. Sites should ensure that they run this job with parameters set to ensure that the created transactions are within the line limit controls.

2.1.21 Reimbursable Expense Adjustment Process

Reimbursable Expense Adjustment is the process which identifies reimbursable budget records (lines) that will potentially result in a reimbursement request that exceeds the award amount.

Description

Reimbursable Expense Adjustment:

The complete Reimbursable Expense Adjustment process has 4 steps:

1. Generate Report or Generate JV XML or Generate Report and JV XML
2. Load JV XML
3. Submit JV XML
4. JV Exception Report

1. Generate Report or Generate JV XML Transaction or Generate Report and JV XML

The process picks up records matching the Budget Structure specified on the Reimbursable Expense Adjustment Parameter (REAP) table. The records are processed from the Budget Structure Level 1, calculation of Program Cost, Available Amount, Available % at Level 1 is done. If the Available % is less than or equal to Tolerance % the record will be picked up for further processing. The Level 2 Program cost, Actual %, Prorated Award and Prorated Program Costs are calculated and populated on the REIM_EXP_ADJ_TMP table.

If the process runs in Report mode, the records are read from the REIM_EXP_ADJ_TMP table and the Report is generated.

If the process runs in Update Mode, the records are read from the REIM_EXP_ADJ_TMP table and the JV Transaction XML is generated.

If the process runs in Report and Update Mode, the records are read from the REIM_EXP_ADJ_TMP table and a report is generated as well as the JV Transaction XML.

NOTE: The process returns a return code of Non Fatal Error if the Reimbursable Expense Job is run in Report Mode so that the other jobs are set to 'Inactive', otherwise the job returns a return code of 'Successful'.

2. Load JV XML

The Load JV XML process uploads the XML File into the JVC.

3. Submit JV XML

The Submit JV XML process submits all of the JVC transactions that were created in the previous step.

4. JV Exception Report

All of the rejected JVC transactions in the Exception File (generated by the Submit JV XML if any JVC transactions are rejected) are picked up one after the other. Details of the errors are written to the Exception Report.

When to Run

The Reimbursable Expense Adjustment Process may be run on an ad-hoc basis or on a regular schedule. It is generally envisioned that this process would be executed prior to the Reimbursement Selection and Calculation process so that the resulting JV lines would be selected for the Reimbursement cycle.

Steps to Run the Adjustment Process (Job Sequence)

When the chain job is run in either Generate Report or Generate Report and JV Transactions, the program clears the temporary data source it uses to generate Journal Vouchers. The Reimbursable Expense Adjustment Parameters (REAP) table also stores a value that cannot be seen online, which indicates a single parameter as being run as Generate Report only. A subsequent run of the chain in Generate JV Transactions Only mode will use that setting to determine which parameter ID run as Generate Report Only that will now have Journal Vouchers created. For these two reasons, the following scenarios will exist for running the chain.

Step 1: Run the chain as Generate Report and JV Transactions

or

Step 1: Run the chain as Generate Report

Step 2: Manually review the report

Step 3: Run the chain as Generate JV Transactions Only

If the chain is run multiple times as Generate Report, the only records available to generate Journal Vouchers for will be those records selected from the last Generate Report run. Records from the prior report runs will no longer exist. They can be selected again with another Generate Report or Generate Report and JV Transactions run.

If the chain is run once as Generate Report with certain selection criteria followed by a run as Generate Report and JV Transactions with different selection criteria, data from the first run is no longer available for a Generate JV Transactions Only run. They can be selected again with another Generate Report or Generate Report and JV Transactions run.

Major Input

Reimbursable Expense Adjustment

Reimbursable Expense Adjustment Parameters (REAP) table

Budget Structure tables of budget type as Reimbursable

Reimbursable Expense Adjustment Temporary table (REIM_EXP_ADJ_TMP)

JV Exception Report

JV Submit Exception File

Output

Reimbursable Expense Adjustment

JVC Transaction

JV Exception Report

JVC Transaction Exception Report (Unprocessed JVC transactions) – This report is generated at the end of the Process.

Parameters

Reimbursable Expense Adjustment

The Reimbursable Expense Adjustment process is comprised of 4 batch jobs that are chained together. Each step in the process is described below.

Job	Parameter	Description	Default Value
Reimbursable Expense Adjustment	AMSEXPORT (** Refer to Note: Assumptions for SWBP on page no. 5)	Export location for JVC transaction	NA
	AMSLOGS (** Refer to Note: Assumptions for SWBP on page no. 5)	Logs location for JVC transaction	NA
	AMSPARM (** Refer to Note: Assumptions for SWBP on page no. 5)	Parameter location for JVC Transaction	NA
	CLIENT_NM	Client Name for Report	NA
	EXCEP_REP_FILE_NM	Exception File Name	JVCExcep.txt
	FILE_NM	Export File Name	JVCDocument.xml
	PARAM_FILE	Load Parameter File	JVCLoad.txt
	PARAM_ID	Parameter Id	NA
	PROG_CST_FOR	Program Costs Formula	NA
	SUBMIT_FILE	Submit Parameter File	JVCSubmit.txt

Job	Parameter	Description	Default Value
Load JV XML	PARAM_FILE	Parameter File (.txt)	\$\$AMSPARM\$\$/JVCLoad.txt

Job	Parameter	Description	Default Value
Submit JV XML	PARAM_FILE	Parameter File (.txt)	\$\$AMSPARM\$\$/JVSubmit.txt

Job	Parameter	Description	Default Value
JV Exception Report	AMSLOGS (** Refer to Note: Assumptions for SWBP on page no. 5)	Logs Location at JV Exception Report Job	NA
	AMSPARM (** Refer to Note: Assumptions for SWBP on page no. 5)	Parameter Location at JV Exception Report Job	NA
	PARAM_FILE	Parameter File(.txt)	JVCLoad.txt

Custom Parameters

If you are running the job in the Financial application, you can click on the Custom Parameter link and select a parameter record. If you are running the job in the Administration application, enter a valid Parameter ID from the Custom Parameter table from the Financial application.

Reimbursable Expense Adjustment Parameters (REAP) Table

Parameter (Attribute) Name	Description (Caption)
PARAM_ID	Parameter Id
RUN_MODE	Run Mode
REIM_STRU_ID	Reimb Budget
REIM_STRU_NM	Budget Name
TOL_PC	Tolerance %
DEPT_CD	Department
DOC_DEPT_CD	Transaction Dept
DOC_UNIT_CD	Transaction Unit
PFX	Prefix
FES_PSCD_ID	FES Posting Code

Parameter (Attribute) Name	Description (Caption)
FY	Accounting Fiscal Year
BFY	Budget Fiscal Year
PER	Transaction Period
REC_DT	Record Date (* Refer to Note: Pivot Date/Year Validation , while entering the date)
DOC_CD	Transaction Code
CLRNG_FUND_CD	Clearing Fund
BES_PSCD_ID	BES Posting Code
LAST_RUN	Last Run
FES_EVNT_TYP_ID	FES Event Type
BES_EVNT_TYP_ID	BES Event Type

Sort Sequence

The XML file is created in the following order:

1. Department
2. Major Program

Selection Criteria

Records are retrieved from the Budget Structure’s Level 1 and Level 2 table based on the parameter selected on the Reimbursable Expense Adjustment Parameter (REAP) table. Calculation for Level 1 Program cost, Available Amount and Available % is done. If the calculated Available % on Budget Structure’s Level 1 is less than or equal to the Tolerance % on the REAP table the records on Budget Structure’s Level 2 are selected and further calculation for Level 2 Program Cost, Prorated Award and Prorated Program Cost is done and the record is selected for calculating the JV Adjustment Line Amount.

Problem Resolution

The whole process is implemented with checkpoints. If the process fails for any reason (such as the network is down or the server is down), then the process restarts from the last point where it stopped.

2.1.22 Reimbursement

Job Name	Reimbursement
Recommended Frequency	Should be run frequently (daily or weekly). Must be run more frequently than the other Reimbursement chains (Reimbursement Generation and Reimbursement Output).
Single Instance Required	Yes.
Can be Restarted?	This is a chain job. There are notes on each job in the chain about its restartability.
Reports Generated	From certain jobs– see details on the jobs.

Overview

The full scope of the Cost Accounting Reimbursement functionality is divided into the three required processes: Reimbursement Selection and Calculation, [Reimbursement Generation](#), and [Reimbursement Output](#). There are also two optional processes: [Reimbursement Recycling](#) and [Reimbursement Expense Adjustments](#). The three required processes are inter-related and must be executed in a sequential order. This particular run sheet describes the Reimbursement Selection and Calculation chain. Please refer to the appropriate run sheets for detailed descriptions related to the other processes.

The Reimbursement Selection & Calculation chain has four jobs:

- 1) [Reimbursement Selection \(ReimSelection\)](#)
- 2) [Load Transactions from XML \(ReimLoadCH\)](#)
- 3) [Submit Transactions \(ReimSubmitCH\)](#)
- 4) [Cleanup \(ReimCleanUp\)](#)

The acceptable job return codes (configured in the Configure Chain Job section of the Job Setup in CGI Advantage) for the jobs in the Reimbursement Selection & Calculation Chain are delivered to be set to “Successful”. As with all CGI Advantage chain jobs, these acceptable return codes are configurable and may be changed to meet certain requirements. For simplicity this run sheet assumes the chain is configured as delivered, including parameter settings.

The fourth job in the chain (Cleanup) is a special ability of the Selection & Calculation chain to perform its own cleanup after an incomplete run. It is important that you do not deactivate this job or change the acceptable job return code without carefully considering the options.

Run Modes

The majority of the cost accounting offline processes can be submitted in three different run modes to provide flexibility when requesting complex and intricate offline processes. A user, likely a cost accounting manager, selects one of three run modes on the Reimbursement Selection Parameter (REIMSEL) table prior to executing the Reimbursement Selection and Calculation chain job. The following are the descriptions of the run modes available:

- **“Generate Report Only”** - This run mode allows offline processes to be submitted without the final/true database updates occurring. The Report-Only mode provides a what-if analysis tool to the users so they can see what records will be selected by the

offline process without having the permanent portions of the database updated. (“Generate Report Only” mode does update some temporary tables, but the next run will clear them away.) The two reports generated in this mode are both from the Reimbursement Selection job: the Selection Report and the Discrepancy Report. Both reports are discussed in more detail in the [“Reimbursement Selection”](#) job run sheet. The Report-Only run mode provides the users and system administrators the capability to make the necessary adjustments to any cost accounting set-up tables prior to the actual database updates by processing transactions. After users make the corrections they can re-run the offline process in Report-Only run mode again to check the set-up data corrections before updating the database. The Report-Only run mode can be run as many times as necessary until the users determine that the what-if results are as expected.

When running in “Generate Reports Only” mode, it is best to create an instance of the chain with the second, third, and fourth jobs deactivated. To not do so will result in the “Load Transactions from XML” job ending with a ‘failed’ return code because an XML file could not be found.

- **“Generate Charge Transactions Only”** - The purpose of this run mode is to perform the updates to the database *with the results of the previous run in “Generate Report Only” run mode*. This is appropriate if the previous run of the process was in the Report-Only run mode and after analyzing the expected results the users decided that the results were correct then this run mode will perform all necessary database updates.

It is important to note that the “Generate Charge Transactions Only” run mode *does not re-process the data*; its sole responsibility is to take the data processed during the previous Report-Only run mode, residing in the temporary tables, and perform updates to the database to record a complete run along with the creation and processing of Charge transactions. Consequently, the “Generate Charge Transactions Only” run mode cannot run unless the previous run mode was Report-Only because the “Generate Charge Transactions Only” run mode does not re-process and/or re-calculate the data.

- **“Generate Charge Transactions and Report”** - The purpose of this run mode is to combine the first two run modes, “Generate Report Only” and “Generate Charge Transactions Only”, into one job submission. If the users are certain the cost accounting set-up data is correct, then they may want to utilize this all-in-one run mode to save time from having to perform two submission processes.

The “Generate Charge Transactions and Report” mode can also be run after “Generate Report Only”. Even though the selection should be the same, some sites appreciate having output reports from the final process. **Warning:** running in this manner requires more processing time than “Generate Charge Transactions Only” because the selection and calculation logic will be performed all over again.

Run Frequency

As this is the first chain in a sequence of chains executed to complete the reimbursement process, the frequency of this job is required to be at a minimum the same as that of the Reimbursement Generation Chain. Reimbursement and Selection can be run even more frequently to process activity as needed to prevent a backlog of data processing in just a single run before Reimbursement Generation.

Major Input

The following are a complete list of inputs. Depending on the mode of the job, only some will be used. Later details in the discussion of the first job (Reimbursement Selection) will provide more information about the use of these inputs.

- 1) Batch Parameters
- 2) Cost Accounting Journal (JRNL_CA)
- 3) Reimbursement Suspense File (CA_REIM_SUSP)
- 4) Reimbursement Selection Parameters (R_REIM_SEL_PARM)
- 5) Reimbursement Suspense Temporary File (CA_REIM_SUSP_TMP)
- 6) Split Differential File (SPLIT_DIFF)
- 7) Automatic Transaction Numbering (AUTO_DOC_NO)
- 8) Posting Code (R_PSCD)
- 9) Many Chart of Account tables
 - i) Major Program (R_MJR_PROG)
 - ii) Funding Priority (R_FPRTY)
 - iii) Funding Profile (R_FPRFL)
 - iv) Funding Line (R_FLINE)
 - v) Program (R_PROG)
 - vi) Program Phase (R_PHASE_PROG)
 - vii) Program Period (R_PPC)
 - viii) Object (R_OBJ)
 - ix) Activity (R_ACTV)
 - x) Task Order (R_TASK_ORD)
 - xi) One or more of the Funding Profile Inference tables that start with page codes of 'FPI' (INF_PRFL_#)
 - xii) Journal Log

Major Output

- 1) Temporary Reimbursement Suspense File (CA_REIM_SUSP_TMP)
- 2) Temporary Ready for Reimbursement File (CA_REIM_TMP)
- 3) Selection Report
- 4) Discrepancy Report
- 5) Reimbursement Suspense File (CA_REIM_SUSP)
- 6) Ready for Reimbursement File (CA_REIM)
- 7) Split Differential File (SPLIT_DIFF)
- 8) Journal Log (JRNL_LOG)
- 9) CH Transaction XML file (ReimSelChDocument.xml)
- 10) Various processing files: RMSC_PARM.txt, ReimExcepRep.txt, & ReimSelParmInfo.txt

Chain Job Return Codes

The following table shows the potential return codes for the Reimbursement Selection & Calculation Chain. Note that the Chain job will end with the highest return code across all of the jobs.

Return Code	Condition
Successful (1)	All the jobs ends successfully
Warning (4)	One of the jobs in the chain ends with a return code of "Warning"

Non Fatal Error (8)	One of the jobs in the chain ends with a return code of “Non Fatal Error”
Failed (12)	One of the jobs in the chain ends with a return code of “Failed”
Terminated (16)	One of the jobs in the chain ends with a return code of “Terminated”
System Failure (20)	One of the jobs in the chain ends with a return code of “System Failure”

Problem Resolution

For performance reasons, the Identify and Archive Stale Gaps job should be run periodically to remove gaps in journal records being tracked and processed by this program. Please refer to the “Identify and Archive Stale Gaps” run sheet in the *CGI Advantage Financial – General Accounting Run Sheets* guide for more details on scheduling and recommended parameters.

Please refer to the individual job “Problem Resolution” section for more details.

Reimbursement Chain: Reimbursement Selection Job

Job Name	ReimSelection
Recommended Frequency	Not applicable since this job is part of the chain.
Single Instance Required	Yes
Can be Restarted?	Yes
Reports Generated	Discrepancy Report Selection Report

Overview

This job has seven different steps to perform, each of which is discussed in more detail below:

- 1) Validation of direct batch parameters and those of the Reimbursement Selection Parameter ID
- 2) Record selection
- 3) Record evaluation
- 4) Back end funding split
- 5) Generate XML file of Back end split Charge transactions (except in “Generate Report Only” mode)
- 6) Update necessary processing tables
- 7) Generate Reports

Details on the processing steps:

- 1) Validation of direct batch parameters and those of the Reimbursement Selection Parameter ID

This step validates the parameters supplied during job submission and from the Reimbursement Selection Parameter (REIMSEL) page.

2) Record selection

This step gathers appropriate accounting data previously charged against all cost accounting major programs associated with reimbursement agreements. Reimbursement eligible transaction information is supplied from two sources. The first is the Cost Accounting Journal (JRNL_CA) and the second is the Reimbursement Suspense (CA_REIM_SUSP) table. The program selects from the journal all records written to the journal since the last run of the process by reading the Journal Log (JRNL_LOG) entry with a process ID of "RSEL" that was created by the last run. The program goes on to search for additional journal records that may have filled in previously recorded gaps on the Journal Log table. Next, all records are selected from the Reimbursement Suspense table, which is a listing of postings that failed the reimbursement logic during a previous reimbursement cycles. All such records are fed into each new run to evaluate if they are eligible for reimbursement now.

3) Record evaluation

Not all Cost Accounting Journal records will be reimbursed. There is a great deal of filtering based on the posting code and what Chart of Account codes are eligible for reimbursement. Reimbursable transactions are defined in the system by a posting code attribute called **Funding Split Indicator**. When a posting has that field set to *Split for Reimbursement* it will be selected for further evaluation by the process. Posting codes with the other settings – *Split for Reporting* and *N/A* – will not be selected. The delivered posting code model has pre-set the Funding Split Indicator to *Split for Reimbursement* for the following categories of posting codes: Cash Expenditures, Charges, and Revenue Credits Transactions.

Once a journal record is found for split selection, editing is performed to see if certain Chart of Account (COA) codes used on the record are all marked as 'eligible' for reimbursement and not marked as 'suspended'. The fields that are included in this analysis are the Activity and Object, plus all of the Cost Accounting fields: Major Program, Program, Phase, Program Period Code, Funding Profile, Funding Priority, and Funding Line. A flag called **Reimb Eligible** exists on the Object and Activity COA tables and the Cost Accounting tables contain another setting, the **Reimb Status**, both of which are used to determine if the record can be reimbursed. Additionally if the record contains a Task Order Billing Major Program, then the Reimb Status of the Task Order is also checked to determine if the record can be reimbursed.

4) Back end funding split

The application of funding split rules for costs associated to Major Programs, classified as back-end split (BES), is one of the primary processing activities of this job step. Cost accounting structures (Major Programs) classified as reimbursable can be split utilizing one of two methods: Front-end Split (FES) and Back-end Split (BES). FES logic applies the funding split rules during transaction processing of accounting transactions (that is, on-line and real time) through the use of what is known as the Front End Split routine. The Major Programs that are identified as BES have the funding split rules calculated during this Reimbursement Selection and Calculation Chain instead of at the time of transaction posting. In other words, BES accounting transactions enter the Selection and Calculation process without the funding participation details identified (that is, un-split). The method utilized by Selection and Calculation process for applying the funding split rules is to place the un-split accounting transaction lines into a transaction (a BES "Charge") and submit the transaction.

5) Generate XML file of Back end split Charge transactions (except in "Generate Report Only" mode)

Information from the application of BES rules is then translated into generated transactions in XML form. The transaction type used is the Charge (CH). The transaction code used as well as other information to determine the full transaction ID of each generated comes from the Reimbursement Parameters ID used. In generating these transactions, two limits are available to control the number of accounting lines in any one transaction. The first is a direct batch input parameter and the other may found on the Transaction Component Requirements table in the Administration Application. A record on that table for a **Transaction Component** of *CH_DOC_ACTG* with a **Property** of *MAX_LINE_LIMIT* would supply another limit with the value in the **Property Value** field. The more restrictive of these two limits will be used.

Prior to release 3.6, Back End Split Charge transactions were generated using the REMISEL parameter ID's Accounting Period, Fiscal Year, and Budget Fiscal Year. If those values were not specified on the parameters, transactions were using the Application Date to determine these fields. In release 3.6, a change was made so that if the REMISEL parameters do not specify Accounting Period, Fiscal Year, and Budget Fiscal Year then the values for these fields will assume the values from the related input record.

6) Update necessary processing tables

There are several processing tables and files updated by this job step. Two are used for report generation – Temporary Reimbursement Suspense File (CA_REIM_SUSP_TMP) and Temporary Ready for Reimbursement File (CA_REIM_TMP). If the process is run in one of the update modes, the Reimbursement Suspense File (CA_REIM_SUSP) that was used as input to the job is also an output as all remaining records that are found to be Suspended for Reimbursement are written to the file for the next run of the chain. Another table updated for the next run of the chain is the Journal Log (JRNL_LOG) table. A record is written to this table with a **Process ID** of *RSEL* with the beginning and ending journal records selected by the run from the Cost Accounting Journal. The last table updated is the Ready for Reimbursement File (CA_REIM). All accounting transactions that have been split either by FES or BES funding rules and are not Suspended for Reimbursement are populated to this table that is the input file for the next jobs in the reimbursement process - Reimbursement Generation and Reimbursement Output. If the run mode is "Generate Report Only" then only the temporary tables are output.

7) Generate Reports

Two reports are also produced by this job step, except when run in "Generate Charge Transactions Only" mode, which produces no reports. The first is the Selection Report which lists all records that were selected from the Cost Accounting Journal and the Reimbursement Suspense File. The report is divided into two sections: Ready and Suspended. All records listed in the Ready section come from the CA_REIM_TMP file and those in the Suspended section come from the CA_REIM_SUSP_TMP file. The second report is the Discrepancy Report which lists FES records from the CA_REIM_TMP file that, if re-split as of the report run, would infer a different funding profile. The determination of data for this report is done by taking selected FES records and finding the funding profiles that would currently default that are different than those originally defaulted. The program uses a temporary data store, the Split Differential (SPLIT_DIFF) table to do this activity.

The following table shows the various steps that the REIMSEL job performs and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation & Parameter File creation	<ul style="list-style-type: none"> • Validating Batch Parameters <ul style="list-style-type: none"> ○ If a parameter is invalid, the invalid value will be displayed in the log along with the error

Process Steps	Messages
	<p>message.</p> <ul style="list-style-type: none"> ○ <i>If all parameters pass validation: Parameters are valid</i> ● Removing temporary gaps created by last run ● Temporary gaps removed ● Creating file: \$\$\$MSROOT\$\$\$/Parms/ReimSelParmInfo.txt (<i>To store Parameter ID</i>) <ul style="list-style-type: none"> ○ If the parameter file cannot be created, an appropriate error as to why is issued.
<p>2. Selection and Evaluation of records</p>	<ul style="list-style-type: none"> ● Emptying Temporary Tables ● Tables Emptied ● Start Gap processing <ul style="list-style-type: none"> ○ Select Gap records from JRNL_CA ○ Selection of Gap records completed ○ Processing the selected records <ul style="list-style-type: none"> ▪ Number of Gaps processed # (<i>where # is the commit block size defined as an input parameter. # is 0 if no records for processing</i>) ○ Processing Gaps completed ● Processing CA_REIM_SUSP records <ul style="list-style-type: none"> ○ Selecting Front & Back End Split Records <ul style="list-style-type: none"> ▪ <i>If no records are selected: No records found (and drop to selecting Front End Split Records below)</i> ○ Selection of records completed ○ Processing the selected records <ul style="list-style-type: none"> ▪ Number of records processed # (<i>where # is the commit block size defined as an input parameter.</i>) ○ Processing completed for Front & Back End Split Records ○ Selecting Front End Split records <ul style="list-style-type: none"> ▪ <i>If no records are selected: No records found</i> ○ Selection of records complete ○ Selecting Back End Split records <ul style="list-style-type: none"> ▪ <i>If no records are selected: No records found</i> ○ Selection of records complete ○ Processing the selected records) <ul style="list-style-type: none"> ▪ Number of records processed # (<i>where # is the commit block size defined as an input parameter.</i>) ▪ <i>If no records are selected: No</i>

Process Steps	Messages
	<ul style="list-style-type: none"> ○ records found ○ Processing completed for Back End Split records ● Processing CA_REIM_SUSP records completed ● Processing JRNL_CA <ul style="list-style-type: none"> ○ Selecting Front & Back End Split Records ○ Selection of records completed ○ Processing the selected records <ul style="list-style-type: none"> ▪ Number of records processed # (where # is the commit block size defined as an input parameter.) ▪ If no records are selected: No records found ○ Processing completed for Front & Back End Split Records ○ Selecting Front End Split records (for split differential determination) <ul style="list-style-type: none"> ▪ If no records are selected: No records found ○ Selection of records complete ○ Selecting Back End Split records <ul style="list-style-type: none"> ▪ If no records are selected: No records found ○ Selection of records complete ○ Processing the selected records (To determine how to split) <ul style="list-style-type: none"> ▪ Number of records processed # (where # is the commit block size defined as an input parameter.) ▪ If no records are selected: No records found ○ Processing completed for Back End Split records ● Processing JRNL_CA completed
<p>3. Generating Reports</p>	<ul style="list-style-type: none"> ● Generating Reports <ul style="list-style-type: none"> ○ Select data from CA_REIM_TMP and CA_REIM_SUSP_TMP tables for Selection Report ○ Select data from SPLIT_DIFF table for Discrepancy Report ○ Rendering Selection report started ○ Rendering Selection report completed ○ Rendering Discrepancy report started ○ Rendering Discrepancy report completed ● Reports Generated

Process Steps	Messages
<p>4. Creation of CH XML file</p>	<ul style="list-style-type: none"> • Generate posting started - CH transaction creation <ul style="list-style-type: none"> ○ Start processing gaps to update JRNL_LOG ○ Selecting gap records ○ Selection of gap records completed ○ Updating the selected records <ul style="list-style-type: none"> ▪ Number of gaps updated: # (<i>where # is the commit block size defined as an input parameter.</i>) ▪ <i>If no gap records existed: No gaps to update</i> ○ Updating gaps completed ○ Creating CH xml file <ul style="list-style-type: none"> ▪ Number of transactions appended to CH xml: # (<i>where # is the commit block size defined as an input parameter.</i>) ▪ <i>If no transactions to be created: No transactions found</i> ○ CH xml file created ○ Creating parameter file \$\$\$AMSR00T\$\$/Parms /RMSC_PARM.txt • Generate posting completed • Run Ended

Restartability Information

This job can be restarted if it fails for parameter validation errors in the first step. The job also contains checkpoint logic that allows for restarts when any one of the three major record processing steps (previous gaps, suspense file, or new journal records) occur. Please see the later section for Job Return Codes to get more details on when restarting is allowed and is the correct action to take.

Major Input

- Tables

The following are a complete list of items, however, depending on the mode of the job, only some will be used.

- 1) Batch Parameters
- 2) Cost Accounting Journal (JRNL_CA) supplies new accounting data against reimbursement agreements.
- 3) Reimbursement Suspense File (CA_REIM_SUSP) supplies accounting data from previously selected Cost Accounting Journal records that were evaluated in a previous run and found to be not eligible at the time.
- 4) Reimbursement Selection Parameters (R_REIM_SEL_PARM) supplies many input parameters in addition to those directly input into the batch jobs.
- 5) Reimbursement Suspense Temporary File (CA_REIM_SUSP_TMP) is used for report creation.

- 6) Split Differential File (SPLIT_DIFF) allows for the determination of FES records that would have a new Funding Profile inferred that is different than the one originally used.
- 7) Automatic Transaction Numbering (AUTO_DOC_NO) supplied transaction ID information only when generating transactions
- 8) Posting Code (R_PSCD) supplies information for the program to determine what accounting activity is eligible for reimbursement
- 9) Many Chart of Account tables provide information for the program to determine if the accounting activity is eligible
 - a) Major Program (R_MJR_PROG): Split Type Reimbursement Status, and Task Order Billing
 - b) Funding Priority (R_FPRTY)
 - c) Funding Profile (R_FPRFL)
 - d) Funding Line (R_FLINE)
 - e) Task Order (R_TASK_ORD)
 - f) Program (R_PROG): Reimbursement Status & Billing Agreement Date
 - g) Program Phase (R_PHASE_PROG): Reimbursement Status & Billing Agreement Date
 - h) Program Period (R_PPC): Reimbursement Status
 - i) Object (R_OBJ): Reimbursement Status
 - j) Activity (R_ACTV): Reimbursement Status
 - k) One or more of the Funding Profile Inference tables that start with page codes of 'FPI' (INF_PRFL_#): Reimbursement Status and for BES Major Programs they provide a Funding Profile
- 10) Reference tables provide field values to populate the output tables:
 - a) Program (R_PROG): Drawdown Group (DRWDWN_GRP), Drawdown Department (DRWDWN_DEPT_CD), and Drawdown Unit (DRWDWN_UNIT_CD).
 - b) Major Program (R_MJR_PROG): Split Type (SPLT_IND), and if Drawdown Group (DRWDWN_GRP), Drawdown Department (DRWDWN_DEPT_CD), and Drawdown Unit (DRWDWN_UNIT_CD) are not present on Program Table (R_PROG) then infer these fields from Major Program Table.
 - c) Funding Line (R_FLINE): Output Bank Account (BANK_ACCT_CD), CMIA Method (CMIA_CALC_METH), Reimbursement Output Format (CR_TYP_IND), Reimbursement Customer (CUST_ID), Reimbursement Frequency (REIM_FREQ), and Split Percentage (SPLT_PC).

Batch Parameters

Those direct parameters supplied through the batch interface:

Parameter	Description	Default Value
AMSEXPORT	Export location for the ReimSelection Job File location where the XML file of transactions will be written. Leave the default value to use the application-wide location (** Refer to Note: Assumptions for SWBP on page no. 5),	\$\$AMSROOT\$\$/Export/Import

Parameter	Description	Default Value
AMSLOGS	Logs location for the ReimSelection Job File location where logs will be written. Leave the default value to use the application-wide location (** Refer to Note: Assumptions for SWBP on page no. 5).	\$\$AMSROOT\$\$/Logs
AMSPARM	Parameter location for the ReimSelection Job File location where the parameters from this job step will be written so they can be passed to subsequent steps. Leave the default value to use the application-wide location (** Refer to Note: Assumptions for SWBP on page no. 5)	\$\$AMSROOT\$\$/Parms
CLIENT_NM	Client name for reports. This name will appear in the heading of both reports if populated.	(blank)
MAX_ACTG_LN_PER_DOC	The Maximum Accounting Lines per generated Transaction. The number entered here will be the limit of accounting lines for any one transaction. This limit must be less than or equal to the maximum line limit in Transaction Component Requirements (DCREQ) for CH_DOC_ACTG. DCREQ may be found in the CGI Advantage Administration application. If this input parameter is left blank, the program will compare the value of accounting line limit from SOPT and the maximum line limit from DCREQ for CH_DOC_ACTG and use the lower of the two limits for this setting. If a value is entered here, but the maximum line limit in DCREQ is zero or the maximum line limit is not found on DCREQ then the parameter value entered will be used. If no parameter value is entered and no entry exists on DCREQ or the DCREQ limit is zero, then the program will use the accounting line limit from SOPT.	
PARAM_ID	Reimbursement Selection Parameter ID, from the REIMSEL page, that should be used for this run.	(blank)

Parameter	Description	Default Value
	<p>When running the job in the Financial application, you can click on the <i>Custom Parameter</i> link to open the Reimbursement Selection Parameters table to select a parameter record or enter a parameter ID. When running the job in the Administration application, a Parameter ID must be manually entered.</p>	
COMMIT_BLOCK	<p>Commit block size</p> <p>A required block size to control messages detailing the progress of record processing. The block size is also used to govern the amount of updates performed at any one time.</p> <p>Please see the note after this table about Performance Parameters.</p>	1000
SELECT_BLOCK	<p>Select block size (the number of records in each "Select" statement).</p> <p>A required performance parameter, this is the number of records that will be selected at a time from the Reimbursement Suspense table or Cost Accounting Journal.</p> <p>Too low of a value can result in more reading and retrieving time, while too high of a value can demand higher memory capacities for application and database servers.</p> <p>Please see the note after this table about Performance Parameters.</p>	1000
SELECT_BLOCK_FO R_GAPS	<p>Select block Size (Number of gap records in each Select statement).</p> <p>A required performance parameter that is the number of gap records selected at one time to be evaluated for completion in the Cost Accounting Journal.</p> <p>Too low of a value can result in more reading and retrieving time, while too high of a value can demand higher memory capacities for application and database servers.</p> <p>Please see the note after this table about Performance Parameters.</p>	3000

Performance parameters must be configured based on the processing and memory capabilities of the application and database servers. These numbers should be determined during implementation and should only be adjusted if hardware or installation settings change. Once determined and set in Job Setup for the Ledger Engine as the default value, the same value can be used for any subsequent runs without user intervention.

These direct parameters are defined on the Reimbursement Selection Parameters (REIMSEL) page and brought into the process based on the Parameter ID:

Parameter	Description	Default Value
Parameter ID (PARM_ID)	Display Only. This field is system-generated and uniquely identifies each record on the Parameter table. On the Encrypted Batch Parameters (ENBP) table the Parameter ID field is used to uniquely identify parameter values.	(system generated)
Run Mode (RUN_MODE)	Required choice of three values: "Generate Report Only", "Generate Charge Transactions Only", or "Generate Charge Transaction and Report".	Generate Report Only
Transaction Prefix (DOC_PREFIX)	Required field if generating transactions that will be used to retrieve the appropriate Automatic Transaction Numbering (ADNT) entry. Please see the note below this table about ADNT fields.	(blank)
Charge Transaction Code (DOC_CD)	Required field if generating transactions that will be used to retrieve the appropriate Automatic Transaction Numbering (ADNT) entry. Must be a valid transaction code defined to the CH transaction type on the Transaction Control table. Please see the note below this table about ADNT fields.	(blank)
Charge Dept Code	Required field if generating transactions that will be used to retrieve the appropriate Automatic Transaction Numbering (ADNT) entry. Must be a valid department code defined on the Department table. Please see the note below this table about ADNT fields.	(blank)
Charge Unit Code	Required field if generating transactions. Must be a valid unit code defined on the Unit table for the Fiscal Year value (if year not specified then the current	(blank)

Parameter	Description	Default Value
	fiscal year is used) and the Charge Dept Code.	
Fiscal Year	<p>Optional year to be written to transactions generated. Required if the Fiscal Period parameter is used. Must be a valid year defined on the Fiscal Year table.</p> <p>If the Fiscal Year is entered, it will be passed to the generated Charge Transactions.</p> <p>If left blank, the transactions will post to the default Fiscal Year of the Charge Record Date, if used, and if not used then the default Fiscal Year of the Application Date will be used.</p> <p>Please see the note below this table about ADNT fields.</p>	(blank)
Fiscal Period	<p>Optional period to be written to transactions generated. Must be valid on the Accounting Period (APD) table.</p> <p>If the Fiscal Period is entered, it will be passed to the generated Charge Transactions.</p> <p>If left blank, the transactions will post to the default Fiscal Period of the Charge Record Date, if used, and if not used then the default Fiscal Period of the Application Date will be used.</p>	(blank)
Charge Record Date	<p>Optional date to be written to transactions generated.</p> <p>If left blank, the transactions will post to the Application Date.</p>	(blank)
Budget Fiscal Year	<p>Optional year to be written to transactions generated.</p> <p>If the Budget Fiscal Year (BFY) is entered, it will be passed to the generated Charge Transactions.</p> <p>If the Fiscal Year (FY) is entered and the BFY is not, then the Charge Transaction common date routine will default the BFY to be equal to the FY, unless a multi-year value (9999) is inferred.</p> <p>If neither the FY nor the BFY is entered then the common date routine will default both the FY and BFY to the current fiscal year based on the CLDT entry associated with the current date.</p>	(blank)

Parameter	Description	Default Value
DOC_TYP (not shown online)	The inferred transaction type of the CH Transaction Code entered.	(blank)
DOC_CAT (not shown online)	The inferred transaction category of the CH Transaction Code entered.	(blank)
START_REC (not shown online)	A field that stores the first selected record from the Cost Accounting Journal selected for processing. In report and update modes the field stores the beginning record number of the Cost Accounting Journal where new records were selected. The range is stored here initially as updates to the Journal Log do not occur until later in the chain job.	(blank)
END_REC (not shown online)	A field that stores the last selected record from the Cost Accounting Journal selected for processing. In report and update modes the field stores the ending record number of the Cost Accounting Journal where new records were selected. The range is stored here initially as updates to the Journal Log do not occur until later in the chain job.	(blank)
LAST_RUN (not shown online)	A flag that indicates the parameter record's last run mode. This parameter is not displayed online and is only updated by the system.	(blank)

ADNT Edit: Fields for Fiscal Year, Transaction Code, Transaction Department, and Transaction Prefix must exist on the ADNT table. If Fiscal Year is not specified, the current fiscal year (default fiscal year of the Application Date) is used.

Major Output

The first five outputs are produced when the job is run in “Generate Report Only” or “Generate Charge Transaction and Report” mode.

- i) The Temporary Reimbursement Suspense File (CA_REIM_SUSP_TMP) is updated with the Front End Split (FES) records that were already split but are Suspended for Reimbursement, in addition to all records that need to undergo Back End Split (BES). When the run mode generates transactions, the temporary table is cleared at the end of processing.
- ii) The Temporary Ready for Reimbursement File (CA_REIM_TMP) is updated with the FES records that were Eligible for Reimbursement. When the run mode generates transactions, the file is cleared at the end of processing.
- iii) A Selection Report listing all records that were selected from the Cost Accounting Journal and the Reimbursement Suspense File to be reimbursed.

- iv) A Discrepancy Report listing records from the CA_REIM_TMP file that, if re-split as of the report run, would infer a different funding profile.
- v) An update to the Reimbursement Selection Parameters record with information in the Start Record, End Record, and Last Run fields.

The remainder of the Outputs are only produced by the update modes: “Generate Charge Transaction and Report” or “Generate Charge Transaction Only”.

- vi) The Reimbursement Suspense File (CA_REIM_SUSP) is updated with all FES and BES records that were already split but are Suspended for Reimbursement, in addition to all BES records where the generated BES Charge transaction failed to process successfully and are still un-split.
- vii) The Ready for Reimbursement File (CA_REIM) is updated with all FES and BES records that are reimbursable and Eligible for Reimbursement. This file serves as input to the Reimbursement Generation and Reimbursement Output processes.
- viii) The Journal Log (JRNL_LOG) is updated with a record indicating the time/date stamp of the run when transactions are generated. This record serves to define a starting point in the Cost Accounting journal for the next run of the Reimbursement Selection & Calculation chain job.
- ix) The Split Differential File (SPLIT_DIFF) is updated with those FES records found to have a different Funding Profile inferred value than the one previously determined.
- x) A CH Transaction XML file (ReimSelChDocument.xml). This file will be loaded and submitted by later jobs in the chain. When the Charge transactions are generated and successfully submitted, the following tables are updated:
 - (1) Header Components (DOC_HDR and CH_DOC_HDR)
 - (2) Accounting Line Components (DOC_ACTG and CH_DOC_ACTG)
 - (3) Posting Line Catalog (PSTNG_LN_CAT)
 - (4) R_IN_UNID will have the numbering record used for transaction ID creation incremented
 - (5) Front End Split Log (FSL)
 - (6) Cost Accounting Journal (JRNL_CA) if journal posting occurs in synchronous mode
 - (7) One or more Cost Accounting budgets will be updated with the Back End Split amount. (At least one Reimbursable budget must exist for reimbursable Major Programs.)
- xi) Processing Files (RMSC_PARM.txt & ReimSelParmInfo.txt) for later jobs in the chain.

Job Return Codes

The following table shows the potential job return codes for the ReimSelect job.

Return Code	Condition
Successful (1)	All selected records are processed successfully
Warning (4)	No records found in table JRNL_LOG (gap processing), JRNL_CA, or CA_REIM_SUSP to process
Non Fatal Error (8)	The job does not end with this status.
Failed (12)	The job will fail under the following conditions: <ul style="list-style-type: none"> - Parameters are invalid - Run time exceptions for unexpected situations.

Return Code	Condition
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

The selected records will be sorted so that all the records having similar Cost Accounting COA are grouped together. The type of record selected determines the sorting criteria as follows:

Record Type	Sort criteria
Front & Back End Split	Descending order of <ul style="list-style-type: none"> • DEPT_CD • MJR_PROG_CD • PROG_CD • PPC_CD • DOC_ID • DOC_ACTG_LN_NO • PSTNG_PR_TYP • DOC_VERS_NO • FPRFL_CD • FPRTY_CD • FLINE_CD • DOC_REC_DT Ascending order of <ul style="list-style-type: none"> • REC_NO of data source table
Front End Split	Descending order of <ul style="list-style-type: none"> • DEPT_CD • MJR_PROG_CD • PROG_CD • PPC_CD • DOC_ID • DOC_ACTG_LN_NO • PSTNG_PR_TYP • DOC_VERS_NO • DOC_REC_DT Ascending order of <ul style="list-style-type: none"> • REC_NO of data source table
Back End Split	Descending order of <ul style="list-style-type: none"> • DEPT_CD

	<ul style="list-style-type: none"> • MJR_PROG_CD • PROG_CD • PPC_CD • DOC_ID • DOC_ACTG_LN_NO • PSTNG_PR_TYP • DOC_VERS_NO • DOC_REC_DT <p>Ascending order of</p> <ul style="list-style-type: none"> • REC_NO of data source table
--	---

The order by is:

- Department (DEPT_CD)
- Major Program (MJR_PROG)
- Program (PROG_CD)
- Program Period Code (PPC_CD)
- Transaction ID (DOC_ID)
- Accounting Line (DOC_AL_LN_NO)
- Posting Pair (PSTNG_PR_TYP)
- Transaction Version (DOC_VERS_NO)
- Funding Profile (FPRFL_CD)
- Funding Priority (FPRTY_CD)
- Funding Line (FLINE_CD)
- Record Date (DOC_REC_DT)
- Record Number (REC_NO)

Selection Criteria

Custom Parameter: Records matching input parameter value selected

Journal Log: Latest record with a Process ID of RSEL

From the Cost Accounting Journal (JRNL_CA)

- Posting Code must be Split for Reimbursement (Funding Split Indicator = 3); and
- Posting Code not C002 (not set to Back end split) (More conditions are checked with back end split posting code C002, such as 1. It's not initial version 2. If version greater than 1 then it should be reclassified record. Third condition checked is if initial version and if transaction is JVA, it's eligible for reimbursement since this is posted to cost accounting journal from Reimbursable Expense Adjustment Process) and
- Major Program must be BES or FES (Split Indicator = 2 or 3) and
- Cost Accounting Journal (CA_JRNL) Record Number is greater than Journal Log (JRNL_LOG) last processed Record Number.

Reimbursement Suspense File (CA_REIM_SUSP): All records selected

Problem Resolution

The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful parameter validation	No action necessary	
Warning (4)	N/A	This step doesn't issue this return code.	
Non Fatal Error (8)	N/A	This step doesn't issue this return code.	
Failed (12)	Required Parameters are not entered Sample Message: Parameter ID cannot be blank	Enter the correct Reimbursement Selection Parameter ID and resubmit the Chain job.	
	Entered Parameters are not valid Sample Message 1: The Parameter record with ID <id> not found. Sample Message 2: Maximum Accounting Lines per Transaction is more than the MAX_LINE_LIMIT set on R_DOC_COMP_RQMTS for CH Accounting Line.	1. Enter a valid Reimbursement Selection Parameter ID and resubmit the Chain job. 2. Enter a value for the MAX_ACTG_LN_PER_DOC parameter that is less than or equal to the value of the MAX_LINE_LIMIT set on R_DOC_COMP_RQMTS for CH_DOC_ACTG.	
	Failed because of runtime exceptions for unexpected situation	Failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	
System Failure	When the job is terminated because of database server or	Reason for the System Failure needs to be investigated before	

Possible Return Codes	Condition	Recommendation	Other Instructions
(20)	network issues	scheduling a new job.	

Step 2: Selection & Evaluation of records. This step will be performed only if the parameter validation is successful.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	The job selected eligible records from at least one of the following: the Journal Log, Cost Accounting Journal, or the Reimbursement Suspense File	No action necessary	
Warning (4)	No records found on any of the three data sources Sample message: No records found on the JRNL_CA	Confirm that the Cost Accounting Journal in fact does not have any new records after ending journal record ID of the last RSEL record on Journal Log. Confirm the CA_REIM_SUSP table had no records Confirm that the Journal Log has records.	
Non Fatal Error (8)	N/A	This step doesn't issue this return code.	
Failed (12)	In this step, the job can fail only because of runtime exceptions for unexpected situation	Failure reason needs to be investigated before scheduling a new job.	No tables have been updated at this point in the process therefore no recovery actions are necessary.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	No tables have been updated at this point in the process therefore no recovery actions are necessary.
System	When the job is	Reason for the	No tables have been

Possible Return Codes	Condition	Recommendation	Other Instructions
Failure (20)	terminated because of database server or network issues	System Failure needs to be investigated before scheduling a new job.	updated at this point in the process therefore no recovery actions are necessary.

Step 3: Generating reports

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	The job created the Reports successfully if run mode produces them.	No action necessary	
Warning (4)	N/A	This step doesn't issue this return code	
Non Fatal Error (8)	N/A	This step doesn't issue this return code.	
Failed (12)	In this step, the job can fail only because of runtime exceptions for unexpected situation	Reason for the failure needs to be investigated. Schedule a new job after resolving the issues.	All the table updates will be rolled back if the job encounters any runtime exception therefore no recovery actions are necessary.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	All the table updates will be rolled back if the job encounters any runtime exception therefore no recovery actions are necessary.
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the System Failure needs to be investigated before scheduling a new job.	All the table updates will be rolled back if the job encounters any runtime exception therefore no recovery actions are necessary.

Step 4: Creation of CH XML file – This step will be performed only if the BES section successfully splits a record

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	XML file is created successfully	No action necessary	
Warning (4)	N/A	This step doesn't issue this return code.	
Non Fatal Error (8)	N/A	This step doesn't issue this return code	
Failed (12)	In this step, the job can fail only because of runtime exceptions for unexpected situation	Reason for the failure needs to be investigated. Schedule a new job after resolving the issues.	No tables have been updated at this point in the process therefore no recovery steps are necessary.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	No tables have been updated at this point in the process therefore no recovery steps are necessary.
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the System Failure needs to be investigated before scheduling a new job.	No tables have been updated at this point in the process therefore no recovery steps are necessary.

Reimbursement Chain: Load Transactions from XML Job

Job Name	ReimLoadCH
Recommended Frequency	Not applicable since this job is part of the chain.
Single instance Required	Yes
Can be restarted?	No
Reports Generated	No

Overview

After the ReimSelection job has completed and the CH transaction XML file has been created, the next job in the chain, Load Transactions job, uploads the XML file to the Transaction Catalog

using the SysManUtil utility. This process first validates the input parameters (i.e. whether the parameter file exists in the specified directory). If the file exists, this job loads the transactions based on the information provided in the parameter file. The job will fail if the parameter file name is blank or the file is not found on the specified directory. If no Back End Split records were created by the Reimbursement Selection job, then the XML file will exist but it will be empty. If the XML file is empty, this job will complete with a return code of "Warning".

Restartability Information

This job cannot be restarted. See the "Problem Resolution" section below for actions to follow if the load process fails.

Major Input

- Batch Files
CH Transaction XML file (ReimSelChDocument.xml)

- Transactions

N/A

- Batch Parameters

Those direct parameters supplied through the batch interface:

Parameter	Description	Default Value
ACTN_CD	SysManUtil Action Code for Import : 171	171
BYPS_ADNT_FL	Bypass Auto Transaction Numbering Value should be 'true' as transaction numbers are assigned in XML creation and not at time of loading.	true
COMMIT_BLOCK	Commit Block Size A required performance parameter that is the number of transactions that will be saved to the application as the XML file is loaded. Too low of a value can result in more time spent selecting and committing small blocks of work. Too high of a value can demand higher memory capacities for application and database servers. Please see the note after this table about Performance Parameters.	10
FILE_NAME	File name of the XML file created from the first job (Reimbursement Selection).	\$\$AMSEXPORT\$\$/ ReimSelChDocume nt.xml

Performance parameters must be configured based on the processing and memory capabilities of the application and database servers. These numbers should be determined during implementation.

Major Output

Charge Transactions on the Transaction Catalog in “draft” phase.

Job Return Codes

Refer to the “SMU Transaction Upload” and “SMU Transaction Submit” actions run sheets in the *CGI Advantage Financial – Utilities Run Sheets* guide for the possible job return codes that will be issued in this job.

Sort Criteria

No sorting is performed – the XML file is processed from start to finish in the order it was created.

Selection Criteria

None – the entire XML file is processed.

Problem Resolution

It is important that if the job fails to load some or all the transactions from the XML file into the Transaction Catalog, then no new instance of this chain should be executed until all the transactions in the XML file are loaded into the transaction catalog. This is because the XML file will get overwritten if another instance of this chain ran successfully.

Those transactions that could not be loaded are written to a new XML file in the Import Error directory with a system-generated file name. Find the file by matching the run time of this job to the created on date of the file in the error data directory. The matching XML file found will contain all the charge transactions that could not load. The reason for them not loading should be addressed and the transactions loaded before proceeding further in this chain.

There are two options for loading the failed XML transactions. The option you choose should be based on whether the failed transactions were a small percentage of the total transactions or the majority of the total. Also, if the next job in the chain (“Submit Transactions”) started you can only use the first option. (Note: Whether or not the “Submit Transactions” job begins will be determined by the acceptable return codes. You may want to consider setting the acceptable return code for the “Submit Transactions” job to be “Successful” instead of warning – this will allow you to use either solution option described below.)

If only a small number of transactions failed to load, or the “Submit Transactions” job submitted, you should copy the *error* file to Import/Export directory. After you resolve the issues that prevented loading, submit a separate SMU job to load the error XML file from the Import/Export directory.

If the Submit Transactions job did not start, and a high percentage of transactions failed, there is a second option for resolution. First, you must determine why the transactions failed to load and fix the problem. The next action would be to go online and discard the transactions or use an SMU job to discard those transactions that did successfully load (they have not submitted yet). It is recommended to verify *ReimSelParmInfo.txt* (*Parameter ID from this file is used in third step*) and *RMSC_PARM.txt* (*This is SysManUtil Parameter file for submit step*) that are generated by the first step and are stored at appropriate location. Now the “Load Transactions from XML” job can be restarted or a new Reimbursement Selection & Calculation chain can be submitted with the first job (Reimbursement Selection) disabled.

There is a third option when not all transactions from the XML file were loaded. → This option has the same requirement that the reason for the failure to load should be addressed first. The next

step would be to go online or use an SMU job to discard those transactions that did successfully load. After discarding all the loaded transactions, since at this point journal log and final reimbursement tables are not updated, the chain can be re-run. It will re-select all the records that were processed in prior failed run. This option will need only one manual intervention required which is running SMU job to discard successfully loaded transactions. After this, the chain can be re-run by using an automated batch scheduler (like cybermation) continuing the rest of the batch cycle, if this is to be done in the middle of the batch process cycle.

The following table shows the possible return codes and recommendations for this job.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters are valid and all the transactions are loaded successfully	N/A	N/A
Warning (4)	The input file is empty. Sample message: No records found on the input file	Make sure that the Reimbursement Selection job creates the input file with records and creates in the directory the "Load Transactions from XML" job is looking.	
	This return code will be issued when the job fails to load some of the transactions. Sample Message: Unable to load all the transactions into the catalog.	Analyze the reason for the failure, resolve the issue and load all the rejected transactions. Until then, no new instance of this chain should be executed.	See "Problem Resolution" notes above this table for more details.
Non Fatal Error (8)	This return code will be issued when the job failed to load all the transactions.	Analyze the reason for the failure, resolve the issue and load all the rejected transactions. Until then, no new instance of this chain should be executed.	See "Problem Resolution" notes above this table for more details.
Failed (12)	This return code will be issued when the parameters are not valid. Sample Message: Parameter file could not be located/read	Make sure that the parameter file exists in the specified folder and schedule a new chain with the prior jobs disabled. Until then, no new instance of this chain should be executed.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	Failed because of runtime exceptions for unexpected situation	Analyze the reason for the failure, resolve the issue and schedule a new chain with the prior jobs disabled. Until then, no new instance of this chain should be executed.	
Terminated (16)	Job is terminated manually by the user.	Analyze the reason for the termination, resolve the issue and schedule a new chain with the prior jobs disabled. Until then, no new instance of this chain should be executed.	
System Failure (20)	Job is terminated because of database server or network issues	Analyze the reason for the system failure, resolve the issue and schedule a new chain with the prior jobs disabled. Until then, no new instance of this chain should be executed.	

Reimbursement Chain: Submit Transactions Job

Job Name	ReimSubmitCH
Recommended Frequency	Not applicable since this job is part of the chain
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	No

Overview

The Submit Transactions job uses the SysManUtil utility to submit each charge transaction loaded by the “Load Transactions from XML” job to update the application.

Restartability Information

This job cannot be restarted. See the “Problem Resolution” section below for actions to follow if the submit process fails.

Major Input

- Batch Files

SMU job parameter file - RMSC_PARM.txt

- Transactions

Charge Transactions loaded to the transaction catalog from the previous job (“Load Transactions from XML”).

Batch Parameters

Those direct parameters supplied through the batch interface:

Parameter	Description	Default Value
ACTN_CD	SysManUtil Action Code for Import : 162	162
EXCEP_REP_FILE_NM	Exception Report File Name	\$\$AMSLOGS\$\$/ReimExcepRep.txt
EXCEP_REP_IND	Exception Report Indicator where (1 = Detailed, 2 = Failed Transactions, 3 = Processed Transactions, 4 = Failed Accounting Lines)	4
PARAM_FILE	Parameter location for the ReimSelection Job File contains information passed from the first job (Reimbursement Selection) to the later jobs in the chain. (** Refer to Note: Assumptions for SWBP on page no. 5)	\$\$AMSPARM\$\$/RMSC_PARM.txt

Major Output

- Submitted Back End Split Charge transactions (or rejected transactions)
- ReimExcepRep.txt

Job Return Codes

Refer to the “SMU Transaction Upload” and “SMU Transaction Submit” actions run sheets in the *CGI Advantage Financial – Utilities Run Sheets* guide for the possible job return codes that will be issued in this job.

Sort Criteria

No sorting is performed – the charge transactions are submitted in order by Transaction ID.

Selection Criteria

None – the input parameter file instructs the SMU on which transactions to submit.

Problem Resolution

It is important that if this job ends with a return code of “Failed”, “Terminated” or “System Failure”, that no new instance of this chain should be executed until all the transactions loaded in the previous job are submitted or are cleaned by the fourth job– Cleanup. If the BES charge transactions do not submit successfully, no future reimbursement for that program will occur.

Even if this job ends with a return code of “Successful”, that is not a guarantee that all transactions submitted successfully. Those that didn’t are logged in the Job Log along with the errors issued.

If the transactions failed to submit, then determine the transaction errors and correct them. After the data is corrected, you can manually submit the transactions or schedule a new Submit Transactions job within a new chain with the prior jobs disabled to submit the remaining transactions.

Unlike other chain jobs that create transactions, this one has a fourth job in the chain (“Cleanup”) that cleans up rejected transactions by discarding them and marking their source records for reprocessing the next chain run. When the Cleanup job is not disabled before submitting the chain, rejected charge transactions will be removed from the application, so there is no chance for manual submission or selection by another SMU job. The fourth job should be allowed to proceed as when it removes rejected transactions from the application, it updates the Reimbursement Suspense File with information from those transactions so an attempt will be made to split those BES records in the next Reimbursement Selection & Calculation run.

The following table shows the possible return codes and recommendations for each.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the transactions submitted successfully	N/A	N/A
Warning (4)	N/A	This job doesn’t issue this return code.	N/A
Non Fatal Error (8)	N/A	This job doesn’t issue this return code.	N/A
Failed (12)	This return code will be issued when the input parameter is not found in the specified directory. Sample Message: Parameter file could not be located/read	Make sure that the parameter file exists in the specified folder and schedule a new chain with the prior jobs disabled. Until then, no new instance of this chain should be executed.	See “Problem Resolution” notes above this table for more details.

Possible Return Codes	Condition	Recommendation	Other Instructions
	Failed because of runtime exceptions for unexpected situation	Analyze the reason for the failure, resolve the issue and schedule a new chain with the prior jobs disabled. Until then, no new instance of this chain should be executed.	See “Problem Resolution” notes above this table for more details.
Terminated (16)	Job is terminated manually by the user.	Analyze the reason for the termination, resolve the issue and schedule a new chain with the prior jobs disabled. Until then, no new instance of this chain should be executed.	See “Problem Resolution” notes above this table for more details.
System Failure (20)	Job is terminated because of database server or network issues	Analyze the reason for the system failure, resolve the issue and schedule a new chain with the prior jobs disabled. Until then, no new instance of this chain should be executed.	See “Problem Resolution” notes above this table for more details.

Reimbursement Chain: Cleanup Job

Job Name	ReimCleanUp
Recommended Frequency	Not applicable since this job is part of the chain
Single Instance Required	Yes
Can be Restarted?	Yes
Reports Generated	Yes

Overview

The Cleanup Job is essential to subsequent processing of accounting activity through the remaining reimbursement chain jobs as well as for the next run of the Reimbursement Selection & Calculation chain. This job performs the following tasks:

1. If Report Only mode then stop with Return Code Successful
2. Set Transaction information from parameter values which are used in BES Charge transactions processing in various steps.
3. Parameter validation
4. Purge the Reimbursement Suspense (CA_REIM_SUSP) and Reimbursement Suspense Temp (CA_REIM_SUSP_TEMP) files so new records can be written from the clean up. These tables are not purged at this stage in the process. Records from CA_REIM_SUSP and CA_REIM are purged in Step #13 before records from Temporary tables are moved to these tables. Records from CA_REIM_SUSP_TMP and CA_REIM_TMP tables are purged as part one of the first step in Reimbursement Selection batch when Mode is not Generate Postings.
5. Obtain all failed accounting lines from the flat file that was generated by SysManUtil from the previous job
6. For each of the failed lines:
 - a. Update corresponding CA_REIM_SUSP_TMP record with STATUS_IND = CH_REJECTED and LST_PRCES_ACT = Transaction error information, so that this record can be moved to CA_REIM_SUSP at later step and can be reprocessed the next time Selection and Calculation is executed.
 - b. Log the line as an error for the report to be generated from this job
 - c. If the errors contain more than 1500 characters only the first 1500 characters will show. If later runs of the process attempt to create Back-End Split charges and still fail, then the new errors will be appended to the bottom of the message. If the new errors will make the total block more than 1500 characters, then the earlier errors (from prior runs) will be truncated so that the more recent errors will show. Due to this truncation, it may appear that the earlier error messages now start mid-sentence; this is the normal behavior.
7. Delete Accounting Lines marked as Failed. This is done by looking at transaction Accounting Lines (TR_DOC_ID and TR_DOC_ACTG_LNO) on CA_REIM_SUSP_TMP records where Status is CH_REJECTED.
8. Resubmit the remaining transactions to process the successful accounting lines.
9. Discard all remaining rejected transactions from the Transaction Catalog
10. Generate the Rejected Transactions Report, listing the rejected transactions along with errors.
11. Obtain all the successful accounting lines from CA_REIM_SUSP_TMP where Split Indicator is BES and (STATUS_IND =CH_NOT_CREATED OR ((STATUS_IND =CH_SUBMITTED AND FLINE_CD IS NULL and get the split posting lines from the posting line catalog and check if any COA are marked 'Suspended for Reimbursement'
 - a. If so, then write the record to the Reimbursement Suspense (CA_REIM_SUSP) table
 - b. If not, then write the record to the Ready for Reimbursement (CA_REIM) table These are the records for which BES CH transactions are successfully submitted and are ready to be billed on the same night / for next processes in the reimbursement sequence.
12. Delete all unsplit Accounting Line records from CA_REIM_SUSP_TMP for successfully submitted transactions.
13. Moves temporary records to the "final" tables:

- a. Moves CA_REIM_SUSP_TMP records to CA_REIM_SUSP.
 - b. Moves CA_REIM_TMP records to CA_REIM.
14. Record update to Journal Log for process ID RSEL
15. Update the Reimbursement Selection Parameters record with Start Record, End Record, and Last Run information

The following table shows the various steps that the Cleanup Job goes through and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> • Validating Batch Parameters <ul style="list-style-type: none"> ○ If the parameter is invalid, the invalid value will be displayed in the log along with the error message. • Parameters are valid
2. Transaction Processing & Table Updates	<ul style="list-style-type: none"> • Opening flat file \$\$AMSROOT\$\$/Logs/ReimExcepRep.txt (To read failed transactions) <ul style="list-style-type: none"> ○ If the parameter file cannot be found, an appropriate error as to why is issued. ○ Number of transactions found: # • Reading file ReimExcepRep.txt completed • Updating status to REJECTED on CA_REIM_SUSP_TMP table for failed transactions <ul style="list-style-type: none"> ○ Number of records updated: # (<i>where # is the commit block size defined as an input parameter.</i>) ○ <i>If no transactions selected:</i> No failed transactions found • Status updated for failed transactions • Selecting REJECTED records from CA_REIM_SUSP_TM • Selection of records completed • Start deleting CH transaction accounting lines for selected records <ul style="list-style-type: none"> ○ Number of accounting lines deleted: # (<i>Where # is the commit block size defined as an input parameter. Block size may be exceeded slightly to delete all lines from a single transaction in one commit.</i>) ○ <i>If no accounting lines deleted:</i> No failed accounting lines found • Deletion of CH transaction accounting lines completed • Selecting failed transactions to resubmit • Selection of failed transactions completed • Resubmitting failed transactions <ul style="list-style-type: none"> ○ Number of transactions resubmitted: # (<i>where # is the commit block size defined as an input parameter.</i>) ○ <i>If no transactions resubmitted:</i> No failed transactions found 100 • Resubmitting of failed transactions completed • Discarding transactions which are not submitted successfully <ul style="list-style-type: none"> ○ Number of transactions discarded: # (<i>where # is the commit</i>

Process Steps	Messages
	<p><i>block size defined as an input parameter.)</i></p> <ul style="list-style-type: none"> ○ <i>If no transactions discarded:</i> No transactions discarded ● Transactions discarded
<p>3. Generate Report & Update Journal Log</p>	<ul style="list-style-type: none"> ● Generating Rejected Transactions report <ul style="list-style-type: none"> ○ Rendering report started ○ Rendering report completed ● Report generated ● Selecting SUBMITTED/CH_NOT_CREATED records from CA_REIM_SUSP_TMP ● Selection of transactions completed ● Start processing transactions for Reimbursement or Suspension <ul style="list-style-type: none"> ○ Number of transactions processed: # (<i>where # is the commit block size defined as an input parameter.</i>) ○ <i>If no records processed for suspension:</i> No records found ● Transactions processed for Reimbursement or Suspension ● Deleting SUBMITTED/CH_NOT_CREATED records from CA_REIM_SUSP_TMP ● Records deleted ● Delete all records from CA_REIM_SUSP ● Copying all records from CA_REIM_SUSP_TMP to CA_REIM_SUSP ● Records copied ● Delete all records from CA_REIM ● Copying all records from CA_REIM_TMP to CA_REIM ● Records copied ● Inserting record into journal log ● Record inserted ● Updating gap status from INTENDED to FINAL ● Gap status updated ● Update Init Unid of Gap Records ● Init Unid updated ● Run Ended

Updating the gap status from INTENDED (6) to FINAL (2) does not occur; this step adds a record to the Journal Log for the PROC_ID = RSEL with Process Run Date, Beginning and Ending Journal Records, Status Flag = 2 (FINAL) and UNID. In other words, the first step adds records to the Journal Log for the PROC_ID = RSEL with Process Run Date, Beginning and Ending Journal Records, Status Flag = 5 (GAP FOUND) and UNID. See print screen of the Journal Log below.

Data Browse : Idaho - PRODFIN (R) (Oracle - O_ADVUSERHRM)

Proc Id	Proc Run Dt	Jrnl Nm	Bgn Jrnl Rec No	End Jrnl Rec No	Run Typ	Init Unid	Ams Row Vers No	Jrnl Id	St Fl	Upid	Bgn Doc Unid	Bgn Doc Ln Id	Err
RSEL	12/30/2005 15:11:38	JRNL_CA	400529	400532	1	1	2	2	5	12905			
RSEL	12/30/2005 15:11:38	JRNL_CA	400678	400683	1	1	2	2	5	12906			
RSEL	12/30/2005 15:11:38	JRNL_CA	400712	400725	1	1	2	2	5	12907			
RSEL	12/30/2005 15:11:38	JRNL_CA	400823	400828	1	1	2	2	5	12908			
RSEL	12/30/2005 15:11:38	JRNL_CA	400853	400858	1	1	2	2	5	12909			
RSEL	12/30/2005 15:11:38	JRNL_CA	400890	400895	1	1	2	2	5	12910			
RSEL	12/30/2005 15:11:38	JRNL_CA	401422	401427	1	1	2	2	5	12911			
RSEL	12/30/2005 15:12:07	JRNL_CA	399995	401869	1	1	2	2	2	12912			
RSEL	12/30/2005 20:35:18	JRNL_CA	402478	402481	1	1	2	2	5	12913			
RSEL	12/30/2005 20:35:22	JRNL_CA	402486	402487	1	1	2	2	5	12914			
RSEL	12/30/2005 20:35:22	JRNL_CA	402504	402549	1	1	2	2	5	12915			
RSEL	12/30/2005 20:38:09	JRNL_CA	401869	401869	1	1	2	2	2	12916			
RSEL	1/2/2006 20:33:59	JRNL_CA	402478	402481	1	1	2	2	5	12950			
RSEL	1/2/2006 20:34:02	JRNL_CA	402486	402487	1	1	2	2	5	12951			
RSEL	1/2/2006 20:34:02	JRNL_CA	402504	402549	1	1	2	2	5	12952			
RSEL	1/2/2006 20:37:05	JRNL_CA	401869	401869	1	1	2	2	2	12953			
RSEL	1/3/2006 20:34:09	JRNL_CA	402478	402481	1	1	2	2	5	12979			
RSEL	1/3/2006 20:34:11	JRNL_CA	402486	402487	1	1	2	2	5	12980			
RSEL	1/3/2006 20:34:11	JRNL_CA	402504	402549	1	1	2	2	5	12981			
RSEL	1/3/2006 20:34:11	JRNL_CA	402602	402639	1	1	2	2	5	12982			
RSEL	1/3/2006 20:34:11	JRNL_CA	402726	402731	1	1	2	2	5	12983			
RSEL	1/3/2006 20:34:11	JRNL_CA	402892	402897	1	1	2	2	5	12984			
RSEL	1/3/2006 20:34:11	JRNL_CA	402978	402983	1	1	2	2	5	12985			
RSEL	1/3/2006 20:34:11	JRNL_CA	403118	403123	1	1	2	2	5	12986			
RSEL	1/3/2006 20:34:11	JRNL_CA	403180	403185	1	1	2	2	5	12987			
RSEL	1/3/2006 20:34:11	JRNL_CA	403325	403326	1	1	2	2	5	12988			
RSEL	1/3/2006 20:34:11	JRNL_CA	403345	403346	1	1	2	2	5	12989			
RSEL	1/3/2006 20:34:11	JRNL_CA	403413	403414	1	1	2	2	5	12990			
RSEL	1/3/2006 20:34:11	JRNL_CA	403427	403428	1	1	2	2	5	12991			
RSEL	1/3/2006 20:34:11	JRNL_CA	403687	403688	1	1	2	2	5	12992			
RSEL	1/3/2006 20:34:11	JRNL_CA	403731	403732	1	1	2	2	5	12993			
RSEL	1/3/2006 20:37:09	JRNL_CA	401870	403808	1	1	2	2	2	12994			

Row 1 of 2420
10:42:03AM 04-Jan-2006

Restartability Information

This job has the Restartability feature as the process has check point logic in the following areas of processing:

- Re-processed failed accounting lines
- Deleted failed accounting lines
- Resubmit failed transactions
- Discard failed transactions
- Generated reports
- Processed successful lines

Major Input

- Batch Files

Flat file generated from offline submit (ReimExcepRep.txt) – from the third job in the chain (“Submit Transactions”).

Text file which stores Reimbursement Selection Parameter ID (ReimSelParmInfo.txt) – from the first job in the chain (“Reimbursement Selection”).

- Transactions

Failed Charge Transactions listed on the ReimExcepRep.txt file from the previous job.

- Batch Parameters

Direct parameters supplied through the batch interface:

Parameter	Description	Default Value
AMSLOGS	Logs location for the ReimSelection Job (** Refer to Note: Assumptions for SWBP on page no. 5)	(blank)
AMSPARM	Parameter location for the ReimSelection Job File contains information passed from the first job (Reimbursement Selection) to the later jobs. (** Refer to Note: Assumptions for SWBP on page no. 5)	\$\$AMSPARM\$\$/RMSC_PARM.txt
CLIENT_NM	Client name for Report	(blank)
COMMIT_BLOCK	Commit Block Size A required performance parameter that is the number of accounting lines that can be deleted at one time. The program will exceed the limit in order to delete all lines in a single transaction at one time. Too low of a value can result in more time spent issuing progression messages. Too high of a value can demand higher memory capacities for application and database servers. Please see the note after this table about Performance Parameters	1000
FILE_REIMSEL_PARM_INFO	Name of text file which stores Reimbursement Selection Parameter ID	ReimSelParmInfo.txt
FILE_REIM_EXCP_REP	Name of Flat file generated from offline submit	ReimExcepRep.txt
SELECT_BLOCK	Select block Size. If not entered then defaulted to 1000. It is the number of records that can be fetched as a single block and stored to be processed. Can be used for Performance tuning. Please see the note after this table about Performance Parameters	1000

Performance parameters must be configured based on the processing and memory capabilities of the application and database servers. These numbers should be determined during implementation and should only be adjusted if hardware or installation settings change. Once determined and set in Job Setup for the Ledger Engine as the default value, the same value can be used for any subsequent runs without user intervention.

Major Output

- Batch Files

Reimbursement Suspense Table (CA_REIM_SUSP)

Ready for Reimbursement (CA_REIM)

Temp files

- Transactions

Failed Charge Transaction lines removed from transactions on the Transaction Catalog and transactions re-submitted.

Failed Charge Transactions removed from Transaction Catalog

Job Return Codes

The following table shows the potential job return codes for the Cleanup job.

Return Code	Condition
Successful (1)	All records and transactions are handled successfully
Warning (4)	When all transactions successfully processed, the job ends with this status.
Non Fatal Error (8)	Job does not end in this status
Failed (12)	The job will fail under the following conditions: <ul style="list-style-type: none"> - Parameters are invalid - Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

N/A

Selection Criteria

N/A

Problem Resolution

The following table shows the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	
Warning (4)	N/A	This step doesn't issue this return code.	
Non Fatal Error (8)	N/A	This step doesn't issue this return code.	
Failed (12)	Required Parameters are not entered Sample Message: Cash Receipt Bank Account code cannot be blank	Enter the correct Bank Account code and schedule a new job.	
	Entered Parameters are not valid Sample Message: Entered Cash Receipt Bank Account is not valid on the Bank Table	Enter a valid Bank account code and schedule a new job.	
	Failed because of runtime exceptions for unexpected situation	Failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	

Possible Return Codes	Condition	Recommendation	Other Instructions
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the System Failure needs to be investigated before scheduling a new job.	

Step 2: Purge:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	
Warning (4)	No transactions rejected	No action necessary.	
Non Fatal Error (8)	N/A	This step doesn't issue this return code.	
Failed (12)	Failed because of runtime exceptions for unexpected situation	Failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the System Failure needs to be investigated before scheduling a new job.	

Step 3: Transaction Processing & Table Updates:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	
Warning (4)	N/A	This step doesn't issue this return code.	
Non Fatal Error (8)	N/A	This step doesn't issue this return code.	
Failed (12)	Failed because of runtime exceptions for unexpected situation	Failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the System Failure needs to be investigated before scheduling a new job.	

Step 3: Generate Report & Update Journal Log

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	
Warning (4)	N/A	This step doesn't issue this return code.	
Non Fatal Error (8)	N/A	This step doesn't issue this return code.	
Failed (12)	Failed because of runtime exceptions for unexpected	Failure reason needs to be investigated before	

	situation	scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the System Failure needs to be investigated before scheduling a new job.	

2.1.23 Reimbursement Generation Process

Job Name	Reimbursement Generation Process
Recommended Frequency	Should be run frequently (daily or weekly) only after the Reimbursement selection job.
Single Instance Required	Yes.
Can be Restarted?	Yes.
Reports Generated	Yes – see details on the jobs.

Description

The Cost Accounting reimbursement functionality is divided into the following three major processes:

1. Selection and Calculation
2. Generation
3. Output

These three processes are inter-related and must be executed in a sequential order to satisfy the requirement of cost reimbursement programs administered by governmental entities. This particular run sheet describes the system administration aspect of Generation. Please refer to the appropriate run sheet for detail descriptions related to the Selection and Calculation and Output processes.

The Generation process performs five key functions:

1. Reads output from the Selection and Calculation process.
2. Selects records based upon user-defined values entered on the Reimbursement Frequency Date table and on the Reimbursement Generation Custom Parameter table.
3. Applies CMIA drawdown rules for programs identified as such.
4. Applies maximum reimbursement amounts for Non-CMIA programs.
5. Processing of negative or modified / reclassified records.

Output from Selection and Calculation: The Reimbursement Generation Process reads the Ready for Reimbursement (CA_REIM) table created by the Reimbursement Selection and Calculation Process and the Reimbursement Holding (R_REIM_HOLD) table. Records are selected from the Ready for Reimbursement and/or the Reimbursement Holding tables based on the Run Date Department Code, Data Source, and Customer Selection Options entered by the user and the Reimbursement Frequency(s) that are applicable from the last run date to the input date.

Reimbursement Frequency Date Table: The Reimbursement Frequency Date table controls record selection for the Reimbursement Generation process in terms of Department and Reimbursement Frequency(ies). If all of the departments can be combined into a common cycle, the option of “All” may be used in the department Code. Departments may still have unique cycles; these should be processed prior to executing the process for “All” departments if they occur in the same cycle. Records are selected according to the billing cycle indicated for the

Frequency Date record (for example, Weekly); records not selected will remain on the Ready for Reimbursement or Reimbursement Holding tables. Frequency Date records that have not been selected specifically since the last execution of the Reimbursement Generation for the Department will also be included in the current process. The record added for any new Department should have the Calendar Date set to the Application Control date.

Applying CMIA Drawdown Rules: Users identify CMIA programs and their CMIA drawdown method (e.g. Average Clearance, Estimated Clearance) at the time the program is established. In cases where the CMIA method dictates a drawdown pattern, records will be split based upon the specified CMIA funding method. CMIA records that should be requested in the current cycle are written to the Reimbursement Request (R_REIM_REQ) table. CMIA records that should not be requested in the current cycle (for example, because of multi-day clearance pattern, or suspended for reimbursement, or have exceptions) are written to the Reimbursement Holding (R_REIM_HOLD) table.

Applying Maximum Reimbursement Amount for Non-CMIA Programs: Users identify Non-CMIA programs and their maximum reimbursement amounts at the time the program is established. In cases where the Non-CMIA records dictate a maximum reimbursement amount, records will be split if the maximum reimbursement amount is exceeded. The total posting amount of Non-CMIA records that are within or equal to the maximum reimbursement amount are selected for reimbursement and written to the Reimbursement Request (R_REIM_REQ) table. Remaining Non-CMIA records that exceeded the maximum reimbursement amount are written to the Reimbursement Holding (R_REIM_HOLD) table.

Processing of Negative or Modified / Reclassified Records: Negative or modified / reclassified records (Transaction Version > 1) that are CMIA applicable are written to the Reimbursement Request (R_REIM_REQ) table. The records are not split as per the CMIA drawdown method. This is done so that negative and modified / reclassified records are selected immediately for reimbursement.

Technical Flow

The Reimbursement Generation Process consists of six steps where the processing sequence is impacted by the selected run mode:

1. Selection of Records
2. Checking the suspended for reimbursement status
3. Splitting records as per CMIA method/Maximum daily amount
4. Generate reports of the records selected from the ready for reimbursement table for processing (Parameters selection report), records that will be included in the current cycle (Request report) and report for all those records that do not have correct data in the CMIA set up table (Exception Report)
5. Writing the processed records to the reimbursement request table or Reimbursement Holding (R_REIM_HOLD) table
6. Deleting processed records from the Reimbursement Request (R_REIM_REQ) table

Step 1 - Selection of Records

The first criterion for selection is the Data Source. Reimbursement Generation will pull records from the Reimbursement Holding (R_REIM_HOLD) and/or Ready for Reimbursement (CA_REIM) based on the value in the Data Source field.

The input Department code and Run Date defined on the custom parameter record are used to read the Frequency Date Table. The processor will get the most recent record for the Department where the Last Run Date from the Frequency Date Table is populated. This occurs when the Generation process is run for the first time for a Department and in this scenario the Run date should be same as the Application Control date.

All records from Last Run Date through the record matching the Run Date of the current Reimbursement Parameter record are selected. All indicated Frequencies from these records will be eligible for selection.

The records from the Data Source that match the department code and any of the frequencies are selected. If the Department is "All", records may be selected from any Department.

The Budget Fiscal Year and Fiscal Year entered on the custom parameter record are used to read the records from Data Source. The Accounting Period entered on the custom parameter record is used conditionally to select the records from Data Source. If the Include Prior Period value is 'Yes' then records with an Accounting Period less than or equal to the Accounting Period entered as the parameter are selected. If the Include Prior Period value is 'No' then records with an Accounting Period equal to the Accounting Period entered as the parameter are selected.

Finally, if the Customer Selection indicator is Not Applicable, no Customer specific selection will be performed. If the Customer Selection is Include, records will only be selected if the Customer equals one of those indicated. If the Customer Selection is *Exclude*, records will only be selected if the Customer does not equal any of those indicated.

Step 2 - Checking the Suspended Status for Reimbursement

This step verifies the suspended status of the record's Chart of Account elements at the following tables.

- Major Program Table (R_MJR_PRG)
- Program Table (R_PROG)
- Funding Profile Table (R_FPRL)
- Funding Priority Table (R_FPRTY)
- Funding Line Table (R_FLINE)
- Phase-Program Table (R_PHASE_PROG)
- Program Period Code Table (R_PPC)
- Task Order Table (TASKORD) (when a Task Order Billing Major Program is used)
- If the record is suspended even at any one level this record is ignored from processing and moved to the Reimbursement Suspense (CA_REIM_SUSP) table with the Status set to *Suspended*.

Step 3 - Splitting Records

First for records that have a CMIA Calculation Method apply the CMIA process applicable:

- Find the clearance pattern for the CMIA method from the CMIA Clearance Table. The CMIA Clearance Table is not applicable for 2 of the 6 methods in use, the Zero Balance Method and Pre-Issuance Funding.
- Generate the drawdown records for all the clearance days.
- Set the CMIA Expected Request Date

- Set the CMIA Expected Drawdown Date
- Set the CMIA Forward Reference Date
- If dollar amount is negative or transaction version > 1 then set the CMIA Forward Reference Date to current cycle day that is, all credits needs to be processed immediately and move these records to Reimbursement Request Table (R_REIM_REQ_TMP).
- For records whose expected request date is today or next day, include these records for the current run that is, set the CMIA Forward Reference Date to current Cycle date and move these records to Reimbursement Request Table (R_REIM_REQ_TMP).
- If CMIA Expected Request Date is greater than next day, move these records to Reimbursement Holding Table (R_REIM_HOLD_TMP).
- When calculating the drawdown dates exclude the holidays and weekends.

Second, for Non-CMIA records if the Funding Line(R_FLINE) has a Maximum Daily Reimbursement Amount then:

- If the running total of the dollar amount of all the records having same Funding Line is less then or equal to the Maximum Daily Reimbursement Amount then write all the group of records matching the funding line to Reimbursement Request Table (R_REIM_REQ_TMP).
- If the running total of the dollar amount of all the records having same Funding Line is greater then the Maximum Daily Reimbursement Amount:
- Write all the group of records matching the funding line with running total of the dollar amount to Reimbursement Request Table (R_REIM_REQ_TMP).
- For the next record, which crosses over the Maximum Daily Reimbursement Amount split the record into two records with the first record having posting amount equal to amount allowable under the Maximum Daily Reimbursement Amount. The second record will have a posting amount equal to the amount by which it exceeds the Maximum Daily Reimbursement Amount.
- Write the first split record to the Reimbursement Request table (R_REIM_REQ_TMP) and second split record along with any other remaining records matching the same funding line to the holding (R_REIM_HOLD_TMP) table.

Finally, for Non-CMIA records if the Funding Line has no Maximum Daily Reimbursement Amount then write all the group of records matching the Funding Line (R_FLINE) to the Reimbursement Request (R_REIM_REQ_TMP) table.

Step 4 - Generate Reports

Generate reports of the records selected from the Ready for Reimbursement table (CA_REIM) for processing (Parameters Selection Report), records that will be included in the current cycle (Request Report) and report for all those records that do not have correct data in the CMIA set up table (Exception Report).

Step 5 - Writing the Processed Records:

Processed records from the temporary Reimbursement Table (R_REIM_REQ_TMP) are appended to the Reimbursement Request Table (R_REIM_REQ). Processed records from the temporary Reimbursement Holding Table (R_REIM_HOLD_TMP) with a status equal to

'Suspended' are written to the Reimbursement Suspense Table (CA_REIM_SUSP) and remaining records are written to the Reimbursement holding table (R_REIM_HOLD).

Step 6 - Deleting Processed Records

Records that have been processed are deleted from the Ready for Reimbursement table (CA_REIM) and Reimbursement Holding Table (R_REIM_HOLD).

Depending on the run mode the above steps will be called:

- In the report only run mode steps 1 to 4 will be called.
- In the update mode steps 5 and 6 will be called.
- In the report and update mode all the steps will be called.

Run Modes

The majority of the cost accounting offline processes can be submitted in three different "run modes" to provide the user community and the system administrators additional flexibility when requesting complex and intricate offline processes. The user selects a Run Mode on the Reimbursement Generation Parameter page prior to running the process. The following are the descriptions of the run modes available in CGI Advantage Financial:

Report Only - The purpose of this "run mode" is to provide the capability for offline processes to be submitted without the final/true database updates occurring. The Report-Only mode provides a "what-if" analysis tool to the users so they can see what potential data impacts will eventually occur by an offline process without having the permanent portions of the database updated. For example, the main output from the Generation process is the group of Reimbursement Request table records that will be subsequently be available for selection by the Output process. The Report-Only run mode can be utilized to produce a detailed report containing the intended output without actually updating the Reimbursement Request table (R_REIM_REQ). Instead the temporary table (R_REIM_REQ_TMP) is updated. In this example, the users would be able to review the "what-if" reports and/or temporary tables to ensure that the process will produce the expected results prior to actually updating the actual Request table. The Report-Only run mode provides the users and system administrators with the capability to make the necessary adjustments to the Advantage business rules found in the cost accounting set-up tables prior to the actual database updates being applied by the offline process results. In other words, if the "what-if" report analysis determines that there are changes required to the Advantage set-up data then the users would be able to make the corrections and then re-run the offline process in Report-Only run mode again to check the set-up data corrections before updating the database. The Report-Only run mode can be run as many times as necessary until the users determine that the "what-if" results are as expected.

Update - The purpose of this "run mode" is to perform the updates to the database with the results of the previous run in Report-Only "run mode". In other words, the if the previous run of a particular offline process was in the Report-Only "run mode" and after analyzing the expected results the users decided that the results were correct then this run mode will perform all necessary database updates. It is important to note that the Update Only run mode does not re-process the data, its sole responsibility is to take the data processed during the previous Report-Only run mode (that is, residing in the temporary tables) and then perform the appropriate database updates to the "permanent" (that is, not temporary) tables. Consequently, the Update Only run mode cannot run unless the previous run mode was Report-Only because the Update run mode does not re-process and/or re-calculate the data.

Report and Update - The purpose of this "run mode" is to combine the first two "run modes" (that is, Report-Only and Update) into one job submission step. It allows the users the capability to have an offline process that updates the database without having to review the "what-if" reports.

For example, if the users are certain the cost accounting set-up data is correct, then they may want to utilize this all-in-one “run mode” to save time from having to perform the two-step submission process (that is, first the Report-Only and then the Update).

When to Run

User’s Discretion – Recommend to be executed on a timely basis (that is daily or weekly). Preferably run after the Reimbursement Selection and Calculation Process.

Run Sequence

Reimbursement Generation Process

Restartability Information

This job has the Restartability feature and can be restarted only when a previous instance of the job ended with a Return Code of *Terminated*. The process has check point logic in the following areas of processing:

- Selection of records
- Processing for Suspended status
- Splitting and processing of records for CMIA
- Splitting and processing of records for Maximum Reimbursement
- Processing of successful records

Job Return Codes

The following table shows the potential job Return Codes for the ReimGeneration job.

Return Code	Condition
Successful (1)	All selected records are processed successfully
Warning (4)	No records found in the R_REIM_REQ or R_REIM_HOLD tables to process.
Non Fatal Error (8)	The job does not end with this status.
Failed (12)	The job will fail under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Inputs

Run Mode	Input
Report Only	Reimbursement Holding Table (R_REIM_HOLD) (Data Source must be “Both” or “Reimbursement Holding”)

	<p>Ready For Reimbursement Table (CA_REIM) (Data Source must be "Both" or "Ready for Reimbursement")</p> <p>Custom Parameter (R_REIM_GEN_PARM)</p> <p>Frequency Date Table (R_REIM_FREQ_DT)</p> <p>CMIA Clearance Table (CMIA_CLR)</p> <p>Funding Line Table (R_FLINE)</p> <p><u>For Suspended Status:</u></p> <p>Major Program Table (R_MJR_PRG)</p> <p>Program Table (R_PROG)</p> <p>Funding Profile Table (R_FPRL)</p> <p>Funding Priority Table (R_FPRTY)</p> <p>Funding Line (R_FLINE)</p> <p>Phase-Program Table (R_PHASE_PROG)</p> <p>Program Period Code Table (R_PPC)</p> <p>Task Order Table (R_TASK_ORD)</p>
Update Only	<p>Reimbursement Request Temp Table (R_REIM_REQ_TMP)</p> <p>Reimbursement Holding Temp Table (R_REIM_HOLD_TMP)</p>
Report and Update	<p>Same as report only and update only.</p>

Outputs

Process Output is dependent on the Run Mode.

Run Mode	Output
Report Only	<p>Reimbursement Request Table (R_REIM_REQ_TMP)</p> <p>Reimbursement Holding Table (R_REIM_HOLD_TMP)</p> <p>Parameter Selection Report</p> <p>Request Report</p> <p>Exception Report</p>
Update	<p>Reimbursement Request Table (R_REIM_REQ)</p> <p>Reimbursement Holding Table (R_REIM_HOLD)</p> <p>Ready For Reimbursement Table (CA_REIM)</p> <p>Reimbursement Suspense Table (CA_REIM_SUSP)</p> <p>Frequency Date Table (R_REIM_FREQ_DT)</p>
Report and Update	<p>Same as in Report Only and Generate Transactions Mode</p>

Report Only Output

- Temporary Reimbursement Request Table (R_REIM_REQ_TMP) – Holds records that are ready for Billing (for the Reimbursement Output Process).
- Temporary Reimbursement Holding Table (R_REIM_HOLD_TMP) – Holds records that are:

- a) 'Suspended' – Suspended for Reimbursement.
 - b) 'On-Hold'– CMIA records not eligible for billing due to Request Date not equal to Today / Next day and Non-CMIA records which have exceeded the maximum reimbursement amount.
 - c) 'Exception' – Due to CMIA clearance records not found or Clearance records do not sum to 100 percent for the Department/PPC or reference Records like Major Program, Funding Line, Task Order not found.
- Reimbursement Generation Parameter table (R_REIM_GEN_PARAM) – Updates the Last Run Flag to Yes (1) for the input parameter record.
 - Parameter Selection Report – Lists all records that were selected from CA_REIM and R_REIM_HOLD table, the report is divided into two sections 'CMIA' and 'Non-CMIA'.
 - Request Report – Lists records from R_REIM_REQ_TMP table that would be included for Request in the current cycle.
 - Exception Report – Lists records from R_REIM_HOLD_TMP table that are not processed due to exceptions or being "Suspended for Reimbursement."

Update Only Output

- Reimbursement Request Table – Records from the temporary table are appended to the actual table.
- Reimbursement Holding table – All records that not 'suspended' are appended from the temporary table to the actual table. All records appended to the Request table are deleted from the Holding table.
- Reimbursement Suspense table – All suspended records from the Temporary Holding table are appended to the Suspense table.
- Ready for Reimbursement table - All records selected are deleted from the Ready for Reimbursement table.
- Reimbursement Frequency Date table – The Last Run Date is updated for the Department Code and Run Date when the Data Source is "Both" or "Ready for Reimbursement". No updates are applied when the Data Source is "Reimbursement Holding".
- Reimbursement Parameter table - Updates the Last Run Flag to No (0) for the input parameter record.

Parameters

Batch Parameters

Field Name	Description (Caption)	Default Value
PARAM_ID	Selected Parameter ID	0
CUST_DETAIL_COUNT	The maximum customer detail records to be saved in reports periodically.	

Field Name	Description (Caption)	Default Value
COMMIT_BLOCK	Commit Block Size Optional field. The value of this parameter should be a positive integer. If not entered, it is defaulted to 1000.	1000
SELECT_BLOCK	Select Block Size Optional field. It is the number of records fetched at a time. The value of this parameter should be a positive integer. If not entered, it is defaulted to 1000.	1000
CLIENT_NM	Client name for Report	

Custom Parameters

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
PARAM_ID	Parameter ID	Primary Key	0	Enter Parameter ID
RUN_MODE	Run Mode	Required	Report Only	Select the value from the CVL_RUN_MODE.
DEPT_CD	Department	Required	Blank	Enter the Department Code, or "ALL".
RUN_DT	Run Date	Required	Blank	Enter the date for which the process is to be run.
DT_FREQ	Date Frequency	Optional	No	Set the value to 'Yes' if Date Frequency is to be selected.
FROM_DT	Date from	Optional	Blank	Enter the field if the frequency required is date range. Records with this frequency for the run date will be selected for processing.
TO_DT	Date to	Optional	Blank	Required if the 'Date from' field is entered.
DATA_SR	Data Source	Optional	Both	Specify the Data Source.

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
C				
CUST_SE L_TYP	Customer Selection	Optional	Not Applicable	Specify how Customer code should be used for selection.
BFY	Budget Fiscal Year	Optional	Blank	Enter a valid Budget Fiscal Year
FY	Fiscal Year	Optional	Blank	Enter a valid Fiscal Year
APD	Accounting Period	Optional	Blank	Enter an Accounting Period
INC_PPD	Include Prior Periods	Required	Not Applicable	Select the value from the CVL_INC_PPD
CUST_ID	Vendor/Customer	Optional	None	At least one Customer is required if Customer Selection is "Include" or "Exclude".

Selection Criteria

Run Mode	Inputs
Report Only	<p>When the Data Source is "Both" or "Reimbursement Holding", select all Reimbursement Holding Table (R_REIM_HOLD) records where:</p> <ul style="list-style-type: none"> • If the Department Code is not "All", the Department Code equals the input Department Code; • If the Customer Selection is "Include", the Customer Code matches one of those listed; • If the Customer Selection is "Exclude", the Customer Code does not match any of those listed; • The Frequency(s) applicable between the Last Run Date and the Input Run Date from the Frequency Date Table (R_REIM_FREQ_DT). <p>When the Data Source is "Both" or "Ready for Reimbursement", select all Ready for Reimbursement Table (CA_REIM) records where:</p> <ul style="list-style-type: none"> • If the Department Code is not "All", the Department Code equals the input Department Code; • If the Customer Selection is "Include", the Customer Code matches one of those listed; • If the Customer Selection is "Exclude", the Customer Code does not match any of those listed; • The Frequency(s) applicable between the Last Run Date and the Input Run Date from the Frequency Date Table (R_REIM_FREQ_DT). <p>The Input Parameters are as per the record on the custom parameter table (R_REIM_GEN_PARM).</p>

Run Mode	Inputs
Update	Select all records from temporary Reimbursement Request Table (R_REIM_REQ_TMP). Select all records from temporary Reimbursement Holding Table (R_REIM_HOLD_TMP).
Report and Update	Same as Report Only and Update modes

In addition to the above selection criteria, the value in the field “Include Prior Period” will also be considered by selecting the records as explained below.

Include Prior Period	Inputs
Not Applicable	Select this option if you have not entered an Accounting Period on this parameter page.
Yes	If the value is ‘Yes’ then the Accounting Period is required. The process will select records with the Accounting Period lesser than or equal to the Accounting Period entered on this parameter page.
No	If the value is ‘No’ then Accounting Period is required. The process will select the records with an Accounting Period equal to the Accounting Period entered on this parameter page.

Sort Criteria

Records selected from the Ready for Reimbursement Table and Reimbursement Holding Table are sorted in the following order (this is only done when run mode equals Report Only or Report and Update):

- Department
- Major Program
- Funding Profile
- Funding Priority
- Funding Line
- Posting Amount

The selected records will be sorted so that all the records having similar Cost Accounting COA are grouped together. In addition the records are sorted in the ascending order of posting amount so that all negative records are processed first and the billing for Non-CMIA records which have Maximum Reimbursable Amount consider all negative posting amount records before the positive posting amount records.

Problem Resolution

If the process fails due to any reason perform the following corrective actions:

- Check the log file for any errors

If the system administrator is not successful after completing the above corrective actions a database restore is required. Restore to the last backup and re-run the complete Reimbursement Process beginning with Selection and Calculation.

Please note it is recommended a backup be taken of the financial database prior to executing any of the Reimbursement components if space and time are not issues.

2.1.24 Reimbursement Output Process

Description

The Cost Accounting reimbursement functionality is divided into the following three major processes:

1. Reimbursement Selection and Calculation
2. Reimbursement Generation
3. Reimbursement Output

These three processes are inter-related and must be executed in a sequential order to satisfy the requirement of cost reimbursement programs administered by governmental entities. This particular run sheet is describing the system administration aspect of the Reimbursement Output process. (Please refer to the appropriate run sheet for detail descriptions related to the Reimbursement Selection and Calculation and Reimbursement Generation processes.)

The Reimbursement Output process performs the following key functions:

- Generation, loading, and submitting of CGI Advantage Financial transactions to record the accounting events of Cost Accounting Reimbursement process, and
- Generation of electronic billing files for external customers: Generation of the Federal Highway Administration (FHWA) electronic file for States payment requests of the Federal share of highway related programs.

Generating Electronic Billing Files: CGI Advantage Financial recognizes that most reimbursement is performed through electronic data exchange. In the delivered model, CGI Advantage Financial provides only one pre-defined format and it is for FHWA.

Generating, Loading, Submitting CGI Advantage Financial Transactions: When a user establishes their Major Programs, Programs, Funding Profiles, Funding Priorities, Funding Lines, and Task Order, they are required to select a Reimbursement Output Type by Task Order Billing Setup and Customer ID. Depending on Task Order Billing Setup and the type of customer (that is, external, internal, or other), the system provides the following Reimbursement Output Types: Generate Receivables Only, Generate Receivables and Cash Receipts, Internal Sale – Intra-Fund, Internal Sale – Inter-Fund, Internal Rev Recog, and None. The Reimbursement Output Process selects records from the Reimbursement Request (R_REIM_REQ) table based on custom parameters (for example, Department, Customer ID, Major Program, and Program Period Code).

The chart of account elements for the accounting lines of the Receivables (RE) and Cash Receipts (CR) are obtained from the Revenue Funding Line (R_REV_FLINE) table. The Internal Exchange Transaction (IET) vendor line and accounting lines chart of account elements are obtained from the Revenue Funding Line (R_REV_FLINE) table and the Buyer Funding Line (R_BUYR_FLINE) table respectively when Task Order Billing is not used, or from the Task Order (R_TASK_ORD) and the Task Order Buyer Line (R_TASK_ORD_BUYR_LN) table respectively if the Reimbursement Request record has a Task Order billing major program. Records are written to an exception report if the following conditions exist:

If there are no records in the Buyer Funding Line (R_BUYR_FLINE) table for the Internal Customer in the Reimbursement Request (R_REIM_REQ) table, and

If the Buyer percentages of the records on the Buyer Funding Line (R_BUYR_FLINE) table are less than 100 percent for a given funding line record.

If the Reimbursement Request record has a Task Order billing major program and the Task Order is not found on the Task Order (R_TASK_ORD) table.

If the Task Order uses Reimbursement Output Type of Internal Sale or Internal Reimbursement, and there are no records in the Task Order Buyer Line (R_TASK_ORD_BUYR_LN) or the Buyer percentages do not sum to 100 percent for the Task Order.

In addition, the records that have been successfully generated into CGI Advantage Financial transactions are copied to the Reimbursement History (R_REIM_HIST) table.

The Receivables (RE) and Internal Exchange Transactions (IET) are loaded and submitted through SysManUtil. The Cash Receipts (CR) are loaded into the Transaction Catalog using SysManUtil, but not submitted. The Receivables (RE) and Internal Exchange Transactions (IET) that fail during transaction submission are written to clean up reports.

Generation of the FHWA Electronic File: The FHWA Electronic file is populated with cost accounting data for those Customers that have been identified as FHWA Customers on the Customer Account Options table. All the selected records in the current run and corrected records (that is, the records rejected by FHWA (FMIS) from any previous Reimbursement cycle) are summarized based on FHWA attributes and copied to the current (new) FHWA Electronic file.

Technical Flow

The Reimbursement Output process consists of multiple steps. Each step described here may consist of one or more individual offline jobs. All of those individual jobs are chained together to complete the entire Reimbursement Output process. The following steps are involved in the process:

1. Reimbursement Output
2. Generate Transactions
3. Clean Up Processes
4. FHWA File Generation

Step #1 - Reimbursement Output

Running the 1st job in the chain, the Reimbursement Output job, completes this step.

Selection of the Records

- Select the records from the Reimbursement Request table depending on the custom parameters entered by the user and sorted by the Customer, Department, Major Program, Funding Profile, Funding Priority, and Funding Line fields.

For records with Output type 'Generate Receivables Only' or 'Generate Receivables and Cash Receipts' Drawdown Grouping will always occur. The records are sorted by Drawdown Group, Drawdown Department, Customer, Billing Profile, AR Department, AR Unit, Output Transaction Department, Output Bank Account, Reimbursement Output Format, Major Program and Program.

For records with Output type 'Internal Sale - Intra-Fund' or 'Internal Rev Recog' or 'Internal Sale - Inter-Fund' the records are sorted by Customer, Vendor, Reimbursement Output Format, Department and Major Program.

- Get the COA elements for these records from the Revenue Funding Line table.

- Copy these records from the Reimbursement Request table to the Reimbursement History Temporary table. The field Reimbursement Output Type is copied to the Reimbursement History Temporary table from the Funding Line table or the Task Order table.
- Based on the Reimbursement Output Type, it is decided which transactions in XML format are to be created. The valid values are as follows:
- If the value of the Reimbursement Output Type is equal to 1, Receivables are to be created.
- If the value of the Reimbursement Output Type is equal to 2, Receivables and Cash Receipts are to be created.
- If the value of the Reimbursement Output Type is equal to 3, 100, or 101, Internal transactions are to be created.
- If the value of the Reimbursement Output Type is equal to 4, no transactions are to be created.

Create Accounting Lines for the Transactions

- The Accounting Lines are generated for the transactions based on two modes (that is, Detail and Summarized). The Accounting line limit can be set at the custom parameter or Transaction Component Requirements table or the System Wide Options table. The most restrictive limit in the three locations is considered as the maximum line limit for accounting lines. For CR and IET transactions, if a Vendor Line limit is set on the Transaction Component Requirements table, transaction creation will also break on this transaction component.
- In the Detail mode, one accounting line is created for one record in the Temporary Reimbursement History (R_REIM_HIST_TEMP) table. In the summarized mode, the records are summarized based on the COA elements and one AL is created for each summarized record and the posting amount is set to the consolidated posting amount of all the summarized records.
- Accounting lines with negative Posting Amounts are allowed in RE and IET Transactions. If the Total Amount of the RE Transaction is negative, then the accounting line that resulted in the Total Amount becoming negative is included on the Exception Report. This condition is checked for in both the modes (that is, Detail and Summarized).
- When creating the XML files for Receivables (RE) and Cash Receipts (CR):
- If no drawdown grouping is performed, one RE/CR transaction is created for each unique combination of Customer Account, Output Transaction Department, Output Bank Account, Reimbursement Output Type, and Major Program.
- If drawdown grouping is performed, one RE transaction is created for each unique combination of Drawdown Group/Drawdown Department/Customer Account and one CR transaction is created for each unique combination of Drawdown Group/Drawdown Department/Output Bank Account.
- A new RE transaction will be created when the maximum line limit is reached. A new CR transaction is created when either the Vendor line limit or the Accounting line limit is reached.
- The provider Customer code for Internal Customers is retrieved from the Major Program (R_MJR_PROG) table.
- While creating Internal Exchange Transactions (IET), one accounting line is created for each Buyer Funding Line record.
- One IET transaction is created for each unique combination of Customer ID, Vendor Customer Code, Reimbursement Output Type, Department, and Major Program. If the

Major Program uses Task Order Billing, an IET transaction is also created when Task Order changes value, in addition to the fields already mentioned.

- A new IET transaction is created when either the Vendor Line limit or the Accounting Line limit is reached. If an Accounting Line limit is reached but all lines cannot be created for a Funding Priority, the program will place all lines on a single transaction. If the line limit is still exceeded, the transaction will not load. Thus any line limit for IET transactions must be greater than or equal to the largest number of 'Internal' Funding Lines for a Funding Priority when Task Order billing is not used or the largest number of Task Order Buyer Lines for a Task Order when Task Order billing is used.
- The newly created Transaction Reference Information (that is, Transaction ID, Vendor Line No., and the Accounting Line No. are stored against each record in the Temporary Reimbursement History (R_REIM_HIST_TEMP) table.

Create Exception Report

- If there are no Buyer Line records in the Buyer Funding Line table for the Internal Customer, details are logged to the Exception Report.
- If the Buyer percentages on the Buyer Line records obtained from the Buyer Funding Line table for any given funding line do not sum to 100%, the details are logged to the Exception Report.
- 1. If the Major Program uses Task Order billing and the Task Order does not exist on the Task Order table, details are logged to the Exception Report.
- 2. If the Task Order uses Reimbursement Output Type of Internal Sale or Internal Reimbursement, and there are no Task Order Buyer Line records in the Task Order Buyer Line table or the Buyer percentages do not sum to 100%, details are logged to the Exception Report.
- If the Total Amount of the RE Transaction becomes negative, then the accounting line due to which the Total Amount became negative is entered to the Exception Report.
- Delete the records that have been included in the exception report from the Temporary Reimbursement History (R_REIM_HIST_TEMP) table.

Create XML's for the Transactions

- One XML file is created for each Transaction type, that is, Receivables, Cash Receipts and Internal Exchange Transactions respectively.
- Receivables contain only one Vendor Line per transaction, whereas Cash Receipts and Internal transactions can have multiple Vendor lines per transaction.

Create Parameter(s) to be passed to FHWA Step

- Once all of the processing is successful a file named ReimOutputParms.txt (assuming the user does not change this file name) will be generated.
- The file will contain the Chain Level Job's ID. It is a way of communicating information to the FHWA Step of the chain.
- It allows flexibility in being able to disable the FHWA step in the chain and then run it at a later point, so long as the ReimOutputParms.txt file has not been changed by another run. The FHWA will use the parameter information from the Chain Level job to perform its work.

Step #2 - Generate Transactions

This step consists of five individual SysManUtil jobs in the Output chain: the loading of RE, CR and IET transactions followed by the submission of RE and IET transactions.

- Load XML file containing the RE transactions to the Receivable transaction data objects (RE_DOC_HDR, RE_DOC_VEND and RE_DOC_ACTG).
- Submit the RE transactions residing in the Transaction Catalog. Please note that SysManUtil generates a flat file for the rejected/failed RE transactions. The flat file contains the failed accounting lines along with the error messages for those lines.
- Load XML file containing the CR transactions to the Cash Receipts transaction data objects (CR_DOC_HDR, CR_DOC_VEND and CR_DOC_ACTG). The generated CR Transactions are not to be submitted offline as part of Reimbursement Output
- Load XML file containing the IET transactions to the Internal Transaction data objects (IET_DOC_HDR, IET_DOC_VEND and IET_DOC_ACTG).
- Submit the IET transactions residing in the Transaction Catalog. Please note that SysManUtil generates a flat file for the rejected/failed IET transactions. The flat file contains the failed accounting lines along with the error messages for those lines.

Step #3 – Clean-Up Processes

This step consists of two individual offline jobs: RE Clean Up and IET Clean Up

- Generate Clean-Up Report for rejected Receivable transactions that lists the rejected Transactions along with errors.
- Generate Clean-Up Report for rejected Internal Exchange Transactions that lists the rejected Transactions along with errors.

Step #4 – FHWA File Generation

This step consists of one individual job, namely FHWA File Generation.

Main Processing

- Delete all records written to the Detail FHWA table (R_DET_FHWA_REC) in the previous run.
- Get all the records from the Temporary Reimbursement History (R_REIM_HIST_TEMP) table sorted by Customer ID and Billing Profile.
- If the Electronic File Type flag is set to “FHWA” on the Customer Accounts Option table for the Customer and Billing Profile combination, then a FHWA file should be generated for that customer.
- Flat file prefix is obtained for these customers from the Customer Account Options table.
- Records for those flat file prefix values for which the FHWA file is to be created in the current run are deleted from the summarized FHWA (R_FHWA_REC) table.
- The records for the current run that are in the Reimbursement History (R_REIM_HIST) table are summarized and copied to the Detail FHWA Records (R_DET_FHWA_REC) table. The records are summarized based on the following fields: Customer, Billing Profile, Department, Major Program, Funding Profile, Funding Priority, and Funding Line. Demo ID is also copied to the Detail FHWA Records (R_DET_FHWA_REC) table from the R_REIM_HIST table.

- The following fields are obtained from the custom parameters entered by the user: Object Class, Recipient ID, Payment Date, and Federal Activity.
- Appropriation field value is obtained from the Program Period Code (R_PPC) table or Funding Line (R_FLINE) table as specified on the Major Program (R_MJR_PROG) table of the record.
- Project-Agreement field value is obtained from the Project-Agreement field on the Funding Line (R_FLINE), Phase-Program (R_PHASE_PROG), Program (R_PROG), or Major Program (R_MJR_PROG) tables in that order.
- The records from the Detail FHWA (R_DET_FHWA_REC) table are summarized by Customer, Appropriation, Activity, Recipient ID, Project-Agreement, Demo ID, and Object-Class. The summarized records are then copied to the FHWA Records (R_FHWA_REC) table.
- Transaction Type is set to 10 when the calculated Detail Amount is positive; otherwise Transaction Type is set to 20 on the FHWA Records (R_FHWA_REC) table.
- FHWA flat file is generated from the records in the FHWA Records (R_FHWA_REC) table. The Job ID is appended to the file prefix obtained earlier. The file prefix is stored in the summarized FHWA (R_FHWA_REC) table. This value is to be used by the FHWA Clean Up Process. Please note the FHWA Clean Up Process is discussed in its separate run sheet; see the [FHWA Clean Up](#) run sheet for additional details.

Deletion and Transfer of Records

- Delete successfully processed records (all records remaining in the Temporary Reimbursement History (R_REIM_HIST_TEMP) table) from the Reimbursement Request (R_REIM_REQ) table.
- Copy records from the Temporary Reimbursement History (R_REIM_HIST_TEMP) table to the Reimbursement History (R_REIM-HIST) table.
- The Temporary Reimbursement History (R_REIM_HIST_TEMP) table is emptied during the next run.

Create Parameter(s) to be passed to Auto Revenue Recognition Chain

- Once all of the processing is successful a file named AutoRevrec.txt (assuming the user does not change this file name) will be generated.
- The file will contain the Chain Level Job's ID. It is a way of communicating information to the Auto Revenue Recognition Chain. It allows the Auto Revenue Recognition Chain to process Reimbursement History records created by the last Reimbursement Output run.

When to Run

User's Discretion – Recommended that the Output process is executed on a timely basis (that is, daily or weekly). The billing transaction volume should be kept at a reasonable level.

Run Sequence

1. Reimbursement Output Process
2. SysManUtil (for loading of Receivables)
3. SysManUtil (for loading of Cash Receipts)

4. SysManUtil (for loading of Internal Exchange Transactions)
5. SysManUtil (for submitting of Receivables)
6. SysManUtil (for submitting of Internal Exchange Transactions)
7. Clean Up Process (for failed Receivables and Internal Exchange Transactions)
8. FHWA File Generation

Inputs

Process	Input
Reimbursement Output Process	Custom Parameter (R_OTPT_PARM_DET) Reimbursement Request Table (R_REIM_REQ) Revenue Funding Line Table (R_REV_FLINE) Funding Line Table (R_FLINE) Buyer Funding Line Table (R_BUYR_FLINE) Task Order (R_TASK_ORD) Task Order Buyer Line (R_TASK_ORD_BUYR_LN)
SysManUtil (for loading of Receivables)	RE XML
SysManUtil (for loading of Cash Receipts)	CR XML
SysManUtil (for loading of Internal transactions)	IET XML
SysManUtil (for submitting of Receivables)	RE Transaction Header (RE_DOC_HDR) RE Transaction Vendor (RE_DOC_VEND) RE Transaction Accounting (RE_DOC_ACTG) Transaction Header (DOC_HDR) Transaction Vendor (DOC_VEND) Transaction Accounting (DOC_ACTG)
SysManUtil (for submitting of Internal transactions)	IET Transaction Header (IET_DOC_HDR) IET Transaction Vendor (IET_DOC_VEND) IET Transaction Accounting (IET_DOC_ACTG) Transaction Header (DOC_HDR) Transaction Vendor (DOC_VEND) Transaction Accounting (DOC_ACTG)
Clean Up Process (for failed Receivables and Internal Exchange Transactions)	Flat file containing the rejected RE transactions Flat file containing the rejected IET transactions
FHWA File Generation	Reimbursement History Table (R_REIM_HIST) Major Program Table (R_MJR_PROG) Funding Line Table (R_FLINE) Program Period Code Table (R_PPC) Program Table (R_PROG) Phase Program Table (R_PHASE_PROG)

Process	Input
	Customers Account Options Table (R_CUST_ACCT_OPT) ReimOutputParms.txt (or whatever valid file name the user enters)

Outputs

Process	Output
Reimbursement Output Process	Temporary Reimbursement History Table (R_REIM_HIST_TMP) Reimbursement Output Exception Report RE XML CR XML IET XML ReimOutputParms.txt (or whatever valid file name the user enters)
SysManUtil (for loading of Receivables)	RE Transaction Header (RE_DOC_HDR) RE Transaction Vendor (RE_DOC_VEND) RE Transaction Accounting (RE_DOC_ACTG) Transaction Header (DOC_HDR) Transaction Vendor (DOC_VEND) Transaction Accounting (DOC_ACTG) Posting Line Catalog (PSTNG_LN_CAT)
SysManUtil (for loading of Cash Receipts)	CR Transaction Header (CR_DOC_HDR) CR Transaction Vendor (CR_DOC_VEND) CR Transaction Accounting (CR_DOC_ACTG) Transaction Header (DOC_HDR) Transaction Vendor (DOC_VEND) Transaction Accounting (DOC_ACTG) Posting Line Catalog (PSTNG_LN_CAT)
SysManUtil (for loading of Internal transactions)	IET Transaction Header (IET_DOC_HDR) IET Transaction Vendor (IET_DOC_VEND) IET Transaction Accounting (IET_DOC_ACTG) Transaction Header (DOC_HDR) Transaction Vendor (DOC_VEND) Transaction Accounting (DOC_ACTG) Posting Line Catalog (PSTNG_LN_CAT)

Process	Output
SysManUtil (for submitting of Receivables)	Flat file containing the rejected RE transactions
SysManUtil (for submitting of Internal transactions)	Flat file containing the rejected IET transactions
Clean Up Process (for failed Receivables and Internal Exchange Transactions)	RE Transaction Exception Report IET Transaction Exception Report
FHWA File Generation	FHWA electronic file Reimbursement Request Table (R_REIM_REQ) Reimbursement History Table (R_REIM_HIST) FHWA Detail Records Table (R_DET_FHWA_REC) FHWA Records Table (R_FHWA_REC) Auto Revenue Recognition parameter file (AutoRevrec.txt)

Reimbursement Output Process

- Reimbursement History table (R_REIM_HIST) – Holds the records that are successfully processed.
- Reimbursement Output Exception Report – Contains the records from Reimbursement Request table that failed while creating the IET or RE XMLs.
- Receivable XML – Receivables XML file containing Header, Vendor and Accounting Line Tags.
- Cash Receipt XML – Cash Receipts XML file containing Header, Vendor and Accounting Line Tags.
- Internal Exchange Transaction XML – Internal Exchange Transaction XML containing Header, Vendor and Accounting Line Tags.

SysManUtil (for Loading Receivables)

- Creates the required transactions as per the XML file

SysManUtil (for Loading Cash Receivables)

- Creates the required transactions as per the XML file

SysManUtil (for Loading Internal Exchange Transactions)

- Creates the required transactions as per the XML file

SysManUtil (for Submitting Receivables)

- Validates and submits the loaded transactions
- Creates a flat file containing failed transaction information

SysManUtil (for Submitting Internal Exchange Transactions)

- Validates and submits the loaded transactions
- Creates a flat file containing failed transaction information

Clean-Up Process (for Failed Receivables and Internal Exchange Transactions)

- RE Transaction Exception Report – The failed records in the SysManUtil (based on the Flat file generated in the Submit Process) process
- IET Transaction Exception Report – The failed records in the SysManUtil (based on the Flat file generated in the Submit Process) process

FHWA File Generation

- FHWA Electronic file is generated as per the guidelines provided by FHWA.
- Reimbursement History (R_REIM_HIST) table – Holds the records that were successfully processed.
- Reimbursement Request (R_REIM_REQ) table – Records are deleted from this table that were successfully processed.
- FHWA Detail Records (R_DET_FHWA_REC) table
- FHWA Records (R_FHWA_REC) table

Job Parameters

Reimbursement Output

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
AMSEXPOR T	Export Location at Reimbursement Output Job	Required	-	** Refer to Note: Assumptions for SWBP on page no. 5
AMSPARM	Parameter Location at Reimbursement Output Job	Required	-	** Refer to Note: Assumptions for SWBP on page no. 5
AMSLOGS	Logs Location at Reimbursement Output Job	Required	-	** Refer to Note: Assumptions for SWBP on page no. 5

CLIENT_NM	Client name for Report	Optional	-	
COMMIT_BLOCK	Commit block Size	Optional	-	Specifies the Commit Block size. Valid value is any positive integer number. Process will commit the transaction after the number of the records specified in this parameter is processed.
JOB_PARAM_OUT_FL	Reimbursable Output Job Parameter File: Pushed Out (communication mechanism between this step and the FHWA Step, if inside of different chain job runs due to failure or conscious stop)	Required	ReimOutputParms.txt	
MAX_ACTG_LN_PER_DOC	Maximum Accounting Lines per generated Receivable and Cash Receipt Transaction			
PARAM_ID	Parameter Id	Required	0	
SELECT_BLOCK	Select Block Size		1000	It is the number of Reimbursement Request records fetched at a time. If not entered, it is defaulted to 1000. Can be used for Performance tuning.

Load RE Transaction

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$AMSEXPORTReimGenRE Docs.xml	

ACTN_CD	Action to be Performed - 171- Export; 162 - Import	Required	171	
BYPS_ADNT_FL	Bypass Auto Transaction Numbering	Required	true	
COMMIT_BLOCK	Commit Block Size	Optional	10	

Load CR Transaction

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$AMSEXPORT\$\$/ReimGenCR Docs.xml	
ACTN_CD	Action to be Performed - 171 - export; 162 - import	Required	171	
BYPS_ADNT_FL	Bypass Auto Transaction Numbering	Required	true	
COMMIT_BLOCK	Commit Block Size	Optional	10	

Load IET Transaction

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$AMSEXPORT\$\$/ReimGenIET Docs.xml	
ACTN_CD	Action to be Performed - 171 - export; 162 - import	Required	171	

BYPAS_ADNT_FL	Bypass Auto Transaction Numbering	Required	true	
COMMIT_BLOCK	Commit Block Size	Optional	10	

Submit RE Transaction

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
EXCEP_REP_FILE_NM	The file name to which the exception report will be written out to	Protected	\$\$AMSLOGS\$\$/ReExp.txt	
PARAM_FILE	Parameter File	Protected	\$\$AMSPARM\$\$/RMRE_PARAM.txt	
ACTN_CD	Action to be Performed - 171 - export; 162 - import	Required	162	
DOC_STAT_CD	Status of the transactions that can be submitted	Optional	1	
DOC_TYP	Type of the transactions that can be submitted	Required	RE	
EXCEP_REP_FILE_NM	The file name to which the exception report will be written out to	Required	\$\$AMSLOGS\$\$/ReExp.txt	

Submit IET Transaction

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
EXCEP_REP_FILE_NM	The file name to which the exception report will be written out to	Protected	\$\$AMSLOGS\$/letExp.txt	
PARAM_FILE	Parameter File	Protected	\$\$AMSPARM\$/RMLET_PARM.txt	
ACTN_CD	Action to be Performed - 171 - export; 162 - import	Required	162	
DOC_STA_CD	Status of the transactions that can be submitted	Optional	1	
DOC_TYP	Type of the transactions that can be submitted	Required	RE	
EXCEP_REP_FILE_NM	The file name to which the exception report will be written out to	Required	\$\$AMSLOGS\$/ReExp.txt	

RE / IET CleanUp

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
AMSLOGS	Logs Location at RE / IET CleanUp Job	Required		** Refer to Note: Assumptions for SWBP on page no. 5
CLIENT_NM	Client name for Report	Optional		
IET_EXCEP_REP_FILE_NM	IET Exception Report File Name	Required	letExp.txt	
RE_EXCEP_REP_FILE_NM	RE Exception Report File Name	Required	ReExp.txt	

FHWA File Generation

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
AMSEXPOR T	Export Location at FHWA File Generation Job	Required		** Refer to Note: Assumptions for SWBP on page no. 5
AMSPARM	IMPORT Location for Param File	Optional	\$\$AMSRO OT\$\$/Par ms	** Refer to Note: Assumptions for SWBP on page no. 5
CHN_LVL_J OB_ID	Chain Level Job ID to be used for acquiring Chain Level Params	Required		The user must supply either a valid Chain Level Job ID or a valid Param Input file name.
JOB_PARM_ IN_FL	Reimbursable Output Job Parameter File: Pulled In	Required	ReimOutput tParams.txt	The user must supply either a valid Chain Level Job ID or a valid Param Input file name.
AUTO_REV_ REC_PARM_ _FILE	Automatic Revenue Recognition Parameter File (.txt)	Required	AutoRevre c.txt	This parameter file stores the Chain Job ID that was assigned to the current run of the Reimbursement Output chain and will be used as input to the Auto Revenue Recognition chain.

Data Object Name R_OTPT_PARM_DET

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
PARAM_ID	Parameter ID	Required		Unique Identifier for each Parameter record
DEPT_CD	Department Code	Required		
CUST_ID	Customer ID	Optional		
MJR_PROG_CD	Major Program Code	Optional		For Drawdown Grouping Major, Program should be Null.
PPC_CD	Program Period Code	Optional		For Drawdown Grouping, the value should be Null. Required if Major

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
				Program is entered
SUMM_TYP	Summarization Type	Required	Detail	CVL: Options are 'Detail' or 'Summarize'
RE_DOC_CD	RE Transaction Code	Required		
RE_DOC_DEPT_CD	RE Transaction Department Code	Required		
RE_DOC_UNIT_CD	RE Transaction Unit Code	Optional		
RE_PFX	RE Prefix	Required		
RE_EVNT_TYP_ID	RE Event Type ID	Required	AR01	
CR_DOC_CD	CR Transaction Code	Required		
CR_EVNT_TYP_ID	CR Event Type ID	Required	AR02	
IET_DOC_CD	IET Transaction Code	Required		
IET_DOC_DEPT_CD	IET Transaction Department Code	Required		
IET_DOC_UNIT_CD	IET Transaction Unit Code	Optional		
IET_PFX	IET Prefix	Required		
IET_EVNT_TYP_ID	Internal Sale Event Type	Required	IN05	
DOC_BFY	Transaction Budget Fiscal Year	Optional		
DOC_PER	Transaction Period	Optional		
FHWA_OBJ_CLS	FHWA Object Class	Optional		
FHWA_PYMT_DT	FHWA Payment	Optional		

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
	Date			
RCPT_ID	Recipient ID	Optional		
FHWA_ACTV	Federal Activity	Optional		

Custom Parameters

If you are running the job in the Financial application, you can click on the **Setup Custom Parameters** link from the Reimbursement Output job step and select a parameter record. If you are running the job in the Administration application, enter a valid Parameter ID from the Custom Parameter table from the Financial application.

The Reimbursement Output Parameters page can be divided into five sections from a user’s perspective.

General Options section

- Parameter ID: This is the unique identifier that is passed to the Reimbursement Output Process.
- Department Code: A department code is required.
- Customer Code: The user may optionally choose to select all the records for a given customer within a department.
- Major Program/Program Period: The user may also choose to select all the records for a Major Program/Program Period combination within a given department. The user may also select records for a given Department/Customer/Major Program/Program Period combination.
- Summarize Type: The user has the option of creating one output accounting line for all records that have the same set of COA elements. Otherwise the user can choose the detailed option which results in one accounting line being created for each record on the Reimbursement Request table.

Output-RE section

- Transaction Code
- Transaction Department
- The Transaction Dept field in the Output-RE section on the Reimbursement Output Parameters (REIMOTPT) table cannot be ALL if the Transaction Unit Code Required flag is selected on the Transaction Control (DCTRL) table for the specified Transaction Code in the Output-RE section on REIMOTPT.
- The Transaction Dept field in the Output-RE section on the Reimbursement Output Parameters (REIMOTPT) table cannot be ALL if the Transaction Unit Code Required flag is selected on the Transaction Control (DCTRL) table for the specified Transaction Code in the Output-CR section on REIMOTPT.
- Note: If a site uses Drawdown Groups, they should run Reimbursement Output with ALL in the Transaction Dept field under the Output-RE section on the REIMOTPT table.

- Transaction Unit: Department/Unit combination is validated against the Unit table; cannot be populated if the Transaction Department is “ALL”.
- Prefix: Fiscal Year/Transaction Code/Transaction Department/Prefix combination validated against the Automatic Numbering table. If the Output-RE’s Transaction Department=ALL, the system uses a wildcard value (‘****’) when performing the validation.
- Event Type: The Transaction Code/Event Type combination is validated against the Allowable Event Types for Transaction Code table

Output-CR section

- Transaction Code
- Event Type: The Transaction Code/Event Type combination is validated against the Allowable Event Types for Transaction Code table

NOTE: CR Transaction Department, Transaction Unit and Prefix are the same as for the RE transaction. Hence there is no need for these three fields to be duplicated in the Output-CR section of the Parameter page.

Output-IET section

- Transaction Code
- Transaction Department:
- The Transaction Dept field in the Output-IET section on the Reimbursement Output Parameters (REIMOTPT) table cannot be ALL if the Transaction Unit Code Required flag is selected on the Transaction Control (DCTRL) table for the specified Transaction Code in the Output-IET section on REIMOTPT.
- Note: Drawdown grouping logic is not applicable to the IET Transaction Type because IET’s have their own “grouping” logic involving unique combinations of Customer ID (Funding Line Customer), Vendor Customer Code (Internal Vendor code from the Major Program page), and other data elements.
- Transaction Unit: Department/Unit combination is validated against the Unit table; cannot be populated if the Transaction Department is ALL.
- Prefix: Fiscal Year/Transaction Code/Transaction Department/Prefix combination validated against the Automatic Numbering table. If the Output-IET’s Transaction Department=ALL, the system uses a wildcard value (‘****’) when performing the validation.
- Event Type: The Transaction Code/Event Type combination is validated against the Allowable Event Types for Transaction Code table.

Output-FHWA section (All Fields are Optional)

- Object Class
- Payment Date
- Recipient ID
- Federal Activity

NOTE: If the user wishes to submit an electronic bill file to the FHWA, the following fields are required in flat file submission and must be provided by the user.

Job Parameters Validation

- JOB_PARM_OUT_FL is validated to ensure that the folder it is being sent to exists and that the file name supplied has an extension. If the file exists it will be cleared out at the beginning of the Reimbursement Output step. It will be seeded at the end if successful. The user has the flexibility of changing this file name, please ensure that this value and the FHWA step value stay in lock step syntax.
- JOB_PARM_IN_FL has minor validations. It again will ensure the file exists. If the file does not an error will be thrown. The value inside of this file will be validated in a manner similar to the validation performed for CHN_LVL_JOB_ID. This value can be changed in the baseline version of the job by the person running the job. So, please ensure that this file name maps perfectly with the associated run from which the user desires the Chain Level Job ID to be used.
- CHN_LVL_JOB_ID this will be checked to ensure it is a valid positive, integer number. If it is not then an error will be thrown specifying that it must be valid. If the value is read in from the in file, it will be validated for the same items. Since this value will usually only be entered in by the user in unusual circumstances, this value will take precedence over the file. So, if the user supplies an invalid value here, while the file contains a valid value, we will never get to the file because the supplied value here takes precedence.

Custom Parameters Validation

- When the Customer is entered, it is validated on the Vendor Customer table.
- When Department is entered, it is validated on the Department table.
- When Major Program is entered, it is validated on the Major Program table.
- When a Program Period Code is entered, it is validated on the Program Period table.
- The Transaction Codes of the Receivables, Cash Receipts, and Internal transactions are validated on the Transaction Generation Control table.
- Output Transaction Prefix is validated against the Auto Transaction Number table.
- Output Department and Unit fields are validated against the Department and Unit tables, respectively.
- When a Major Program is entered, the Program Period Code must be entered. Program Period Code must be valid for the Major Program and Department combination.
- Event types are validated against the Allowable Event Types for Transaction Code table.
- If FHWA payment date is not entered the current system date is defaulted.
- Budget Fiscal Year and Accounting Period are validated and should not be closed.

Sort Criteria

Reimbursement Output Process

- The selected records from the Reimbursement Request table are sorted on the Customer ID, Department Code, Major Program Code, Funding Profile Code, Funding Priority Code and Funding Line Code fields.
- For records with Output type 'Generate Receivables Only' or 'Generate Receivables and Cash Receipts' Drawdown Grouping is done. For Drawdown Grouping, the selected records from the Reimbursement Request table are sorted on the Drawdown Group, Drawdown Department, Customer ID, Billing Profile, AR Department, AR Unit, Output Transaction Department, Output Bank Account, Reimbursement Output Format, Major Program Code and Program Code.
- For records with Output type 'Internal Sale - Intra-Fund' or 'Internal Rev Recog' or 'Internal Sale - Inter-Fund' the records are sorted by Customer, Vendor, Reimbursement Output Format, Department and Major Program.

FHWA File Generation Process

- The selected records from the Reimbursement History table are sorted on the fields Customer ID, Billing Profile Code, Department Code, Major Program Code, Funding Profile Code, Funding Priority Code and Funding Line Code.
- All the records from the FHWA Records table are sorted on the field Customer ID.

Selection Criteria

Reimbursement Output Process

The records are selected from the Reimbursement Request (R_REIM_REQ) table based on the Custom Parameters entered by the user. For example, if a user enters Customer ID and Department in the Custom Parameter table records are selected based on Customer ID and Department. If a user enters a combination of Department, Major Program, and Program Period Code records are selected based on these key attributes.

FHWA File Generation Process

The records are selected from the Reimbursement History (R_REIM_HIST) table only for those Customer/Billing Profile combinations for whom the attribute Electronic File Type is set to 'FHWA' on the Customer Account Options table.

Problem Resolution

If the process fails due to any reason perform the following corrective actions:

- Check whether the XML files were created in the directory/Data Files/Export-Import.
- Check the log file for any errors.
- Check whether the corresponding records exist on the Funding Line (F_LINE) table.

If the system administrator is not successful after completing the above corrective actions a database restore is required. Restore to the last backup and re-run the complete Reimbursement Process beginning with Selection and Calculation. Please note it is recommended a backup be taken of the financial database prior to executing any of the Reimbursement components if space and time are not issues. This batch program produces new Advantage transactions, which are subject to the line limit functionality constraints. Sites should ensure that they run this job with parameters set to ensure that the created transactions are within the line limit controls. Job log errors issued from the first job in the chain for input parameter edits should be addressed by

adjusting the input parameters or updating reference tables. The job step can then be restarted. Job log errors issued for invalid MAX_LINE_LIMITS should be addressed by correcting or deleting the invalid limits on the Transaction Component Requirements table in the Administration Application.

2.1.25 Reimbursement Request Recycling Process

Description

The Reimbursement Request Recycling process is run when a portion of a reimbursement request has been denied. Some reasons a reimbursement request may be denied are for insufficient available funding, or because the terms of the request are invalid (incorrect CMIA number, Letter of Credit, etc). The records related to the denied portion of a prior reimbursement request will be identified by terms retrieved by the specification of a Reimbursement Request Recycling (RECYCLE) parameter record ID into the chain job. That parameter ID will also specify the job ID of the original reimbursement request with the denied records. Department and Program or Appropriation are used to identify specific records in the original request that were denied. After the Reimbursement Request Recycling process will is run, the reimbursement records should be placed on hold until corrective actions are made in the cost accounting area. The records will be available for inclusion in a subsequent reimbursement request if not placed on hold, or when the hold is removed.

This process is a chain job consisting of 7 chains.

1. Reimbursement Recycle Request Process
2. SysmanUtil for uploading RE Transactions from XML
3. SysmanUtil for uploading CR Transactions from XML
4. SysmanUtil for uploading IET Transactions from XML
5. SysmanUtil for submitting CR Transactions
6. SysmanUtil for submitting RE Transactions
7. SysmanUtil for submitting IET Transactions

The Reimbursement Recycle Request Process retrieves records from the Reimbursement History (R_REIM_HIST) table based on a RECYCLE parameter. These retrieved records are then processed, and modified Cash Receipt, Receivable and Internal Exchange Transaction XML files are created based on the Reimbursement Output Type (attribute REIM_OTPT_FRT) of the Funding Line values selected. If the value is 1 then the process generates modification XML for Receivable transactions, if the format is 2 then the process generates modification XML for both Cash Receipt and Receivable transactions, if the format is 3 or 4 then the process generates modification XML for Internal Exchange Transaction transactions. The process also updates the Recycle Request Temporary (R_RCYC_REQ_TMP) table with all the records it processed. In case of any discrepancy while generating the XML, the process logs the error message on the corresponding record on the Recycle Request Temporary (R_RCYC_REQ_TMP) table. After the data processing is over, the process then generates reports based on the Recycle Request Temporary table. After the report generation process performs table updates on the basis of Reimbursement Request Temporary to back out the Reimbursement History table and to move data into the Ready for Reimbursement table so that the data is available for subsequent recycle process.

Thus, the selected records included in a reimbursement request will be backed out of the Reimbursement History table and from receivable transactions, and placed on the Ready for Reimbursement table. They will then be available for inclusion on a subsequent reimbursement request.

Additional jobs included in the chain will be steps using SysManUtil to load and process XML files created by the Reimbursement Request Recycling process.

Technical Flow

The Reimbursement Recycle Request Process consists of the following steps:

- Selection of Records from the Reimbursement History (R_REIM_HIST) table.
- Writing to the Reimbursement Request Temporary (R_RECYC_REQ_TMP) table.
- Based on the Reimbursement Output format (REIM_OTPT_FRMT) it generates XML transactions (that is, if output format is 1 then generate Receivable transactions, if output format is 2 then generate Cash Receipt and Receivable Transactions else if output format is 3 or 4 then generate Internal Exchange Transaction Transactions).
- Updates the Reimbursement Request Temporary (R_RECYC_REQ_TMP) records, with an error code (if any).
- Exception Report Generation.
- Table updates (if run in Report and Update mode).
- Upload RE, CR and IET Transactions from XML using SysManUtil.
- Submit RE, CR and IET Transactions using SysManUtil.

When to Run

At any time.

Major Input

Major input for this batch job is listed below:

Data Object	Definition
R_REIM_REQ_RCYC_PARM	This table stores the primary batch execution parameters, which are essential for the execution of the batch job. These parameters include Cycle and Job ID.
R_REIM_REQ_RCYC_SEL	This table stores the records selection criteria for the batch job. These parameters include the Recycle Type (that is, Program, Appropriation and Drawdown Group), Department (Common for Program and Appropriation Selection), Program (Program to be recycled), Appropriation (Appropriation to be recycled) and Drawdown Group (Drawdown Group to be recycled).
R_REIM_HIST	This table is the primary input table.

Output

This process generates two Reports as listed below.

1. Recycled Request Report.
2. Recycled Request Exception Report.

After the generation of Reports, the process does the table updates as given below.

1. Copies all the records from the Reimbursement Request Temporary table to the Reimbursement Request History (R_RECYC_REQ_HIST) table.

2. Copies all the records from the Reimbursement Request Temporary table to the Ready for Reimbursement (CA_REIM) table for which the XML transactions were created.
3. Deletes all the records from the Reimbursement History table included in a reimbursement request temporary table.

Parameters

Job	Parameter	Description	Default Value
Reimbursement Recycle Request	CLIENT_NM	Client Name to be displayed on Report	<blank>
Reimbursement Recycle Request	PARAM_ID	Parameter ID	Parameter ID to be used by the job

Custom Parameters

If you are running the job in the Financial application, you can click on the **Custom Parameter** link and select a parameter record. If you are running the job in the Administration application, enter a valid Parameter ID from the Custom Parameter table from the Financial application.

SWBP Parameters:

Reimbursement Recycle Request

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
AMSEXPOR T	Export Location at Reimbursement Recycle Request Job	Required	-	** Refer to Note: Assumptions for SWBP on page no. 5
AMSPARM	Parameter Location at Reimbursement Recycle Request Job	Required	-	** Refer to Note: Assumptions for SWBP on page no. 5
AMSLOGS	Logs Location at Reimbursement Recycle Request Job	Required	-	** Refer to Note: Assumptions for SWBP on page no. 5

Upload RE Transaction

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$AMSEXPOR T\$\$/R	-

			E_RCYC.X ML	
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/ E_RCYC.X ML	-

Upload CR Transaction

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/ R_RCYC.X ML	-
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/ R_RCYC.X ML	-

Upload IET Transaction

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
FILE_NM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/ T_RCYC.X ML	-
IMP_EXP_FI LENM	file name (export to/import from)	Protected	\$\$AMSEX PORT\$\$/ T_RCYC.X ML	-

Submit RE Transaction

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
EXCEP_REP _FILE_NM	The file name to which the exception report will be written out to	Protected	\$\$AMSLO GS\$\$/ RE_ EXCP.TXT	-
PARM_FILE	Parameter File	Protected	\$\$AMSPA RM\$\$/ RE_ PAR M.txt	-

Submit CR Transaction

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
EXCEP_REP_FILE_NM	The file name to which the exception report will be written out to	Protected	\$\$AMSLOGS\$/CR_EXCP.TXT	-
PARAM_FILE	Parameter File	Protected	\$\$AMSPARM\$/RECYCR_PARAM.txt	-

Submit IET Transaction

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
EXCEP_REP_FILE_NM	The file name to which the exception report will be written out to	Protected	\$\$AMSLOGS\$/IET_EXCP.TXT	-
PARAM_FILE	Parameter File	Protected	\$\$AMSPARM\$/RECYIET_PARAM.txt	-

Data Object Name: R_REIM_REQ_RCYC_PARM

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
CYC_ID	Cycle	Optional	<blank>	Cycle Note: When Cycle is entered, Job ID should be blank.
JOB_ID	Job ID	Optional	<blank>	Job ID Note: When Job ID is entered, Cycle should be blank.

Data Object Name: R_REIM_REQ_RCYC_SEL

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
RCYC_TYP	Recycle Type	Required		Valid values are Program, Appropriation and Drawdown Group.
DEPT_CD	Department	Optional	*	Entered Value should exist in R_DEPT table. This field is mandatory.
PROG_CD	Program	Optional	*	Entered combination of Department and Program should exist in R_PROG table. This attribute is required when the Recycle Type is set as "Program".
APPR_CD	Appropriation	Optional	*	Entered Appropriation for current fiscal year should exist in R_APPR table. This attribute is required when the Recycle Type is "Appropriation".
DRWDWN_GRP	Drawdown Group	Optional	*	Entered Drawdown Group should exist in R_DRWDWN_GRP table. This attribute is required when the Recycle Type is set as "Drawdown Group".
REAS_CD	Reason	Optional		Reason code. This field is required.

Sort Sequence

The Sort Sequence is **OTPT_DOC_ID, OTPT_DOC_CD, OTPT_DOC_DEPT_CD, OTPT_ACTG_LN_NO, REC_NO** as we have to create XML Transaction based on the Output Transaction information.

Selection Criteria

Minimum One Selection criteria (in terms of Recycle Type) are essential for this process. Hence the selection criteria depend on the selection parameter entered for the job.

Problem Resolution

If the process fails due to any reason then perform the corrective actions:

- Check the Log file for any errors that may have occurred while the program was running.
- Re-run the batch job.
- This batch program produces new Advantage transactions, which are subject to the line limit functionality constraints. Sites should ensure that they run this job with parameters set to ensure that the created transactions are within the line limit controls.

2.1.26 Update Assistance Listing Numbers

Chain or Job Name	Update Assistance Listing Numbers
Recommended Frequency	On Demand
Single Instance Required?	Yes
Can Be Restarted?	No
Reports Generated	None

Overview

The Update Assistance Listing Numbers chain updates existing records in, and adds records to, the Assistance Listing Number table based on data published at <https://sam.gov/content/assistance-listings>. The expected input file is the comma-separated value (CSV) file provided daily at <ftp://ftp.cfda.gov>.

Please note that <https://sam.gov/content/assistance-listings> provides two file options known as the “daily” and “weekly” files. Both files contain an extract of all programs. However, the weekly file (published once per week) contains significantly more data elements than what is stored as Assistance Listing data in Advantage Financial and is not supported by this chain job. The Update Assistance Listing Numbers chain only supports the daily file (published daily). <https://sam.gov/content/assistance-listings> describes the daily file as:

Daily File: This file is updated nightly and contains the following limited program data fields: Program Number, Program Title, and Agency. The file name adheres to the following naming convention: "programsYYDDD" with the 2-digit year and 3-digit Julian day, e.g., programs09159.csv

The Program Number, Program Title, and Agency data fields in this file respectively map to the Assistance Listing Number, Name, and Agency fields in the Assistance Listing (CFDA) page. However, the extract consistently also includes additional attributes that map to the Assistance Listing’s Published Date and Assistance Listing URL fields. Other fields within the Assistance Listing page (specifically the Federal Catalog Prefix and Federal Catalog Suffix fields) are **not** provided in the file but are derived automatically by CGI Advantage as records are inserted/updated.

The first step of the Update Assistance Listing Numbers chain converts the daily CSV file into an Extensible Markup Language (XML) file. The second step invokes the System Maintenance Utility with the XML file in “table overlay mode” to read each record from the XML file and either inserts it as a new record in the Assistance Listing Numbers database table or updates an existing record.

- The chain job invokes the following steps:
- Flat File to XML
- Load XML (System Maintenance Utility)

Major Input

- Input file (daily CSV file from <https://sam.gov/content/assistance-listings>)

Major Output

- Assistance Listing Number (CFDA / R_CFDA)

Chain Return Code

The following table shows the potential return codes for the Update Assistance Listing Numbers chain job.

Return Code	Condition
Successful (1)	All of the selected records are successfully added to or updated in the table.
Warning (4)	No eligible records found.
Non-Fatal Error (8)	This condition is not applicable for this chain job.
Failed (12)	The job fails under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid. • Run time exceptions for unexpected situations. When this job ends with a return code of <i>Failed</i> , subsequent jobs in the chain is set to <i>Inactive</i> .
Terminated (16)	This return code is issued when the job is terminated by the user.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues.

Problem Resolution

- In case any step fails, fix the issue and schedule a new chain.
- Update Assistance Listing Numbers: Flat File to XML

Overview

This process converts the daily CSV file into an XML file.

Process Steps	Messages
1. Parameter Validation	Parameters are valid or invalid depending on the validation.
2. Processing the Flat File	If the file is empty, the following message is issued: "No records found."

Major Input

- Input file (daily CSV file from <https://sam.gov/content/assistance-listings>)

Major Output

- Output file (XML file)

Batch Parameters

- Note: The default values listed are those delivered with the software. Actual values may vary based on your site's setup.

Parameter	Description	Default Value
Attribute List (ATTRIBUTE_LIST)	A required mapping parameter of the database field names from the Assistance Listing Number table that would match the fields (in sequential order) from the flat file. For example, this parameter value tells the system to map the first field in the flat file to the CFDA_NM field in Assistance Listing Numbers. The sequence of the list must match the sequence in the flat file.	CFDA_NM, CFDA_NO, AGCY_NM, PUB_DT, AGCY_SH_NM, CFDA_URL
Data Object Name (DATAOBJECT_NAME)	The required name of the database table used in XML creation.	R_CFDA
Data Delimiter (DELIMITER)	The required character is used to separate the fields in the flat file.	,
Flat File Name (FLAT_FILE)	The required file name of the CSV file of Assistant Listing Numbers and associated data. The file name changes daily and can be downloaded from the https://sam.gov/content/assistance-listings site. The file must be placed in the AMSExport directory.	(No Default)
Replace Special Characters (REPLACE_SPECIAL_CHARACTERS)	An optional output parameter indicating whether special characters need to be removed/replaced from the flat file. A value of <i>True</i> indicates that this data scrubbing occurs. A value of <i>False</i> or (blank) indicates that the data scrubbing is not occur. Some use this to replace special single-quote, double-quote, and hyphen characters with standard characters.	False
XML File Name (XML_FILE)	A required file name for the XML file created from the flat file for a later job step to load.	CFDANumbers.xml

Parameter Validation

In this step, the process verifies the parameters. If the parameter validation is successful, the job goes to the next step. Otherwise, the job ends with a return code of *Warning* or *Failed*.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	The flat file was not found in the Export directory.	Schedule a new chain, verify that the file is present in the Export directory, and change the Flat File Name parameter value, if necessary.	N/A
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	One or more parameters is/are not valid, and/or one or more required parameters is/are not provided.	Schedule a new chain and change the parameter values according to the messages in the log.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated.	A new job can be scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the system failure needs to be investigated.	A new job can be scheduled.

Processing the Flat File

Based on the attribute list provided as a parameter, this job creates an XML file, which is used as the input to the next job in the chain.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the records from the CSV file are written to the XML file successfully.	N/A	N/A
Warning (4)	This step doesn't issue this return code.	N/A	N/A
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	Job failed due to a fatal condition.	In this step, the job can fail if it encounters a runtime exception. Investigate the exception	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
		reported by the process, resolve the error and schedule a new job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated.	A new job can be scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the system failure needs to be investigated.	A new job can be scheduled.

Update Assistance Listing Numbers: Load XML

Overview

This process invokes the System Maintenance Utility in “table overlay mode” to read each record from the XML file and either inserts it as a new record updates an existing record.

Process Steps	Messages
1. Parameter Validation	Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value is displayed in the log.
2. Processing the XML File	If the file is empty, the following message is issued: “No records found.”

Major Input

- Input file (XML file)

Major Output

- Assistance Listing Numbers (CFDA / R_CFDA)

Batch Parameters

Note: The default values listed are those delivered with the software. Actual values may vary based on your site’s setup.

Parameter	Description	Default Value
Action Code (ACTN_CD)	The required code identifying the Table Overlay action to the	204

	SMU job.	
XML File Name (FILE_NM)	The required XML file name created in the previous job step.	\$\$AMSEXPORT\$\$/CFDANumbers.xml

Parameter Validation

In this step, the process verifies the parameters. If the parameter validation is successful, the job goes to the next step. Otherwise the job ends with a return code of *Failed*.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	This step doesn't issue this return code.	N/A	N/A
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	One or both parameters is/are not valid, and/or one or both required parameters is/are not provided.	Schedule a new chain and change the parameter values according to the messages in the log.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated.	A new job can be scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the system failure needs to be investigated.	A new job can be scheduled.

Processing the XML File

In this step the System Maintenance Utility job adds a record in the Assistance Listing table for each record in the XML file generated from the previous job. If a record in the table already exists, it is overwritten. Any records that exist in the Assistance Listing table but are not in the XML file are not be deleted or changed.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the records from the XML file are processed	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
	successfully.		
Warning (4)	At least one record was rejected for not meeting table constraints.	Copy the failed records to a new file, correct the data as indicated by the error(s), and schedule a new chain.	Because existing records are updated, simply editing the existing flat file and reprocessing all records is also an option.
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	Job failed due to a fatal condition.	In this step, the job can fail if it encounters a runtime exception. Investigate the exception reported by the process, resolve the error and schedule a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated.	A new job can be scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the system failure needs to be investigated.	A new job can be scheduled.

2.1.27 Warrant Reclassification

Chain Job Name	Warrant Reclassification
Recommended Frequency	After (standard) Reclassification
Single Instance Required	Yes
Can be restarted?	Yes, see the individual jobs for more details.
Reports generated	Yes, some of the jobs in the chain generate the exception report. Please refer to the individual jobs for more details.

Overview

This system process is necessary when the following situation applies:

- The Disbursement Option on the Expense tab of the System Options page is set to *Standard Warrant* or *Standard Warrant/EFT*.
- Front End Split is used where primary chart of accounts such as Fund and Sub Fund are set at the posting line (or any others that are used for tracking accounts for Warrants Payable and Cash).
- The Reimbursement chain job is used to modify disbursements when funding arrangements have changed.

The situation that requires this Warrant Reclassification process is when an Automated Disbursement (AD) or Manual Disbursements (MD) transaction has been redeemed with the processing of a Warrant Reconciliation (WR or AWR) transaction, the postings from the reimbursement modification of that disbursement will reclassify warrants payable that has already been cleared through cash.

Warrant Reclassification process has the following batch jobs in the chain.

1. [Warrant Reclassification and CH XML Generation](#)
2. [Load and Submit Transactions](#)
3. [Warrant Reclass Exception Report](#)

Major Input

- Transaction Headers for the Automatic Disbursement (AD_DOC_HDR) and Manual Disbursement (MD_DOC_HDR)
- Front End Split Log (FSL)

Major Output

Charge transactions

Chain / Job Return Code

The job will end with the following return codes as per conditions encountered.

Return Code	Condition
Successful (1)	All of the jobs end successfully.

Warning (4)	One of the jobs in the chain ends with a return code of "Warning".
Non Fatal Error (8)	One of the jobs in the chain ends with a return code of "Non Fatal Error".
Failed (12)	One of the jobs in the chain ends with a return code of "Failed".
Terminated (16)	One of the jobs in the chain ends with a return code of "Terminated".
System Failure (20)	One of the jobs in the chain ends with a return code of "System Failure".

Problem Resolution

- In case any step fails, fix the issue and schedule a new chain.

Warrant Reclassification Chain Job: Warrant Reclassification and CH XML Generation

Job Name	Warrant Reclassification and CH XML Generation
Recommended Frequency	See chain
Single Instance Required	See chain
Can be restarted?	Yes
Reports Generated	No

Overview

The Warrant Reclassification and CH XML Generation kicks off all the other jobs in the chain. It includes selection of eligible records from the Funding Split Log table and matches them to the AD and MD Header Catalogs.

Steps

The process can be divided into the following logical steps: parameter validation, selection of FSL records, Update FSL Records, AD/MD Header Verification and creating XML.

1. **Parameter Validation** – The job parameters are validated before the actual processing starts. See the parameter validation process step in the table that follows for the primary parameter edits beyond just the "required parameter missing" edits.
2. **Selection logic** – If the parameters are validated successfully, the process will select the eligible records from the Funding split log (FSL) table.
3. **Update FSL Records** – All records selected from above selection logic are updated as *true* for the Warrant Reclassification indication.
4. **AD/MD Header Verification** –With the set of FSL records that were reclassified since the last Warrant Reclassification process, the job step will then look to see if the AD or MD has been cleared with a WR/AWR (e.g. Warrant Reconciliation indication on AD/MD Header is *true*). If not cleared, then when the clearing occurs the most recent posting lines of the disbursement (resulting from the Reclassification) will be used in the clearing.
5. **XML Creation** – CHDocument.xml is then created for the transactions.

The following progression messages indicate the status of the job.

Process Steps	Messages
1. Parameter Validation	<p>Run Started</p> <p>Each valid parameter is listed followed by any value.</p> <ul style="list-style-type: none"> • When the Event Type specified does not exist on AETDC for the Transaction Code parameter then “Invalid Event Type for Transaction Code” is recorded in the job log. • If Progression Counter is not a valid integer greater than zero, then “Progression Counter must be a positive whole number greater than zero” is recorded in the job log. • When the Charge transaction code is invalid then “Charge Transaction Code not valid and active on DCTRL for CH transaction type” is recorded in the job log • When the Department entered is invalid on Department (DEPT) then “Entered CH Department is not valid” is recorded in the job log. • When the Unit is not valid in the current fiscal year (according to the Application Date) with the department or it is valid but inactive or not within effective dates then “Unit entered is either invalid, inactive or not within the effective date range” is recorded in the job log. • When the combination of transaction parameters are not valid on ADNT in the current fiscal year then “ADNT entry not found for CH Prefix, CH Transaction Code, and CH Department is recorded in the job log. <p>Parameters Successfully Validated</p> <p>Run Ended</p>
2. Selection of Records	<p>Begin selection of eligible FSL Records</p> <p>## Records Selected</p> <p>Finished processing FSL Records</p> <p>If there are no eligible records found on Funding Split Log (FSL), the message “No records found for reclassification” is logged.</p>
3. Update FSL Records	<p>No messages are logged for this step</p>
4. AD/MD Header Verification	<p>No messages are logged for this step</p>
5. XML Creation	<p>No messages are logged for this step</p>

Major Input

- Transaction Headers for the Automatic Disbursement (AD_DOC_HDR) and Manual Disbursement (MD_DOC_HDR)
- Front End Split Log (FSL)

Batch Parameters

Note: The default values listed are those delivered with the software. Actual values may vary based on your site's setup.

Parameter	Description	Default Value
AMSEXPORT	Required Export Location for Warrant Reclassification	\$\$AMSROOT\$\$/ExportImport
AMSPARM	Required Parameter Location for Warrant Reclassification	\$\$AMSROOT\$\$/Parms
AMSLOGS	Required Logs Location for Warrant	\$\$AMSROOT\$\$/Logs
Charge Transaction Code (DOC_CD)	Required Transaction Code used to create transactions.	No Default (CH delivered)
Charge Department Code (DOC_DEPT)	Optional Department Code used to create transactions.	No Default
Charge Unit Code (DOC_UNIT)	Optional Unit Code used to create transactions.	No Default
Event Type (EVNT_TYP)	Required Event Type used on all accounting lines.	No Default (DR22 delivered)
Maximum Number of Accounting Lines per Transaction MAX_ACTG_LN	Required The maximum number of accounting lines written to one transaction instance before starting a new instance. Ensure not greater than the CH_DOC_ACTG record for MAX_LINE_LIMIT on Transaction Component Requirements (DCREQ).	100
Charge Prefix (PREFIX)	Optional Transaction Prefix used to create transactions for ease of identification to this system process.	No Default
Progression Counter Size PROG_CNTR	Required Progression Counter Size used to issue progression messages to the job log	1000

Major Output

CHDocument.xml

Batch Return Codes

The following table shows the potential job return codes for the Warrant Reclassification process.

Return Code	Condition
Successful (1)	The job ran to completion without any errors or exceptions.
Warning (4)	No eligible FSL records were selected for processing.
Non Fatal Error (8)	This job step does not end with this return code.
Failed (12)	Parameters are invalid. Runtime exceptions encountered for any unexpected situations. When the job ends with a return code of Failed, subsequent jobs in the chain are set to Inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain are set to Inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain are set to Inactive.

Sort Sequence

NA

Selection Criteria

Front End Split (FSL) Selection

1. Where Transaction Type = AD or MD
2. Where Transaction Function = 2 (*Modification*)
3. Where Warrant Reclassification indication is *false*
4. Where Posting Code has Funding Split = *Split for Reimbursement*

Select the same FSL record from the prior version as it will also be used.

Problem Resolution

The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain.

Possible Return Codes	Condition	Recommendation	Other Instructions
Failed (12)	Failed while validating the parameter. Sample Message: Invalid	Recommendation: Schedule a new job, with valid parameters.	Refer to the job log message for the reason of invalid

	Parameter Failed with a runtime Exception.	Recommendation: Refer the application logs.	parameter. User need to try to find out the problem that made the job fail unexpectedly, through application logs if possible, and then try scheduling a new instance.
--	---	---	---

Warrant Reclassification Chain Job: Load & Submit Transactions

Job Name	Load & Submit Transactions
Recommended Frequency	See chain
Single Instance Required	See chain
Can be restarted?	Optionally, based on the Save Restart Information parameter.
Report Generated	No

Overview

This job starts by first validating the batch parameters. If the parameters are valid, then it loads the records from the XML file(s) generated by the Warrant Reclassification job into the Transaction Catalog. This job uses the Multi-Process Import feature to load and submit the transactions created in the XML file from the first job step.

Major Input

- CHDocument.XML

Process Steps	Messages
1. Parameter Validation	<p>Run Started</p> <p>Each valid parameter is listed followed by any value.</p> <p>Validating Batch Parameters.</p> <ul style="list-style-type: none"> • If the Thread Count is not a valid integer then “Invalid value received for parameter THREAD_COUNT (expected number): ##” is recorded in the job log. • If the Thread Count is not valid integer > 0 then “The valid job thread count starts with 1. Receive value: ##” • If the Mode is not provided then “MODE parameter cannot be left blank (expected values 1, 2, or 3)” is recorded in the job log. • If the Mode is provided and mode is not => 1 and not =< 3 then “Invalid value received for parameter MODE (expected values 1, 2, or 3): ##” is recorded in the job log.

	<ul style="list-style-type: none"> • If the Commit Block Size is not a valid integer > 1 then “Invalid value received for parameter COMMIT_BLOCK_SIZE (expected value >1): ##” is recorded in the job log. • If the Block Size is not a valid integer > 1 then “Invalid value received for parameter BLOCK_SIZE (expected value >1): ##” is recorded in the job log. • If the Polling Frequency is a valid integer and < 5 then “Invalid value received for parameter SLEEP_INTERVAL (expected value >=5): ##” is recorded in the job log. • If the Logging Frequency is a valid integer and less than 30 then “Invalid value received for parameter LOG_STATUS_INTERVAL (expected value >= 30): ##” is recorded in the job log. • If the Catalog ID of the System Maintenance Utility job is not a valid integer greater than one then “Invalid value received for parameter SMU_CTLG_ID (expected value >1): ##” is recorded in the job log. • If the Other Action is not provided and run mode is 3, then “Other Action is required when run on mode 3” is recorded in the job log. • If the Other Action is provided and run mode is 1 or 2 then “Other Action is valid only for Mode 3” is recorded in the job log. • If the location of the source file of records is not a valid directory, then “Invalid value read for parameter FILE_INPUT_DIR (expected value = a valid directory): #####” is recorded in the job log. • If the Stagger Time is not a valid integer > 1 then “Invalid value received for parameter STAGGER_TIME (expected value >1): ##” is recorded in the job log. • If the Comma Separated list of input XML files is not provided then “No input file to process” is recorded in the job log. • If the Comma Separated list of input XML files is provided and a file does not exist or does not end with xml then “Invalid / non existing file input part in value for parameter FILE_LIST (expected value = a valid list if file):#####” is recorded in the job log. • If the Output Location for the file segments is not a valid directory then “Invalid value read for parameter FILE_OUTPUT_DIR (expected value = a valid directory): #####” is recorded in the job log. <p>Parameter validation completed.</p>
<p>2. Multiple Job Processing</p>	<ul style="list-style-type: none"> • Setting up job for parameter file: (Parameter Location path) \ Parm_#####_#.txt • Enabled job with Id=##### • Created Job ##### for Parm file (Export Import directory path)\Parm_#####_#.txt (repeated for each file and job combination) • Committed child jobs • Child job pending count:# Job Id[#####] Status=X Repeated

	<p>for each job)</p> <ul style="list-style-type: none"> • Deleting part exception file: (Export Import directory path)\WarRclsExcep_#####_#.txt (Repeated for each file) • Deleting split input XML file: (Export Import directory path)\CHDocument_#####_#.xml, (repeated for each file) <p>Run Ended</p>
--	--

Batch Parameters

Note: The default values listed are those delivered with the software. Actual values may vary based on your site’s setup.

Parameter	Description	Default Value
BLOCK_SIZE	Optional block size for the number of transactions to be split out into files. If left blank, 1 will be the default. Set this performance parameter to a value that will split an XML file into individual files sufficient for timely processing.	10
COMMIT_BLOCK_SIZE	Optional number of records to commit at a time. If left blank, the ADV30Parms.ini file will supply a value.	10
FILE_INPUT_DIR	Required location of the source file of records.	\$\$AMSROOT\$\$/ ExportImport/
FILE_LIST	Required comma separated list of input XML files to load.	CHDocument XML file.
FILE_OUTPUT_DIR	Required output location for the file segments.	\$\$AMSROOT\$\$/ Logs/
FILE_PREFIX	Optional file prefix for Parameter and Data Files created by jobs.	No default
I_SMU_APPLY_OVERRIDES	Required System Maintenance Utility flag to indicate Apply Overrides in Import Mode.	true
I_SMU_BYPASS_APPROVAL	Required System Maintenance Utility flag to indicate Bypass Approvals in Import Mode.	false
I_SMU_BYPASS_ADNT_FL	Required System Maintenance Utility flag to indicate Bypass ADNT in Import Mode.	true
I_SMU_DOC_STA_CD	Required System Maintenance Utility Transaction Status for Import Mode.	2
I_SMU_OVERRIDE_LVL	Optional System Maintenance Utility Override Level for Import Mode. If 0 or left blank and the Apply Overrides parameter is <i>true</i> then the level of the user executing the chain job will be used.	No default
I_SMU_RESTART_FL	Required System Maintenance Utility flag to indicate Save Restart info in Import	true

	Mode.	
LOG_STATUS_INTERVAL	Required logging frequency (in seconds) for the controller thread reporting status of child threads to the system log.	300
MODE	Required mode (1=Import Only, 2-Import & Submit, 3-Import, Other Action, & Submit).	2
SLEEP_INTERVAL	Required polling frequency (in seconds) for internal controller thread for checking child processes.	5
SMU_CTLG_ID	Required Catalog ID of the System Maintenance Utility job that is spawned as the child process.	3
STAGGER_TIME	Required lag time in seconds between the spawning of each child process.	30
S_SMU_APPLY_OVERRIDE	Required System Maintenance Utility flag to indicate Apply Overrides in Submit Mode.	true
S_SMU_BYPASS_APPROVAL	Required System Maintenance Utility flag to indicate Bypass Approvals in Submit Mode.	false
S_SMU_EXCEPT_FILE_NM	Required System Maintenance Utility Exception File for Submit Mode.	WarRclsExcep.txt
S_SMU_EXCEPT_IND	Required System Maintenance Utility Exception Report Indicator for Submit Mode. Values are: 1 = Detailed, 2 = Failed Transactions, 3 = Processed Transactions, 4 = Failed Transaction Lines, and 5 = Transaction Status.	4
S_SMU_EXP_SEV_FL	Optional System Maintenance Utility flag to indicate the Severity Flag in Submit Mode.	1
S_SMU_OVERRIDE_LVL	System Maintenance Utility Override Level for Submit Mode. If 0 or left blank and the Apply Overrides parameter is <i>true</i> then the level of the user executing the chain job will be used.	No default
S_SMU_RESTART_FL	Required System Maintenance Utility to indicate Save Restart info in Submit Mode.	true
THREAD_COUNT	Required number of jobs to start.	1

Major Output

- The result of this job is the loading of transaction records from the input file. Files are first split and then imported.
- During job processing there will be two categories of temporary files (they will be deleted upon successful job completion):
 - Split input files

- Exception report files for each child job.
- If any of the child jobs encounter an exception an appropriate exception message will be merged into the Exception Report file for the job (specified by name in the job parameters).
- A parameter file is generated for each split XML file, and then the spawned jobs read those input parameter files in a sequential order.

Batch Return Codes

The following table shows the potential job return codes for Load & Submit Transactions.

Return Code	Condition
Successful (1)	All of the records are loaded into the Transaction Catalog successfully or the input file is empty.
Warning (4)	This return code is issued when some of the records failed to load.
Non Fatal Error (8)	No records loaded into the Transaction Catalog.
Failed (12)	<ul style="list-style-type: none"> • Parameters are invalid • When the input file is not found in the specified directory • Runtime exceptions encountered for any unexpected situations • When the job ends with a return code of Failed, subsequent jobs in the chain are set to inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain are set to inactive.

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

If the process fails for any reason the Load and Submit Transactions job supports restart functionality. Upon restarting the Load and Submit Transactions job, the failed job will be picked up and run. A checkpoint is maintained for the overall Load and Submit Transactions job as well as the individual child jobs to identify how far the job successfully executed before failing. Therefore, when the user restarts the failed job the system does not start from the first step of the Load and Submit Transactions but instead starts processing from the exact point at which it failed.

For example, assume that the failed job split the input file into five child jobs and that the third and the fourth child jobs failed. When restarted, only those failed child jobs, that is, the third and the fourth ones will be picked up and processed. The restarted job would not process child jobs one, two or three given they have already completed successfully in the earlier run.

The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain. For general errors and recommendations, refer to the SMU Transaction Upload run sheet in the *CGI Advantage Financial – Utilities Run Sheet Guide*.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All transactions were loaded and submitted.	N/A	N/A
Warning (4)	This return code is issued when the job fails to load some of the transactions. Sample Message: Unable to load all of the transactions into the catalog.	Analyze the reason why records failed to load to the Transaction Catalog. When the current run is a re-execution of a previous run (the same input file is used in a new job) this error may be seen because the records that previously loaded successfully will not load again (duplicate entry on the Transaction Catalog is not allowed). If the records failed to load to the Transaction Catalog due to any other reason, then analyze the reason, resolve it and schedule a new job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.
Non-Fatal Error (8)	This return code is issued when the job failed to load all of the transactions.	Correct what is most likely a setup issue and restart the spawned SMU job and then restart the load/submit job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.
Failed (12)	Failed under the following conditions: <ul style="list-style-type: none"> • Issues in spawning jobs • Files not found • Runtime exceptions • Parameter Edits 	The reason for the failure needs to be investigated and fixed before restarting the job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated and fixed before restarting the	If another instance of the job has already been scheduled and ran successfully, then

		chain.	this job should not be restarted – only a new job should be scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before restarting the chain	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.

Warrant Reclassification Chain Job: Warrant Reclass Exception Report

Job Name	Warrant Reclass Exception Report
Recommended Frequency	See chain
Single Instance Required	See chain
Can be restarted?	No
Report Generated	Yes – Exception Report

Overview

This job in the Warrant Reclassification process generates an exception report that lists all of the errors encountered when the CH transactions are submitted in the earlier step. The report contains the following information:

- Rejected CH transactions
- Detailed error description along with the error code.

The following table shows the progression messages issued in this job.

Process Steps	Messages
1. Parameter Validation	<ul style="list-style-type: none"> • Validating Batch Parameters • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. It will be followed by message “Batch Parameter validation failed”. • Batch Parameters validation completed.
2. Creating exception report	<ul style="list-style-type: none"> • Generation of Exception Report started. • Generation of Exception Report completed.

Restartability Information

- Job cannot be restarted in this run mode. If the job fails in any of the above steps, a new instance of this job should be scheduled after correcting the errors that caused the job to fail.

Major Input

- Draft CH transactions in the catalog with a Rejected status

Batch Parameters

Parameter	Description	Default Value
AMSPARM	Required. Parameter Location.	\$\$AMSROOT\$\$/Parms

Output

Warrant Reclass Exception Report

Batch Return Codes

The following table shows the potential job return codes for the Warrant Reclass Exception report job in the chain.

Return Code	Condition
Successful (1)	All of the transactions generated in that run submitted successfully.
Warning (4)	Not applicable for this job.
Non Fatal Error (8)	Not applicable for this job.
Failed (12)	<ul style="list-style-type: none"> • Input file is blank. • Input parameter file is not found in the specified folder. • Runtime exceptions encountered for any unexpected situations. • When the job ends with a return code of failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain are set to inactive.

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Input file is not entered. Sample Message: Exception file is required	Specify the Input file name and schedule a new job.	
	Input file is not found on the specified directory. Sample Message: The system cannot find the specified file.	Make sure that the specified file exists in the directory and restart the job. Alternatively, a new job can be scheduled. At times, the directory path may not be accurate. If the job cannot be restarted, schedule a new job.	Correct the path and reschedule the job.
	Failed while restarting the job since another instance of the job has already been run successfully. Sample Message: Cannot restart the job since another instance of this job has already been run successfully.	Schedule a new job.	
	Failed because of runtime exceptions for an unexpected situation.	The reason for the Runtime Exception needs to be investigated. Correct the problem and schedule a new job.	System error log and VLS log should be investigated to find out the possible reason

	Sample Message: Error occurred during validateParam: <Runtime Exception Message>		of exception.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or rescheduled.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or rescheduled.	

2.2 Cost Accounting Report Processes

The Cost Accounting report run sheets included in this section are:

- [CMIA Discrepancy Report](#)
- [CMIA Interest Calculation Report](#)
- [Negative Draw Prevention Report](#)

2.2.1 CMIA Discrepancy Report

Description

The purpose of the Cost Allocation Process is to ensure that the CMIA table is set up correctly for the offline reimbursement processes. Specifically, where CMIA drawdown records have been created on the CMIA setup table, the total clearance percentage for all records for each Program Period should equal to 100 percent.

Functional Flow

The job reads the CMIA Setup table. Records are sorted in order by Department, Major Program, Program Period and Clearance Day. For each combination of Department, Major Program, Program Period, the total clearance percentage for all records should equal 100 percent. An exception report is created to identify those Program Period records whose total clearance percentage is not equal to 100 percent.

Technical Flow/Specifications

When to Run

It is recommended that this job be run before the Reimbursement Generation Process as that process is dependent on the CMIA setup table.

Run Sequence

There is a single job and only run mode exists for the job. It is not part of any of the Reimbursement Process chains.

Parameters

Batch Parameters

None

Custom Parameters

None

Inputs

All records on the CMIA Setup Table

Outputs

CMIA Discrepancy Report highlights those Program Period records that are referenced on the CMIA Setup table but whose total clearance percentage is not equal to 100%.

Sort Criteria

Records are sorted in order by Department, Major Program, Program Period and Clearance Day.

Selection Criteria

All records from the CMIA Setup table are selected

Troubleshooting

If the process fails due to any reason then check the Log file for any errors that may have occurred while the program was running.

2.2.2 CMIA Interest Calculation Process

Description

- The Cash Management Improvement Act (CMIA) of 1990 makes the following provisions:
- Federal agencies must make timely fund disbursements and grant awards to States.
- States and Federal agencies must minimize the time between the transfer of Federal funds to States and the presentment of States' checks/warrants or settlement of Electronic Fund Transfer (EFT) payments for program purposes.
- With minor exceptions, States are entitled to interest from the Federal Government for the time State funds are advanced for program purposes pending Federal disbursement. The Federal Government is entitled to interest from the States for the time Federal funds are in State accounts pending presentment of checks/warrants and settlement of EFT payments for program purposes.
- Treasury may charge responsible Federal agencies if they are found to be egregious or repeatedly incur Federal interest liabilities. Interest charges will be paid from agency operating budgets and not amounts available for program funding.
- The Reimbursement Selection and Calculation, Generation and Output processes will do the required processing to bill different customers for different Major Programs and Programs. The Reimbursement Generation process calculates the CMIA expected request date and CMIA expected Drawdown date. This information is stored in the Reimbursement History (R_REIM_HIST) table after the reimbursement output process completes the billing. If there is a difference between the expected Drawdown date and Actual Drawdown date interest is due to or from the Federal Government. The CMIA Interest Process calculates this interest. The interest is calculated based upon the CMIA method on the record. The CMIA method is determined at the point of reimbursement setup when the user specifies a CMIA Calculation method for each funding line.
- The Financial Management Service (FMS) requires a State to submit an Annual Report by December 31st accounting for interest liabilities of the State's most recently completed fiscal year. The reports produced by the CMIA Interest Calculation process are in compliance with those regulations set forth by FMS.

Steps in Running this Process

The CMIA Interest Calculation Process consists of six steps where the processing sequence is impacted by the selected run mode:

1. Selection of records
2. Checking the CMIA Interest Bearing flag
3. Calculate the interest
4. Generate the Exception Report
5. Generate the detail report with the interest calculated for each record
6. Generate the summarized report with the calculated interest

Step 1 - Selection of Records: The criteria for selection depends upon Customer Id, Department, Major Program, Program Period Code and the Date range defined on the custom parameter record. The records that match the entered values and have the Record date within the date range will get selected from Reimbursement History Table (R_REIM_HIST) for

processing. Only those records that have the Output Transaction Code field populated and a CMIA Calculation method that is not set to 'None' are selected. All records with negative posting amount and meeting the above criteria are selected. In addition to the above criteria, for records with positive posting amounts, credit type indicator should be '2'; that is, the posting amount should be greater than \$10,000.

Step 2 - Checking the CMIA Interest Bearing Flag:

This step verifies the CMIA Interest Bearing Flag on the Funding Profile (R_FPRFL) table and excludes those records for which this flag is set to *false*.

Step 3. - Calculating Interest:

This step verifies the CMIA method and calculates the interest accordingly:

Based on the Output Transaction code, Output transaction ID, Output Vendor line number and Output Accounting line number obtained from the Reimbursement History (R_REIM_HIST) table the process looks up the CR accounting line reference fields (CR_DOC_ACTG). If the CR is final the process obtains the Transaction Record date and the Bank Deposit date from the CR Transaction Header. (CR_DOC_HDR). If the CR is not final but in draft phase the record is ignored from processing. The Bank Deposit date required for the interest calculation. If this field is blank then no interest will be calculated.

If the posting amount is negative the Transaction Record date of the CR is used whereas for positive records the Bank deposit date is used.

Calculation of Interest: The interest calculation method is the same for all records selected, the one exception being those records that use the Pre-Issuance Funding method.

For Pre-Issuance Method

Find the clearance pattern for the CMIA method from the CMIA Clearance Table.

Calculate interest for all the clearance days

Interest basis = (Clearance Percentage * Clearance Day * Posting Amount). Interest amount = Interest basis * (Interest rate / 365)

Sum the interest calculated for each clearance day

For All Other CMIA Methods

For positive dollar amounts the interest calculation formula is: Interest basis = (Bank deposit date – CMIA Draw down date) * Posting Amount. Interest amount = Interest basis * (Interest rate / 365)

For negative dollar amounts the interest calculation formula is: Interest basis = (Transaction Record date – CMIA Draw down date) * Posting Amount. Interest amount = Interest basis * (Interest rate / 365)

NOTE: Weekends and holidays are to be excluded while calculating the Interest basis. Weekends and Holidays are verified by a look up in Calendar Date Table (R_CLDT) for Weekend Flag (WKND_FL) or Holiday Flag (HLDAY_FL) or Federal Bank Holiday Flag (FED_BANK_HLDAY_FL).

Step 4 - Generate Exception Report:

This step creates the Exception report and two exceptions are tracked in the Exception report.

If the CMIA Calculation Method on the record from the Reimbursement History (R_REIM_HIST) table differs from that on the Funding Line (R_FLINE) table.

If the CMIA Calculation method is Pre-Issuance and the Total Clearance Percentage in the CMIA Setup table for all records for a Program Period does not total to 100%.

The records for which the Clearance Percentage total does not equal 100% will be excluded from further processing and written to the exception report.

Step 5 - Generate Detail Report:

This step will create the CMIA Detail report.

The report will be at a detail level (for each record with calculated interest, CMIA methods etc.) and grouped by Customer and within Customer by the Programs funded by the Customer. The report grouping is as per Customer and Program irrespective of the parameters entered.

Step 6 - Generate Summarized Report:

This step will create the CMIA Summarized report.

This report will show the total interest Due To Federal Government and Interest Due From the Federal Government separately for each Program (Department, Major Program and Program Period) grouped by Customer.

Depending on the run mode the above steps will be called:

In the detail run mode steps 1 to 5 will be called.

In the summarized mode steps 1 to 6 will be called.

Run Modes

The majority of the cost accounting offline processes can be submitted in three different “run modes” (for this process, only two modes are applicable) to provide the user community and the system administrators additional flexibility when requesting complex and intricate offline processes. The user selects a Run Mode on the CMIA Interest Calculation Parameter page prior to running the process. The following are the descriptions of the run modes available in the CMIA Interest Calculation Process:

Detail

In the detail mode interest is calculated and a detail report is generated which contains the details for each record that was processed. The detail mode allows users to produce CMIA Interest Calculations periodically for reporting purposes.

Summarized

In the summarized mode all the steps of the detail mode are included, that is, the detail report is produced. In addition, this mode generates a summarized report which shows the total interest calculated for each Department, Major Program, , and Program Period Code combination. This report complies with the annual report format required by FMS.

Please note the records with the calculated interest amount are not stored in any table. They are written directly to the appropriate report.

When to Run

User’s discretion. Summarized mode must be executed prior to the calendar year end to report interest liabilities to FMS.

Major Input

Run Mode	Input
Detail	Reimbursement History Table (R_REIM_HIST) Custom Parameter (R_CMIA_INTR_PARM) Funding Line Table (R_FLINE) Funding Profile Table (R_FPRL) CMIA Clearance Table (CMIA_CLR) Calendar Date Table (R_CLDT)
Summarized	Same as Detail mode

Outputs

Process Output is dependent on the Run Mode

Run Mode	Output
Detail	CMIA Detail Report Exception Report
Summarized	CMIA Detail Report Exception Report Summarized Report

Parameters

Batch Parameters

Field Name	Description (Caption)	Default Value
PARM_ID	Parameter ID	0

Custom Parameters

If you are running the job in the Financial application, you can click on the **Custom Parameter** link and select a parameter record. If you are running the job in the Administration application, enter a valid Parameter ID from the Custom Parameter table from the Financial application.

Data Object Name R_CMIA_INTR_PARM

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
PARAM_ID	Parameter ID	Primary Key	0	Enter Parameter Id
EXEC_MOD	Execution Mode	Required	Report Only	Select the value from the CVL_SUMM_TYP.
DEPT_CD	Department	Required	Blank	Enter the Department Code
CUST_ID	Customer Id	Optional	Blank	Enter the Customer Id. If entered Major Program and Program must not be entered.
MJR_PROG_CD	Major Program	Optional	Blank	Enter the Major Program Code. If entered Program Period is required and Customer is prohibited.
PPC_CD	Program Period	Optional	Blank	Enter the Program Period Code If entered Major Program is required and Customer is prohibited.
INTR_RATE	Interest Rate	Required	Blank	Enter the Interest Rate. Must be between zero and 100.
STRT_DT	Start Date	Required	Blank	Enter the start date for record selection Should be less than the End Date.
END_DT	End Date	Required	Blank	Enter the start date for record selection. Default value is Current Date.
BANK_INTR*	Bank Interest	Optional	Blank	Enter Bank Interest.

Field Name	Description (Caption)	Required/Optional	Default Value	Comments
				Required if the Run mode is UPDATE. Must be a positive amount
BANK_CHRG*	Bank Charge	Optional	Blank	Enter Bank Charges. Required if the Run mode is UPDATE. Must be a positive amount

Please note the fields with asterisks (*) beside them are included in the summarized report as required by the format specified for summarized report. These are not included in the calculations.

The Department, Major Program, Program Period Code and Customer ID are validated against respective set up tables.

Sort Criteria

Records selected from the Reimbursement History Table are sorted in the following order:

- Customer
- Department
- Major Program
- Program Period
- Funding Profile
- Funding Priority
- Funding Line

The selected records will be sorted so that all the records having similar Cost Accounting Chart of Accounts that are grouped together.

Selection Criteria

Run Mode	Inputs
Detail	Select the records from Reimbursement History Table (R_REIM_HIST) that match the parameters entered, CMIA Expected Request Date is within the date range, CMIA Calculation method not = '1' (None) and the Credit Type Indicator is equal to 2 OR Posting Amount is negative
Summarized	Same as Detail mode

Problem Resolution

If the process fails due to any reason then perform the corrective actions:

- Check for the CMIA Interest Bearing flag for the Funding Profile.
- Check that the Cash Receipts (CR) are created for the Receivables (RE) on the record and created Cash Receipt (CR) is on the Transaction Catalog as “Final.”
- Check the Log file for any errors that may have occurred while the program was running.
- Correct the errors.
- Re-run the process.

2.2.3 Negative Draw Prevention Process

Job Name	Negative Draw Prevention Process
Recommended Frequency	Should be run frequently (daily or weekly). Must be run more frequently after the other Reimbursement chains (Reimbursement Generation and Reimbursement Output).
Single Instance Required	Yes.
Can be Restarted?	Yes.
Reports Generated	For this batch job – see details on the jobs.

Description

This job will retrieve records from the selected data source, excluding records matching exception criteria as per parameter definition. The selected records will then be sorted and summarized at a high level. The detail input records will then be reprocessed, and those that belong to a group of records that summarized to zero, a negative amount, or below an applicable minimum will be removed from the Reimbursement Request table and placed on the Ready for Reimbursement table. A minimum amount will apply if all of the following are true:

1. The customer has a non-zero Minimum Billing Amount.
2. The Final Billing flag is not checked for the program.
3. The Prevention Control field is set to one of the following on the Negative Draw Prevention ID used.
 1. *Department, Program and Customer Account*
 2. *Department, Major Program and Customer (Account)*

Technical Flow

The Negative Draw Process consists of the following steps:

- Selection of Records from the specified Data Source passed as a parameter and copying them to Temporary tables.
- Identification of Exception Records and marking them as Exception records.
- Report Generation.
- Table updates (if run in Report and Update mode)

When to Run

User's Discretion – Recommend to be executed in conjunction with the Reimbursement Generation process.

Restartability Information

This job has the Restartability feature and can be restarted only when a previous instance of the job ended with a Return Code of *Terminated*. The process has check point logic in the following areas of processing:

- Selection of records
- Processing of negative records

Job Return Codes

The following table shows the potential job Return Codes for the Negative Draw Prevention Process job.

Return Code	Condition
Successful (1)	All selected records are processed successfully.
Warning (4)	No records found in the R_REIM_REQ, R_REIM_HOLD, or R_REIM_REQ_TMP tables to process.
Non-Fatal Error (8)	The job does not end with this status.
Failed (12)	The job will fail under the following conditions: <ul style="list-style-type: none"> • Parameters are invalid • Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Major Input

Major input for this batch job is listed below:

Data Object	Definition
R_NEG_DRW_PRV_PARM	This table stores the primary batch execution parameters, which are essential for the execution of the batch job. These parameters include Run Mode (indicates the mode in which the job should be executed, that is, Report only or Report and Update); Data Source (the Data Source to be considered as an input for the job, that is, Reimbursement Holding , Temporary Reimbursement Request or Reimbursement Request); Expected Request Date and Prevention Control (this parameter specifies the criteria for grouping).
R_NEG_DRW_EXCP_DTL	This table stores the Exception criteria for the batch job (if any). These parameters include the Exception type (that is, Event or Program) and Exception value (Event Type or Department and Program).
R_REIM_HOLD	This table is the primary input table if the Data Source is set to Reimbursement Holding.
R_REIM_REQ	This table is the primary input table if the Data Source is set to Reimbursement Request.

Data Object	Definition
R_REIM_REQ_TMP	This table is the primary input table if the Data Source is set to Temporary Reimbursement Request.

Output

This process generates all **three** Reports as listed below:

1. Adjusted Reimbursement Request Report
2. Reimbursement Request Exception Report
3. Negative and Minimum Draw Report

If this batch is executed in Report and Update mode, this batch job apart from generating reports also performs table updates as listed below.

1. Sets the EXCP_FL on the Reimbursement Request (R_REIM_REQ) table as true for those records that matches the exception criteria.
2. Copies the Negative Draw Records from the Reimbursement Request (R_REIM_REQ) table to the Ready for Reimbursement table (CA_REIM).
3. Deletes the Negative Draw Records from the Reimbursement Request (R_REIM_REQ) table.

Parameters

Patch Parameters

Job	Parameter	Description	Default Value
Negative Draw Prevention	CLIENT_NM	Client Name to be displayed on Report	<blank>
Negative Draw Prevention	PARAM_ID	Parameter ID	Parameter ID to be used by the job

Custom Parameters from Negative Draw Prevention (NEGPREV)

If you are running the job in the Financial application, you can click on the **Custom Parameter** link and select a parameter record. If you are running the job in the Administration application, enter a valid Parameter ID from the Custom Parameter table from the Financial application.

Data Object Name: R_NEG_DRW_PRV_PARM

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
RUN_MODE	Run Mode	Required	Report Only	Valid Values are Report Only and Report and Update.
DATA_SRC	Data Source	Required	Reimbursement Holding	Valid Values are Reimbursement Holding, Temporary Reimbursement Request

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
				and Reimbursement Request. Note: Data Source must be set to Reimbursement Request to run this process in Update Mode.
EXP_REQ_DT	Expected Request Date	Optional	<blank>	Value is required when Data Source is Reimbursement Holding. Used to select records according to CMIA Expected Request date.
PRVN_CTRL	Prevention Control	Required	Department and Appropriation	Valid Values are: Department and Appropriation; Department and Major Program; Department and Program; Assistance Listing Number; Customer; Letter of Credit; Appropriation; Drawdown Group; Department, Major Program and Program Period; or Department, Program and Customer Account.

Data Object Name: R_NEG_DRW_EXCP_DTL

Field Name	Description (Caption)	Required /Optional	Default Value	Comments
EXEP_TYP	Exception Type	Required	Event Type	Valid values are Event Type and Program.
EVNT_TYPE_ID	Event Type	Optional	*	Entered Value should exist in R_EVNT_TYP table. This attribute is required when the exception type is set as "EVENT TYPE."
DEPT_CD	Department	Optional	*	Entered Value should exist in R_DEPT table. This attribute is required when the exception type is set as "PROGRAM."

PROG_CD	Program	Optional	*	Entered combination of Department and Program should exist in R_PROG table. This attribute is required when the exception type is set as "PROGRAM."
---------	---------	----------	---	---

Sort Sequence

The Sort Sequence depends on the Prevention Control parameter. For example, if the Prevention control is set as Department and Program then, the sorting will be done on the basis on Department and Program. If the Prevention control is set as Department, Program and Customer Account then, the sorting will be done on the basis of Department, Program, AR Unit, Customer Code, Billing Profile, Final Billing flag and Minimum Billing Amount.

Selection Criteria

The selection criteria depend on the Data Source selection. If the selected Data Source is Reimbursement Holding then all records from Reimbursement Holding Matching Expected Request Date (that is, attribute CMIA_EXP_REQ_DT) will be considered for processing. Otherwise, all the records from the specified Data Source will be considered for processing.

Problem Resolution

If the process fails due to any reason then perform the corrective actions:

- Check the Log file for any errors that may have occurred while the program was running.
- Re-run the batch job.