CGI Advantage® 4

Conversion Processes Run Sheets Guide

Table of Contents

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:

- 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.
- 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.
- 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.
- The CGI Advantage HRM Payroll Engine System Administration Guide describes the system control tables and utilities for CGI Advantage HRM.
- 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 Conversion Batch and Chain Processes

Descriptions of the Conversion Processes are organized in this section in alphabetical order.

2.1.1 1042-S Journal/Disbursement Conversion

Chain or Job Name	1042-S Journal/Disbursement Conversion
Recommended Frequency	One time use; can be run multiple times for different date ranges.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	None

Overview

The 1042-S Journal/Disbursement Conversion batch job updates 1042-S Tax Rate and IRS Country information on the 1099 Journal records, Posting Lines, and the Vendor and Accounting Lines for the AD and MD Transaction Types. The process only updates records having Vendor/Customer Codes and COA elements that are 1042-S reportable. This conversion process first identifies 1099 Journal records that are 1042-S reportable. Using the eligible 1099 Journal records, the process identifies the related Posting Lines, Accounting Lines, and Vendor Lines. If all criteria for 1042-S eligibility are met, all of the identified records are updated with 1042-S Tax Rate and IRS Country information.

This conversion process is intended for one-time use, but can be executed multiple times with the same or different parameters. However, once the process successfully updates and commits changes to a set of related records, those records will not be reprocessed.

Before running the 1042-S Journal/Disbursement Conversion batch job:

- It is required that you setup your 1042-S eligible vendors and Chart of Accounts before running this process. This includes:
    - Create records on the new 1042-S Reporting Information (1042I) table for TIN/TIN Types that are eligible for 1042-S reporting.
    - Update the corresponding Vendor Customer records to include the appropriate 1042-S information (1042-S Recipient Account Number, 1042-S Recipient Code, IRS Country of Residence and IRS Country Sub Code).
    - Setup the 1042-S Type of Income (1042T) table to associate eligible 1042-S Income Codes, 1042-S Recipient Codes and Types of Income.
    - Update eligible Chart of Accounts (Object, Sub Object, Balance Sheet Account and Sub Balance Sheet Account) to include 1042-S Type of Income information.
        - It is strongly recommended to run Offline Journalization (Journalization Engine) to process as many unjournalized Posting Lines as possible. Because 1042-S Journal/Disbursement Conversion identifies records from the 1099 Journal, it will not update any 1042-S-eligible unjournalized Posting Lines. However, 1042-S Journal/Disbursement Conversion may be run again, after Offline Journalization, to ensure any newly journalized Posting Lines are initialized. When 1042-S Journal/Disbursement Conversion is run multiple times, it simply looks for unprocessed 1042-S eligible records.

Process Steps	Messages

1. Parameter Validation	• Validate Batch Parameters, the following messages may be issued:
	  • Validating the batch parameters.
	  • Invalid Format for 'date'. Should be MM/dd/ccyy
	  • Start Date must be on or before the End Date
	  • 'block size' should be a valid positive integer: 'block size value'
	  • A blank value was supplied for 'block size'
	  • Setting default value for 'block size'
	• Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log.
	• Parameter validation completed.

2. Selection of Records	• Select eligible records, the following message should be issued: "Begin selection of records." • If the selection query does not execute successfully the following message may be issued: "Exception occurred in processJournalEntries() while attempting to retrieve 1099 Journal Entries: 'exception message' " • If the selection returns 0 records, then the following message will be issued: "No eligible records found for 1042-S Journal/Disbursement Conversion". • Selection of Records completed, the following message should be issued: "Selection of records completed."

3. Process Conversion	• Update eligible 1042-S records
	• Verify if the Vendor Code is 1042-S reportable
	• The following additional messages may be issued:
	  • "No records found in the Vendor/Customer table for Vendor/Customer Code: 'n' "
	  • "No matching 1042-S Information found for Recipient Account Number: 'n' "
	  • "The Recipient Account Number 'n' was not marked 1042-S Reportable"
	• Verify if the COA element is 1042-S reportable
	• The following additional messages may be issued:
	  • "IRS Country information not set for Vendor/Customer Code: 'n' on the Vendor/Customer table"
	  • "No record found in the Expense System Options table for Fiscal Year: 'n' "
	• Perform 1099 Journal Updates
	• Perform Transaction Vendor Line Updates
	• Perform Posting Line Updates
	• Perform Accounting Line Updates
	• Process Conversion Completed. The following

Process Steps	Messages
	message may be issued: • "Number of records processed: 'n' "

Major Input

Tables

- 1099 Reporting Journal (J1099, JRNL_1099)
- Vendor/Customer (VCUST, R_VEND_CUST)
- 1042-S Reportable Information (1042I, R_1042S_RPT_INF)
- Object (OBJ, R_OBJ)
- Sub Object (SOBJ, R_SOBJ)
- Balance Sheet Account (BSA, R_BSA)
- Sub Balance Sheet Account (SBSA, R_SBSA)
- 1042-S Income Code/Tax Rate (ICTX, R_1042S_ICTX)
- System Options (SOPT, R_EXP_SOPT)

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
START_DT	Start Date (mm/dd/yyyy). Required	No Default
END_DT	End Date (mm/dd/yyyy). Required	No Default
PROG_CTR_SZ	Progression Message Block Size (Defaulted to 100 if empty) Required	100
SELECT_BLOCK	Select Block Size. If not entered then it is defaulted to 100. 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. Required.	100
COMMIT_BLOCK	Commit Block Size (if not entered then defaulted to 100)	100

Major Output

- Updates to the 1099 Journal Table (JRNL_1099)
- Updates to the Transaction Accounting Table (DOC_ACTG)

- Updates to the Posting Line Table (PSTNG_LN_CAT)
- Updates to the AD Transaction Accounting Line Table (AD_DOC_ACTG)
- Updates to the MD Transaction Accounting Line Table (MD_DOC_ACTG)
- Updates to the AD Transaction Vendor Line Table (AD_DOC_VEND)
- Updates to the MD Transaction Vendor Line Table (MD_DOC_VEND)

Job Return code

The following table shows the potential job return codes for 1042-S Journal/Disbursement Conversion:

Return Code	Condition
Successful (1)	All of the 1099 Journal entries are processed successfully.
Warning (4)	No eligible records found for 1042-S Journal/Disbursement Conversion. This could be because of the following reason: • There are no journal entries without an IRS Country Code with a Vendor Customer record having a recipient account number and recipient code. The Recipient Account Number 'n' was not marked 1042-S Reportable.
Non Fatal Error (8)	This could be because of the following reasons: • No records found in the Vendor/Customer table for Vendor/Customer Code: 'n' • No matching 1042-S Information found for Recipient Account Number: 'n' • No record found in the Expense System Options table for Fiscal Year: 'n' • IRS Country information not set for Vendor/Customer Code: 'n' on the Vendor/Customer table
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Failed to commit the transaction. • 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

The 1099 Journal records are sorted by the following fields to process updates to the transactions:

- Vendor/Customer Code
- Fiscal Year

Selection Criteria

- The 1042-S Journal/Disbursement Conversion job selects 1099 Journal entries based on the following selection criteria:
- Transaction Record Date is less than or equal to the End Date parameter value.
- Transaction Record Date is greater than or equal to the Start Date parameter value.
- Transaction Type is either 'AD' or 'MD'.
- The 1099 Journal IRS Country Code is empty.
- The 1099 Journal Vendor/Customer code has a Recipient Account Number and Recipient Code on the Vendor/Customer table.

Problem Resolution

There is no need to back out any updates if this job fails in any of the steps since the job can be rescheduled to process the rows that may have been missed.

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)	Invalid Format for 'n'. Should be mm/dd/ccyy **Sample Message:** Invalid Format for Start Date. Should be mm/dd/ccyy	Enter the required date value in the format mentioned, for example: 01/01/2008	
	Start Date must be on or before the End Date.	Enter a Start Date that occurs on or before the End Date.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	'n' should be a valid positive integer: 'I' **Sample Message:** Progression counter size should be a valid positive integer: -900	Make sure the block size provided is a positive integer.	
	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before rescheduling the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can be 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 be rescheduled.	

Step 2: Selection of records

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	No eligible records found for 1042-S Journal/Disbursement Conversion.	Make sure all of the 1042-S information has been entered for the records that were expected to be selected.	Alternatively, the job can be rescheduled with a different set of parameters.
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.	In this step the job can fail with fatal conditions only on encountering unknown exceptions. If that happens, investigate the	

Possible Return Codes	Condition	Recommendation	Other Instructions
		exception reported by the process, resolve the error and reschedule the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can be 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 be rescheduled.	

Step 3: Process Conversion

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	The Recipient Account Number ‘n’ was not marked 1042-S Reportable.	Make sure the recipient account number on the 1042I page has the appropriate 1042-S Reportable setting.	
Non Fatal Error (8)	No records found in the Vendor/Customer table for Vendor/Customer Code: ‘n’	Verify that the desired entry should be updated. If this record needs to be updated, the vendor customer record will need to be entered for the corresponding Vendor/Customer code.	
	IRS Country information not set for Vendor/Customer Code: ‘n’ on the Vendor/Customer table.	Verify the Vendor/Customer Code on the VCUST record has IRS Country information entered, if desired.	
	No matching 1042-S Information found for Recipient Account Number: ‘n’.	Verify the proper data has been entered in the 1042I page for this account number.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	No record found in the Expense System Options table for Fiscal Year: 'n'.	Verify there's an entry on Expense SOPT for the FY provided.	
Failed (12)	Failed to commit the transaction.	If this happens, investigate the exception reported by the process, resolve the error and reschedule the job.	
	Failed because of runtime exceptions for an unexpected situation.	In this step the job can fail with fatal conditions only on encountering unknown exceptions. If that happens, investigate the exception reported by the process, resolve the error and reschedule the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can be 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 be rescheduled.	

2.1.2 ADVFN00011655 Data Fix - Fix Referenced Liquidated Amount for Half Postings

Chain or Job Name	ADVFN00011655 Data Fix - Fix Referenced Liquidated Amount for Half Postings
Recommended Frequency	Once. Refer to the "Scripts to be run before executing the process" and "Important Instructions on execution of batch process" sections under "Overview" for important instructions that need to be followed before the execution of the batch process.
Single Instance Required	Only a single instance can be run at a time.
Can be restarted?	The job cannot be restarted.
Reports generated	Report On Corrupted Referenced Liquidated Amount is generated.

Overview

The Referenced Liquidated Amount on the Accounting Line (RFED_LQD_AM) stores the total amount that the current AL liquidates another AL and also equals to the sum of the Liquidation Posting Line's Posting Amount. This data fix program will identify an incorrect value of the Referenced Liquidated Amount (RFED_LQD_AM) for Accounting lines that have half posting liquidation lines. Accounting lines for Final non-cancelled transactions are fixed. Half posting Standard lines are generated for a transaction when any of the Posting pairs on the Event type has any of the following conditions satisfied:

1. Debit and Credit posting codes are defined and both are non-offset.
2. Debit posting code is defined and is non-offset and Credit posting code is empty.
3. Credit posting code is defined and is non-offset and Debit posting code is empty.

In all of above cases, the value of the Offset Posting Code (OPSTNG_CD_ID) is empty for the generated half posting lines (PSTNG_LN_CAT). Any transaction referencing these Half posting Standard lines will generate half posting liquidation lines.

Once the program identifies Accounting lines that have half posting liquidation lines, it obtains the existing value of RFED_LQD_AM from the Accounting line. Then it computes the correct value for RFED_LQD_AM. The correct value of RFED_LQD_AM is equal to the sum of the Line amount (LN_AM) of Liquidation lines belonging to the Accounting line for the Final transaction. It then compares the existing value of RFED_LQD_AM with the computed/correct value of RFED_LQD_AM. If the values only differ in sign then the program will report this line with a message indicating that the Accounting line will be fixed with a correct value of RFED_LQD_AM once it has been run in update mode. If the values differ in absolute values then the program will report this line with a message indicating that the Accounting line cannot be fixed and such lines need to be researched further to figure out the correct Referenced Liquidated Amount on the Accounting line (RFED_LQD_AM) and these lines will not be fixed by the program.

Scripts to be run before executing the process

1. Pre-scripts need to be run before running the batch process to create a temporary index on the Posting Line Catalog (PSTNG_LN_CAT) on its DOC_PHASE_CD and LN_FUNC_CD fields. This temporary index is required for efficient and faster running of

the process. These pre-scripts have already been included in the database CD. Before running the batch process, please ensure that this temporary index is created. Depending on the database being used, execute the following pre-script file:

a. Pre_DF11655_XX_DDL.sql

Where XX is ORA for Oracle database

XX is DB2UDB for DB2 database

XX is SQLSRVR for Microsoft SQL Server database

XX is DB2zOS for DB2 ZOS database

2. Post scripts need to be run only after the successful run and verification of the job in the update mode. These scripts should only be run when the batch process is not intended to be run further. These scripts will drop the temporary index created by pre-scripts on the Posting Line Catalog (PSTNG_LN_CAT). Depending on the database being used, execute the following pre-script file:

a. Post_DF11655_XX_DDL.sql

Where XX is ORA for Oracle database

XX is DB2UDB for DB2 database

XX is SQLSRVR for Microsoft SQL Server database

XX is DB2zOS for DB2 ZOS database

Important Instructions on execution of batch process

- Before Job Run:

Before this batch job can be run the system must be brought down and then brought up again. System here implies all application servers hosting the application or Job Managers. There should not be any online or offline activities going on in the system. There should not be any other Batch application servers polling other than the ones meant to run this process batch job. The System would only be running this job and there should not be any other activities going on.

It is highly recommended to run the data fix in report mode before running in update mode for the batch process to verify that transactions selected by the data fix to be fixed have in fact corrupted the Referenced Liquidated Amount on the Accounting line (RFED_LQD_AM).

Run Modes

1. Report mode:

The program can be run in Report mode to verify the identified corrupted values of RFED_LQD_AM on Accounting lines. Report mode does not do any updates; it only identifies Accounting lines with incorrect RFED_LQD_AM and reports them. Once the results are verified, then the program can be run in Update mode to rectify the incorrect RFED_LQD_AM on the identified Accounting lines. Details on how to verify the results are listed under section "Verification of amounts".

2. Update Mode:

The job must be run in this mode only after it has been run in the report mode and after successful verification of the corrupt records. When the program is run in Update mode it

updates the RFED_LQD_AM with the correct computed value of RFED_LQD_AM on the identified Accounting lines and a report is generated listing the Accounting lines that it has fixed. Please note that the Accounting lines that were displayed with a message that they will not be fixed, will not be fixed or updated when run in Update mode.

Process Steps	Messages
1. Parameter Validation	If the parameter is invalid, an error message will be displayed.
2. Selection of Records	• If an Accounting line is not found on the Transaction Specific Accounting table for the given liquidation posting line then a message is displayed to inform the same. • Once all of the corrupted records have been found, the number of Accounting lines found to be corrupted and displayed on the report is listed as a message.

Restartability information

When the job ends with a Return Code other than Successful, then restarting the job is not an option. The reason of job failure needs to be investigated by looking at the job error log and error log files.

Major Input

Tables

- Posting line catalog (PSTNG_LN_CAT)
- Event Type (R_EVNT_TYP)
- Transaction Accounting Catalog (DOC_ACTG)

Batch Parameters

Parameter	Description	Default Value
Client Name	Client Name to be displayed on the Report. This parameter value is optional and is overrideable.	
Commit Block Size	Commit Block Size. If a value is not entered then a value of 1000 is defaulted. This gives the number of processed lines after processing, where a commit is issued. This is a performance tuning parameter. The value can be tuned as per system configuration and its processing capabilities. Too low of a value can increase the number of database commits and	1000

	too high of a value can cause out of memory issues.	
Purge ADVFN00011655_AL Table (1 - NO, 2 - YES)	Purge ADVFN00011655_AL table. This is a required overrideable parameter. Each run of this program, irrespective of its run modes, inserts Accounting lines that have incorrect RFED_LQD_AM into ADVFN00011655_AL table. For the current run, if existing accumulated records added by previous runs have to be purged first before inserting new records into ADVFN00011655_AL, then set the Purge table parameter value as 2 (YES).	1
Run mode (1-Report, 2-Update)	Run Mode. This parameter value is required. Valid run modes are 1 – Report Mode and 2-Update mode.	1

Major Output

Tables

- ADVFN00011655_AL
- The program identifies (in report and update modes) and fixes (only in update mode) corrupted Referenced Liquidated Amount field values on the following transaction components:
    1. ABS Transaction Accounting Component
    2. AD Transaction Accounting Component
    3. ARE Transaction Accounting Component
    4. CH Transaction Accounting Component
    5. CI Transaction Accounting Component
    6. CL Transaction Accounting Component
    7. CR Transaction Accounting Component
    8. DC Transaction Accounting Component
    9. Transaction Accounting Catalog
    10. Transaction Commodity Catalog
    11. FA Transaction Accounting Component
    12. IET Transaction Accounting Component
    13. ITA Transaction Accounting Component
    14. ITI Transaction Accounting Component
    15. MA Transaction Accounting Component
    16. MA Transaction Commodity Component
    17. MA Transaction Header
    18. MA Transaction Vendor Component
    19. MD Transaction Accounting Component

20. PO Transaction Accounting Component
21. PO Transaction Commodity Component
22. PO Transaction Header
23. PO Transaction Vendor Component
24. PR Transaction Accounting Component
25. PYRL Transaction Accounting Component
26. RE Transaction Accounting Component
27. RQ Transaction Accounting Component
28. RQ Transaction Commodity Component
29. RQ Transaction Commodity Group Component
30. RQ Transaction Header
31. WO Transaction Accounting Component
32. ABS Transaction Accounting Component
33. IET Transaction Accounting Component
34. ITA Transaction Accounting Component
35. PYRL Transaction Accounting Component

- Report On Corrupted Referenced Liquidated Amounts

  Fields displayed on the Report
  The following information will be displayed on the report regarding Accounting lines with an incorrect RFED_LQD_AM.

  - Transaction Code
  - Transaction Department Code
  - Transaction Id
  - Transaction Version Number
  - Transaction Vendor Line Number
  - Transaction Commodity Line Number
  - Transaction Accounting Line Number
- Existing Referenced Liquidation Amount
- Calculated Referenced Liquidation Amount - During update mode this value will be used to update and fix RFED_LQD_AM for Accounting lines reported as “will be fixed”.
- Comments – This describes if the identified Accounting line with an incorrect RFED_LQD_AM will be fixed or not.

Chain / Job Return code

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

Return Code	Condition
Successful (1)	Corrupted Accounting lines were successfully reported / fixed.
Warning (4)	If an Accounting line is not found on the Transaction Specific Accounting table for the given liquidation posting line then the job continues execution but ends in a job return code of Warning. Necessary details of the line are logged in the job log.
Failed (12)	This return code will be issued under several conditions: • Parameter validation failed. • 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

Posting lines fetched from the Posting line catalog (PSTNG_LN_CAT) are sorted by the Transaction Code, Transaction Department Code, Transaction Id, Transaction Version Number, Transaction Vendor Line Number, Transaction Commodity Line Number and Transaction Accounting Line Number.

Selection Criteria

The program selects all of the records from the Posting Line Catalog (PSTNG_LN_CAT) that satisfies all of the following conditions:

1. Offset Posting Code (OPSTNG_CD_ID) is empty (this condition selects only half posting lines)
2. Transaction Phase Code is Final
3. Posting Line Function Code is Liquidation
4. Reference Type is (Partial, Final)
5. Transaction Function Code is not equal to Cancelled

Problem Resolution

The following table shows the possible return codes and recommendations.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Warning (4)	Job ended with a Warning because an Accounting line is not found on the Transaction Specific Accounting table for the given liquidation posting line. **Sample Message:** Transaction Accounting component not found for Transaction Code GAP, Transaction Dept Code 010, Transaction ID DOC1, Transaction Vers No 1 and Transaction Actg Internal Line No 1	Occurrence of this condition is highly unlikely. However, if this occurs then investigate for the missing Accounting line on the Transaction Specific Accounting table.	N/A
Failed (12)	The job can fail under the following two conditions: 1. Required Parameters are not entered. **Sample Message:** Run Mode Value is required. 2. Run time exceptions for unexpected situations.	For Condition 1, enter valid parameters and run the job again. For Condition 2, investigate the cause of the issue by looking at the job log and error log files.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	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. Schedule a new job.	N/A

Verification of amounts

Scenario 1: Event type ID with both Debit posting code side and Credit posting code side values defined and both as non-offset.

Event type PR07 with:

Posting Pair A

Debit Posting Code: P001

Debit Offset: false

Credit Posting Code: P002

Credit Offset: false

1. A CT was processed using Event type ID PR07 for $100. Two half posting lines created (shown 1st below) for the single accounting line (shown 2nd). Samples below are before the reference so the subsequent updates by the referencing transaction can be seen.

SQL for Commodity Based Transactions:

Select doc_cd, doc_dept_cd, doc_id, doc_vers_no, doc_vend_ln_no, doc_comm_ln_no, doc_actg_ln_no, ln_am, pstng_am, pstng_ln_clsd_am, ln_func_cd, pstng_cd_id from pstng_ln_cat where doc_cd = 'XX' and doc_dept_cd = 'XXX' and doc_id = '##' and doc_vend_ln_no = '#' and doc_comm_ln_no = # and doc_actg_ln_no = # order by doc_cd, doc_vers_no

SQL for Accounting Based Transactions:

Select doc_cd, doc_dept_cd, doc_id, doc_vers_no, doc_vend_ln_no, doc_comm_ln_no, doc_actg_ln_no, ln_am, pstng_am, pstng_ln_clsd_am, ln_func_cd, pstng_cd_id from pstng_ln_cat where doc_cd = 'XXX' and doc_dept_cd = 'XXX' and doc_id = '##' and doc_vend_ln_no = # and doc_comm_ln_no = 0 and doc_actg_ln_no = # order by doc_cd, doc_vers_no

Doc Cd	Doc Dept Cd	Doc Id	Doc Vers No	Doc Vend Ln No	Doc Comm Ln No	Doc Actg Ln No	Ln Am	Pstng Am	Pstng Ln Clsd Am	Ln Func Cd	Pstng Cd Id
CT	010	12220500000000000001	1	1	1	1	-100	-100	0	1	P002
CT	010	12220500000000000001	1	1	1	1	100	100	0	1	P001

SQL for Commodity Based Transactions:

Select doc_cd, doc_id, doc_vers_no, doc_vend_ln_no, doc_comm_ln_no, doc_actg_ln_no, ln_am, al_clsd_am, open_am, rfed_ln_am, rfed_lqd_am from doc_actg where doc_cd = 'RQS' and doc_dept_cd = 'MG' and doc_id = '##' and doc_vend_ln_no = '0' and doc_comm_ln_no = '1' and doc_actg_ln_no = '1' order by doc_cd, doc_vers_no, doc_vend_ln_no, doc_comm_ln_no, doc_actg_ln_no

SQL for Accounting Based Transactions:

Select doc_cd, doc_id, doc_vers_no, doc_vend_ln_no, doc_comm_ln_no, doc_actg_ln_no, ln_am, al_clsd_am, open_am, rfed_ln_am, rfed_lqd_am from doc_actg where doc_cd = 'XXX' and doc_dept_cd = 'XXX' and doc_id = '##' and doc_vend_ln_no = '#' and doc_comm_ln_no = '0' and doc_actg_ln_no = '#' order by doc_cd, doc_vers_no, doc_vend_ln_no, doc_comm_ln_no, doc_actg_ln_no

Doc Cd	Doc Id	Doc Vers No	Doc Vend Ln No	Doc Comm Ln No	Doc Actg Ln No	Ln Am	Al Clsd Am	Open Am	Rfed Ln Am	Rfed Lqd Am
CT	12220500000000000001	1	1	1	1	100	0	100	0	0

2. A PRC referencing the CT was processed for $100 with a reference type of Final. When the PRC went to Final, two Liquidation Posting lines (LN_FUNC_CD = 3) were generated for the CT. Shown below are the posting lines and accounting lines for both transactions. Note that the Referenced Liquidated Amount (RFED_LQD_AM) is $100.00 on the PRC accounting line when it would have been -$100.00 if the event type on the CT had produced a single, combined posting line. This data fix program will correct that amount to be a negative $100.00.

Posting Lines

Doc Cd	Doc Dept Cd	Doc Id	Doc Vers No	Doc Vend Ln No	Doc Comm Ln No	Doc Actg Ln No	Ln Am	Pstng Am	Pstng Ln Clsd Am	Ln Func Cd	Pstng Cd Id
PRC	010	1222050000000000003	1	1	1	1	-100	-100	0	3	P001
PRC	010	1222050000000000003	1	1	1	1	100	100	0	3	P002
PRC	010	1222050000000000003	1	1	1	1	100	100	0	1	D011

Doc Cd	Doc Dept Cd	Doc Id	Doc Vers No	Doc Vend Ln No	Doc Comm Ln No	Doc Actg Ln No	Ln Am	Pstng Am	Pstng Ln Clsd Am	Ln Func Cd	Pstng Cd Id
CT	010	12220500000000000001	1	1	1	1	-100	-100	-100	1	P002
CT	010	12220500000000000001	1	1	1	1	100	100	100	1	P001

Accounting Lines

Doc Cd	Doc Id	Doc Vers No	Doc Vend Ln No	Doc Comm Ln No	Doc Actg Ln No	Ln Am	Al Clsd Am	Open Am	Rfed Ln Am	Rfed Lqd Am
PRC	12220500000000000030	1	1	1	1	100	0	100	0	100

Doc Cd	Doc Id	Doc Vers No	Doc Vend Ln No	Doc Comm Ln No	Doc Actg Ln No	Ln Am	Al Clsd Am	Open Am	Rfed Ln Am	Rfed Lqd Am
CT	12220500000000000001	1	1	1	1	100	100	0	100	0

- When the batch process is run in Report mode, the report displays PRC transaction information with Existing Referenced Liquidation Amount as $100 and Calculated Referenced Liquidation Amount as -$100.
- When the batch process is run in Update mode then RFED_LQD_AM on PRC accounting line is updated to the correct value, i.e., Calculated Referenced Liquidation Amount (-$100).

Here is logic based on which Calculated/Correct Referenced Liquidation Amount is computed for Accounting lines those have half posting lines:

If the Event type ID used on the Accounting line has both Debit and Credit Posting codes values defined and both are non-offset then sum the Line amount (LN_AM) of Liquidation lines those have Posting code equal to Debit Posting code. For the above CT this condition is used.

If the Event type ID used on the Accounting line has either Debit or Credit Posting codes values defined and defined posting code (Debit or Credit) is non-offset then sum the Line amount (LN_AM) of Liquidation lines those have Posting code equal to defined non-offset Posting code.

Verification of Transactions Where Cause of Discrepancy in Unknown

- If an Event Type existed with a posting pair with two non-offset posting codes that were subsequently removed, there may be entries on the report with this error message with a calculated referenced liquidation amount equal to $0. These do not have to be fixed. They were identified by the job and added to the report because of the change to the Event Type table.
- If a transaction listed with this message can be modified or cancelled, then this line must be closed out. To do that, modify the accounting line amount down to equal the closed amount. Then insert a new accounting line for what was open. There should be no liquidation posting.

If there is, there will have to be an adjustment to the referenced transaction to close out any amount re-opened.

- For any transactions on the report with this error message that cannot be modified or cancelled, there is no action required. This may include AD and MD transaction codes, for example. As once these have cleared the bank, modifications and cancellations are not allowed.

2.1.3 ADVFNAM14421 Data Fix – Fix to Insert & Update records in Fixed Asset Registries

Chain or Job Name	ADVFNAM14421 Data Fix - Fix to Insert & Update records in Fixed Asset Registries
Recommended Frequency	This job is required to run only once to correct the invalid data on Fixed Asset Registry tables
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

Fixed Asset Registry tables are like Central Repository for Advantage Fixed Assets. It contains Header, Component and Accounting level information for both Acquisitions and Betterment documents. This data fix program fixes multiple issues that occurred on Fixed Asset Registry tables.

This data fix will first identify all the records that are present in FA Document Accounting line (FA_DOC_ACTG), Fixed Asset Header table (R_FAR_HDR) and Fixed Asset Component table (R_FAR_COMP) but not present in Fixed Asset Accounting table (R_FAR_ACTG) inserts the missing records in Fixed Asset Accounting Registry table (R_FAR_ACTG). This data fix will only select the records; the entire process of inserting the missed records in the Fixed Asset Accounting Registry table (R_FAR_ACTG) is taken care of by redirecting to existing code.

Once this process inserts the missing records in the Fixed Asset Accounting Registry table (R_FAR_ACTG), it also automatically updates all the relevant information on the Fixed Asset Header Registry table (R_FAR_HDR) and Fixed Asset Component Registry table (R_FAR_COMP).

Process Steps	Messages
1. Parameter Validation	• Parameters are not necessary for this job.
2. Selection of Records	• This job selects all the records that are present on Fixed Asset Header table (R_FAR_HDR) and Fixed Asset Component table (R_FAR_COMP) but not present in Fixed Asset Accounting table (R_FAR_ACTG) • If the above selection returns records, then the following message is printed in the logs: *"Records with FA Number: FA_NO and Component Number: COMP_NO are picked up by the job*" • Once the records are inserted in Fixed Asset Accounting Registry table (FARACTG) then the following message is printed in the logs: *"Fixed Asset Registry Accounting (FARACTG) entries are added for FA Number : FA_NO and Component Number : COMP_NO"*

Restartability information

When the job ends with a Return Code other than Successful, then restarting the job is not an option. The reason for job failure should be investigated by looking at the job error log and error log files.

Major Input

Tables

- FA Document Accounting line (FA_DOC_ACTG)
- Fixed Asset Header Table (R_FAR_HDR)
- Fixed Asset Component Table (R_FAR_COMP)
- Fixed Asset Accounting table (R_FAR_ACTG)

Major Output

Tables

- Fixed Asset Header Table (R_FAR_HDR)
- Fixed Asset Component Table (R_FAR_COMP)
- Fixed Asset Accounting table (R_FAR_ACTG)

Standalone / Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected FA Document records are inserted

Return Code	Condition
	successfully.
Failed (12)	The job will fail under the following conditions: • When there are any application or code level issues for the selected FA documents
Terminated (16)	This Return Code is issued when user terminates the job.
System Failure (20)	This Return Code is issued when the job is terminated because of database server or network issues.

Sort Criteria

There is no sort criteria applied for this job.

Selection Criteria

This job selects all the records that are present on Fixed Asset Header table (R_FAR_HDR) and Fixed Asset Component (R_FAR_COMP) table but not present in the Fixed Asset Accounting table (R_FAR_ACTG).

Problem Resolution

The following table shows the possible return codes and recommendations.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	When all the selected records inserted and updated successfully.	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non Fatal Error (8)	N/A	N/A	N/A
Failed (12)	When any application or code level issues are encountered by the system	N/A	N/A
Terminated (16)	When the Job is manually terminated by the user.	The reason for the termination needs to be investigated. The job can be scheduled again	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. The job can be scheduled again	N/A

2.1.4 ADVFNAM27365_DataFix - EV Bid Tabulation

Job Name	ADVFNAM27365_DataFix - EV Bid Tabulation
Recommended Frequency	This job is required to run only to insert records in the Bid tabulation component for all the Evaluation (EV) type transactions that were created prior to the implementation of the Bid Tabulation Pivot table in Advantage 4.
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	No

Overview

This conversion job inserts record in the Bid tabulation component on the application as well as in the EV_DOC_BDTAB table for all the Evaluation (EV) type transactions that were created prior to the implementation of the Bid Tabulation Pivot table in Advantage 4.

This batch job will process all those Evaluation (EV) type transactions that currently have one dummy line present in the EV_DOC_BDTAB table, depending on the DOC_CD provided in the batch job parameters for the EV type transactions.

Assumptions

- The Bid Tabulation Pivot table exists for the Evaluation (EV) type transactions in the Financial application.
- Before running the Data Fix batch job, please take backup of the EV_DOC_BDTAB table.
- Evaluation (EV) transactions should run before the Evaluator (EVT) transactions. If the user tries to run the batch job for the EVT transactions, before running the job for its corresponding EV type transactions, no lines are inserted on the Bid tabulation component for the EVT transaction, and the user will get an error in the job logs stating that the corresponding EV type transaction has not been processed yet.

Restartability Information

This job does not support restartability. If the job fails for any reason, a new job must be scheduled.

Major Input

- EV Type Transaction codes (DOC_CD)
- EV Type Transaction Department Codes (DOC_DEPT_CD)

Parameters

Batch Parameters

Description (Caption)	Parameter Name	Default Value
EV Type Transaction Codes	Transaction Code	No Default.

	DOC_CD	• This is a mandatory parameter value on this job; therefore, an error is issued if the mandatory parameter value is not provided in the job. • Supports only EV type transactions. • This parameter allows a comma delimited list of EV Transaction Type - Transaction Codes to be selected for updating the Bid Tabulation component (for example, EV, GEV and so forth.)
EV Type Transaction Department Codes	Transaction Department DOC_DEPT_CD	No Default. • This is an optional parameter value on this job. • If provided, system updates EV type transactions only corresponding to a particular transaction department code. • This parameter allows a comma delimited list of EV Transaction Department Codes to be selected for updating the Bid Tabulation component.
Commit block size	COMMIT_BLOCK	1000 This is an optional parameter value on this job.

Major Output

This data fix inserts record in the Bid Tabulation component for all the Evaluation (EV) type transactions that were created prior to the implementation of Bid Tabulation Pivot table in Advantage 4.

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected EV Transaction Code and Transaction Department Codes records are updated successfully.
Warning (4)	No eligible records found.
Failed (12)	The job may fail under the following conditions: • When the mandatory parameter (EV Type Transaction code) value is not provided in the job.

Return Code	Condition
	• When the transaction type provided on the mandatory parameter (EV Type Transaction code) is other than the EV Transaction Type. • Parameters are invalid. • Run time exceptions for unexpected situations.
Terminated (16)	This return code is issued when the job is terminated because of database server or network issues.
System Failure (20)	This return code is issued when the job is terminated by the user.

Selection Criteria

This job updates all those Evaluation (EV) type transactions that currently have one dummy line present in the EV_DOC_BDTAB table, depending on the DOC_CD provided in the batch job parameters for the EV type transactions, which needs to be updated.

2.1.5 ADVFNAM29166_Datafix - Vendor Verification Password Update

Job Name	ADVFNAM29166_Datafix - Vendor Verification Password Update
Recommended Frequency	One Time
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	No

Overview

This conversion job should be run to set the Vendor Verification Password in R_PNT_VEND_CUST and R_VEND_CUST for vendors where Vendor Verification Password Type is *Use My TIN* and Vendor Verification Password is *null* or *blank*.

This batch job will fetch the TIN Number from Headquarter Account (R_PNT_VEND_CUST) and Location Vendor (R_VEND_CUST) and will update the Vendor Verification Password (VSS_PASS) and Confirm Verification (VEND_PSWD_VERIFY) fields on Headquarter and Location vendors where Vendor Verification Password Type is *Use My TIN*.

The batch job fetches vendor records where Vendor Verification Password Type is *Use My TIN* and Vendor Verification Password is null or blank. It will update the Vendor Verification Password (VSS_PASS) and Confirm Verifications (VEND_PSWD_VERIFY) fields as follows:

1. If the Headquarter and Location vendor has a shadow key (PARITY_STR), it will generate the vendor verification details from the TIN Number and use the existing shadow key to encrypt the vendor verification details. The list of vendors that is updated through this step will be found in the VendDetUpdWithTin.txt file and this can be found in the Export Import Directory.
2. If the Headquarter and Location vendor does not have a shadow key (PARITY_STR), a new key will be generated and it will then generate the vendor verification details from the TIN Number and use the existing shadow key to encrypt the vendor verification details. The list of vendors that is updated through this step will be found in the VendDetUpdWithTin.txt file and this can be found in the Export Import Directory.

Restartability Information

This job does not support the ability to restart. If the job fails for any reason, a new job must be scheduled.

Process Steps	Messages
1. Parameter validation	• Validating Batch Parameters • If the parameter is invalid, the log message will show the error statement. • Batch Parameter validation completed
2. Update Record	Vendor verification details will be updated in Headquarter and Location vendor if the Vendor Verification Password Type is *Use My TIN* and Vendor Verification Password is *null* or *blank*
3. Generate text files	VendDetUpdWithTin.txt file will be created with the records that got updated through this process.

Major Input:

1. VSSSchemaForSA13 parameter value (i.e. value is VSS schema name) in the ADV30Params.ini file
2. Update grant access to VSS schema

   Queries need to run on the VSS schema. To provide the updates, grant access:

   GRANT SELECT, UPDATE ON R_VEND_CUST to <<FIN schema name>>
   /

   GRANT SELECT, UPDATE ON R_PNT_VEND_CUST to <<FIN schema name>>
   /
3. TIN Number from Headquarters (R_PNT_VEND_CUST) and Location (R_VEND_CUST)

Parameters

Batch Parameters

Description (Caption)	Parameter Name	Default Value
Export Import Directory Required Parameter. The export import directory is the place where the output text file generated will be maintained.	AMSEXPORT	$$AMSROOT$$/ExportImport
Commit block size	COMMIT_BLOCK	1000
Select block size	SELECT_BLOCK	1000

Major Output

1. Updates Vendor Verification Password details in R_PNT_VEND_CUST and R_VEND_CUST for Vendors where Vendor Verification Password Type is *Use My TIN* and Vendor Verification Password is *null* or *blank*.
2. Generates the VendDetUpdWithTin.txt file.

Selection Criteria

Update the Vendor Verification Password in R_PNT_VEND_CUST and R_VEND_CUST for Vendors when Vendor Verification Password Type is *Use My TIN* and Vendor Verification Password is *null* or *blank*.

2.1.6 Cancelled Inverse Transactions Correction

Chain or Job Name	Cancelled Inverse Transactions Correction
Recommended Frequency	Once.
Single Instance Required	Only a single instance can be run at a time.
Can be restarted?	The job can be restarted.
Reports generated	The Records Intended to update are reported in the Report Mode and Records Actually Updated are reported in the Update mode.

Description

General

This Cancelled Inverse Transaction Correction process is a data fix process. Previously, cancellation was not allowed on correction transaction. Now the requirement has changed and correction transactions can be cancelled. Event type processor was coded considering that cancellation is not allowed on correction transactions. Due to this, after canceling the correction transaction, invalid amount was populated on field posting amount of the liquidation posting line of cancellation version of correction transaction. This invalid amount on cancellation transaction will incorrectly update the budget buckets, Fund Balance and BSA Balance. Also the value of field 'posting line closed amount' of posting line of correction referenced transaction is updated incorrectly.

This process creates JV transaction to correct budget updates, Fund Balance, and BSA Balance and also updates the referenced transaction's posting line closed amount.

Typically, this conversion process will be scheduled only once. But if it is run more than once then the process will select the records (posting line catalog) after the last record processed in the previous run. The primary key of the last record processed is stored in parameter 'CAN_INV_DOC_CORR' in the Application Parameter (APPCTRL) table. This parameter should not be modified/deleted for correct results.

Batch Setup – Report Mode

When the job has to be run in the Report Mode, the 2nd and 3rd step in the Chain Job, namely, Load JV and Submit JV, have to be disabl rescheduled as only records with data problems ed. If not done, the 2nd step would fail.

Restartability information

When the job ends with a Return Code other than Successful, then restarting the job is not the only option. The reason of job failure needs to be investigated by looking at the job error log and error log files.

Major Input

Tables

Posting Line Catalog (PSTNG_LN_CAT)

Batch Parameters

Job	Parameter	Description	Default Value
Referenced PL Update	AMSEXPORT (**** Refer to Note:** Assumptions for SWBP on page no. 6)	Export Location	$$AMSEXPORT$$/
	AMSPARM (**** Refer to Note:** Assumptions for SWBP on page no. 6)	Parameter Location	$$AMSPARM$$/
	CAN_DOC_CD	Cancellation Transaction's Code (Required)	GAEC,CEC
	IMPORT_PARM_FL_NM	Import Parameter Filename	RefPstngImport.txt
	JV_DOC_CD	JV Transaction Code (Required)	JVA
	JV_DOC_DEPT_CD	JV Transaction Department Code (Required)	
	JV_DOC_PFX	JV Transaction Prefix	
	JV_XML_FL_NM	JV Transaction XML Filename	RefPstngUpdDoc.xml
	SUBMIT_PARM_FL_NM	Submit Parameter Filename	RefPstngSubmit.txt
	CLIENT_NM	Client Name for Report	
	RUN_MODE	Run Mode. Valid Values: 1 - Report and 2 - Update	1
Load JV	PARM_FILE	Import Parameter Filename	$$AMSPARM$$/RefPstngImport.txt
Submit JV	PARM_FILE	Submit Parameter Filename	$$AMSPARM$$/RefPstngSubmit.txt

Major Output

Tables

- Posting Line Catalog (PSTNG_LN_CAT)
- Report on Records updated

  Fields displayed on the Report

  The following information will be displayed on the report regarding corrupted posting line records

  - Transaction Code
  - Department Code
  - Transaction Id
  - Vendor Line Number
  - Commodity Line Number
  - Accounting Line Number
  - Posting Line Number
  - Referenced Transaction Code
  - Referenced Department Code
  - Referenced Transaction Id
  - Referenced Vendor Line Number
  - Referenced Commodity Line Number
  - Referenced Accounting Line Number
  - Referenced Posting Line Number

Transactions

JV Transactions

Data Setup

Following entries are required in table DARF (Transaction Allowable References) for successful submission of generated JV transactions.

Transaction Type	Transaction Code	Reference Type	Reference Code
JV	<JV Transaction Code>	ABS	GAEC
JV	<JV Transaction Code>	PR	CEC

Note: <JV Transaction Code> is the transaction code which user is going to enter in field JV Transaction Code of job parameters. This code should be of Transaction Type JV. These entries should be removed after successful run of this process.

Chain / Job Return Code

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

Return Code	Condition
Successful (1)	Corrupted Cancelled Correction Transactions were successfully fixed.
Warning (4)	If no corrupted record in found in Posting Line table then the job continues execution but ends in a job return code of Warning. Necessary details are logged in the job log.
Failed (12)	This return code will be issued under several conditions: • Parameter validation failed. • 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

The corrupted cancelled correction transactions are sorted by Transaction Posting Number, Transaction Code, Transaction Department Code, Transaction ID, Transaction Version Number.

Selection Criteria

Selection of records from the Posting Line Catalog table:

- Posting Amount (PSTNG_AM) is zero.
- Transaction Code (DOC_CD) is as per the batch parameter value.
- Transaction Function Code (DOC_FUNC_CD) is ‘Cancellation’
- Reference Type (RF_TYP) is ‘Inverse’
- Line Function Code (LN_FUNC_CD) is ‘Liquidation’

If process is restarted, selection criteria will always include checkpoint information of last processed record. If the last record processed information is found on APPCTRL parameter ‘CAN_INV_DOC_CORR’ and the process is a not restarted the process will selected records after the last processed records.

Checkpoint information consists of the following:

- Transaction Posting Number (DOC_PSTNG_NO)
- Transaction Code (DOC_CD)

- Transaction Department Code (DOC_DEPT_CD)
- Transaction ID (DOC_ID)
- Transaction Version Number (DOC_VERS_NO)

Problem Resolution

The following table shows the possible return codes and recommendations.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	Job ended with a Warning because • No records are selected for processing. Sample Message: No records selected.	Investigate whether there are no such records in the system which needs to be corrected by running the process in Report Mode.	N/A
Failed (12)	The job can fail under the following two conditions: 1. Required Parameters are not entered. **Sample Message:** Run Mode Value is required. 2. Run time exceptions for unexpected situations.	For Condition 1, enter valid parameters and run the job again. For Condition 2, investigate the cause of the issue by looking at the job log and error log files.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated.	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. The job can be restarted if possible.	N/A

2.1.7 Cash and Fund Balance Data Fix

Job Name	Cash and Fund Balance Data Fix
Recommended Frequency	Once
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	Yes

Overview

A defect was found for pending transactions submitted to final or rejected back to draft. If the Fund and or Sub Fund values changed due to Funding Priority changes since going to pending, records were left on the various audit tables used for cash and fund balance tracking.

This Cash-Fund Balance Data Fix job reviews records on the various audit tables (R_FBAL_AUD, R_CBAL_AUD, R_BSA_CBAL_AUD) to determine if the record is for a transaction in pending status with the Fund/Sub Fund and possibly BSA/Sub BSA listed. If this determination finds the record should not be on the audit table, it is deleted by the program and any amount is reverted from the pending amounts of the associated detail balance (R_FBAL, R_CBAL, R_BSA_CBAL) and summary balance (R_FBAL_SMRY, R_CBAL_SMRY) record.

Please note this is not the Cash and Fund Balance Synch Process job found in the Batch Jobs folder under General Accounting used to synchronize the tables when System Assurance 2 reports an out of sync condition.

This job may be run in two modes:

1. Report Mode
2. Report and Update Mode.

The **Report Mode** generates a report listing the records from each of the audit tables that should not exist. The report would display the table key along with the pending amounts for each of the audit table.

The table key would be a concatenation of the keys of the audit table as shown below:

Audit table	Table Key
Fund Balance Audit table (R_FBAL_AUD)	DOC_CD^DOC_DEPT_CD^DOC_ID^FUND_CD^SFUND_CD
Cash Balance Audit table (R_CBAL_AUD)	DOC_CD^DOC_DEPT_CD^DOC_ID^FUND_CD^SFUND_CD
BSA Cash Balance Audit table (R_BSA_CBAL_AUD)	DOC_CD^DOC_DEPT_CD^DOC_ID^FUND_CD^SFUND_CD^BSA_CD^SBSA_CD

The **Report and Update Mode** in addition with generating a report, deletes the eligible records from each of the audit tables and reverts the pending amounts of the associated detailed and summary balance records.

The following table shows the various steps that the Cash and Fund Balance Data Fix job goes through and the messages issued at each step

Process Steps	Messages
Parameter Validation	Validating Batch Parameters: 1. COMMIT_BLOCK_SIZE: ensures that it should be a positive integer. If not, an error is logged. 2. PROG_CTR_SZ: ensures that it should be a positive integer. If not, an error is logged. 3. RUN_MODE: ensures that it is a valid run mode. (1 = Report, 2 = Report and Update). If not, an error is logged.
Process and delete the Audit records	1. Processing records of the audit table. 2. No Eligible Audit Records for the audit table. 3. Number of records processed would be displayed depending upon the progression counter size. 4. Total number of records processed would be displayed.
Revert pending amounts	N/A
Report Generation	N/A

Restartability Information

This job does not support restartability. If the job fails for any reason, a new job must be scheduled. Please see the “Problem Resolution” section for more information.

Major Input

- Fund Balance Audit (R_FBAL_AUD) – Eligible records in the fund balance audit table.

- Cash Balance Audit (R_CBAL_AUD) – Eligible records in the cash balance audit table.
- BSA Balance Audit (R_BSA_CBAL_AUD) – Eligible records in the BSA cash balance audit table.
- Posting Line Catalog (PSTNG_LN_CAT)

Batch Parameters

Description (Caption)	Parameter Name	Default Value
Commit Block Size	COMMIT_BLOCK_SIZE	50
Progression Message Block Size	PROG_CTR_SZ	50
Run Mode. 1 = Report, 2 = Report and Update	RUN_MODE	1

Major Output

- Cash and Fund Balance Data Fix Report – Report containing the eligible records of the audit tables (R_FBAL_AUD, R_CBAL_AUD, R_BSA_CBAL_AUD) which are to be deleted. This is applicable only for the both the Run Modes.
- Audit tables – Audit tables (R_FBAL_AUD, R_CBAL_AUD, R_BSA_CBAL_AUD) which are to be deleted. This is applicable only for the Run Mode 2.
- Detailed and Summary balance tables – Reverting the pending amounts of the associated detailed (R_FBAL, R_CBAL, R_BSA_CBAL) and summary (R_FBAL_SMRY, R_CBAL_SMRY) balance tables of the audit tables. This is applicable only for the Run Mode 2.

Job Return Code

The following table shows the potential job return codes for the Cash and Fund Balance Data Fix job

Return Code	Condition
Successful (1)	• All the records are processed successfully
Warning (4)	• N/A
Non Fatal Error (8)	• N/A
Failed (12)	• Required parameters are not entered. • Entered parameters are invalid. • Runtime Exceptions for unexpected situations.
Terminated (16)	• Job is terminated by the user.
System Failure (20)	• Job is terminated because of database or network issues.

Sort Criteria

Transaction Code, Transaction Department Code, Transaction Id.

Selection Criteria

- All the records of the Audit tables whose associated transactions are not in pending phase or whose key elements combination as shown below are not found on PSTNG_LN_CAT for the transaction identifier on the audit record.

Key element combination for Audit table:

Audit table	Key element combination
R_FBAL_AUD	(FUND_CD, SFUND_CD)
R_CBAL_AUD	(FUND_CD, SFUND_CD)
R_BSA_CBAL_AUD	(FUND_CD, SFUND_CD,BSA_CD, SBSA_CD)

Problem Resolution

If the job fails due to some reason, then schedule a new job to process the records.

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)	All parameters are valid.	N/A	N/A
Warning (4)	N/A	This step doesn't 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:** Run Mode cannot be empty.	Schedule a new job and enter a valid Run Mode.	N/A
	Entered Parameters are not valid. Sample Message: Invalid Run Mode. Valid run	Schedule a new job and enter a valid Run Mode.	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
	modes are 1. Report Only and 2. Report and Update Mode. Commit Block Size should be a positive integer	Schedule a new job and enter a positive integer for Commit Block Size.	
	Failed because of runtime exceptions for an unexpected situation.	The failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	The job is terminated manually by the user.	Investigate the reason for the termination, resolve it and schedule a new job with the same set of parameters.	
System Failure (20)	When the job is terminated because of database server or network issues.	Investigate the reason for the termination, resolve it and schedule a new job with the same set of parameters.	

Step 2: Process and delete the Audit records: This step will be performed only if the parameter validation step is successful.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Processing and deleting audit records are 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)	Run time exceptions for unexpected situations. Sample Message: Batch Process Failed. An unexpected exception was encountered. Please contact the system administrator for more	Schedule a new job.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	information.		
Terminated (16)	Job is terminated manually by the user.	Investigate the reason for the termination, resolve it and schedule a new job with the same set of parameters.	
System Failure (20)	When the job is terminated because of database server or network issues.	Investigate the reason for the termination, resolve it and schedule a new job with the same set of parameters.	

Step 3: Revert the pending amounts of the detailed and summary balance records: This step will be performed only if the parameter validation step is successful.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Reverting the pending amounts of the associated detailed and summary records is 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)	Run time exceptions for unexpected situations. Sample Message: Batch Process Failed. An unexpected exception was encountered. Please contact the system administrator for more information.	Schedule a new job.	
Terminated (16)	Job is terminated manually by the user.	Investigate the reason for the termination, resolve it and schedule a new job	

Possible Return Codes	Condition	Recommendation	Other Instructions
		with the same set of parameters.	
System Failure (20)	When the job is terminated because of database server or network issues.	Investigate the reason for the termination, resolve it and schedule a new job with the same set of parameters.	

Step 4: Report Generation: This step will be performed only if the parameter validation step is successful.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Generation of report is 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)	Run time exceptions for unexpected situations. Sample Message: Batch Process Failed. An unexpected exception was encountered. Please contact the system administrator for more information.	Schedule a new job.	
Terminated (16)	Job is terminated manually by the user.	Investigate the reason for the termination, resolve it and schedule a new job with the same set of parameters.	
System Failure (20)	When the job is terminated because of database server or network issues.	Investigate the reason for the termination, resolve it and schedule a new job with the same set of parameters.	

2.1.8 Convert Lifecycle Inquiry Transaction Process

Description

This process creates a transaction reference record in the R_DOC_RF table to the originating transaction itself, with SPEC_RF_NM as “Originates”, and ‘DOC_RF_TYP” as “Special”.

When to Run

This process will be run after Database Upgrade Scripts for the build are executed.

Before running this process,

3. The Database administrator should disable any trigger which is used to populate TBL_LAST_DT on R_DOC_RF table.

Major Input

- DOC_HDR
- R_DOC_RF

Output

A new record will be created in R_DOC_RF table for transactions in Final or Historical Final states, and whose transaction types are one of the following:

"ABS", "AD", "ARE","CH","CI","CL","CR","DC","EV","FA","IET","IN","ITA", “ITI","JV","MA","MD","OC","PE", "PO", "PR", "RC", "RE", "RN", "RO", "SN", "SO", "SR","SRO", "TI","TM","TR","WO"

Parameters

Description (Caption)	Parameter Name	Default Value
Transaction Types in the form of comma separated list. If not provided, the process will quit. This parameter specifies for which transaction types this process will convert transaction references.	DOC_TYP	
From Date (MM/DD/YYYY). This parameter is optional. The process will only convert transactions whose submit date is later than the From Date.	FROM_DATE	
To Date (MM/DD/YYYY). This parameter is optional. The process will only convert transactions whose submit date is earlier than the To Date.	TO_DATE	

Information on Parameters

- From Date should be earlier than To Date.

Sort Sequence

N/A

Selection Criteria

1. Get all the records from DOC_HDR whose transaction type matches the ones given as the job parameter, and DOC_LAST_DT later than From date, and earlier than To date.
2. For each transaction record,
    - Select TBL_LAST_DT from the R_DOC_RF table if at least one row exists in R_DOC_RF table whose DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO match the transaction identifier.
    - Create a transaction reference for the transaction
        - If corresponding rows exist in the R_DOC_RF table, use the minimum TBL_LAST_DT of the R_DOC_RF records as the TBL_LAST_DT for the new reference record.
        - If corresponding rows do not exist in the R_DOC_RF table, use the DOC_LAST_DT of the transaction as the TBL_LAST_DT for the new reference record.
    - Insert the transaction reference into the R_DOC_RF table.

Problem Resolution

This program has transaction implemented. If the process fails for any reason, the transaction should be rolled back, and the process can be restarted.

2.1.9 Convert Lifecycle Inquiry EV Transaction Process

Description

Three new fields EV transaction Code(EV_DOC_CD), EV transaction department code (EV_DOC_DEPT_CD), EV transaction ID (EV_DOC_ID) were added to transaction header tables for the following types: PO transaction (PO_DOC_HDR), MA transaction (MA_DOC_HDR), and RE transaction (RE_DOC_HDR). These fields will store the EV transaction identifier if the transaction is created from an EV transaction. The Lifecycle Inquiry EV transaction Conversion process populates these new fields for the existing records on PO, MA and RE transaction header tables.

The Lifecycle Inquiry EV transaction Conversion process then creates the EV transaction references for the above changed transactions.

When to Run

This process will be run after Database Upgrade Scripts for the build are executed.

Before running this process,

1. The Database administrator should disable any trigger which is used to populate TBL_LAST_DT on R_DOC_RF table.
2. Depending on database types, the Database administrator may need to create a before insert trigger for R_DOC_RF table to populate the UNID column if sequence is used for this column.

Major Input

- EV_DOC_MAPO
- PO_DOC_HDR
- MA_DOC_HDR
- RE_DOC_HDR

Output

The fields EV Transaction Code(EV_DOC_CD), EV Transaction Department Code (EV_DOC_DEPT_CD), EV transaction ID (EV_DOC_ID) are populated on all Records of the PO transaction header, MA transaction header and RE transaction header where the header identifier can be found on the EV_DOC_MAPO table.

R_DOC_RF

Create transaction references record for PO, MA, and RE transactions which link to EV transactions.

Parameters

N/A

Sort Sequence

N/A

Selection Criteria

- For each record in the PO_DOC_HDR table, set the EV_DOC_CD, EV_DOC_DEPT_CD, and EV_DOC_ID fields with values from DOC_CD, DOC_DEPT_CD, DOC_ID fields of the EV_DOC_MAPO record whose GEN_DOC_CD, GEN_DOC_DEPT_CD, and GEN_DOC_ID equal to the PO transaction identifier.
- For each record in the MA_DOC_HDR table, set the EV_DOC_CD, EV_DOC_DEPT_CD, and EV_DOC_ID fields with values from DOC_CD, DOC_DEPT_CD, DOC_ID fields of the EV_DOC_MAPO record whose GEN_DOC_CD, GEN_DOC_DEPT_CD, and GEN_DOC_ID equal to the PO transaction identifier.
- For each record in the RE_DOC_HDR table, set the EV_DOC_CD, EV_DOC_DEPT_CD, and EV_DOC_ID fields with values from DOC_CD, DOC_DEPT_CD, DOC_ID fields of the EV_DOC_MAPO record whose GEN_DOC_CD, GEN_DOC_DEPT_CD, and GEN_DOC_ID equal to the PO transaction identifier.
- For each PO_DOC_HDR record,
    - Select DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, DOC_FUNC_CD, DOC_LAST_USID, DOC_LAST_DT from PO_DOC_HDR.
    - Select DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, DOC_FUNC_CD, DOC_LAST_USID, DOC_LAST_DT from EV_DOC_HDR who's DOC_CD, DOC_DEPT_CD, DOC_ID match the EV_DOC_CD, EV_DOC_DEPT_CD, EV_DOC_ID of the PO_DOC_HDR record.
    - Insert a new transaction reference record with the value above into the R_DOC_RF table.
- For each MA_DOC_HDR record,
    - Select DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, DOC_FUNC_CD, DOC_LAST_USID, DOC_LAST_DT from MA_DOC_HDR.
    - Select DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, DOC_FUNC_CD, DOC_LAST_USID, DOC_LAST_DT from EV_DOC_HDR who's DOC_CD, DOC_DEPT_CD, DOC_ID match the EV_DOC_CD, EV_DOC_DEPT_CD, EV_DOC_ID of the MA_DOC_HDR record.
    - Insert a new transaction reference record with the value above into the R_DOC_RF table.
- For each RE_DOC_HDR record,
    - Select DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, DOC_FUNC_CD, DOC_LAST_USID, DOC_LAST_DT from RE_DOC_HDR.
    - Select DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, DOC_FUNC_CD, DOC_LAST_USID, DOC_LAST_DT from EV_DOC_HDR who's DOC_CD, DOC_DEPT_CD, DOC_ID match the EV_DOC_CD, EV_DOC_DEPT_CD, EV_DOC_ID of the RE_DOC_HDR record.
    - Insert a new transaction reference record with the value above into the R_DOC_RF table.

Problem Resolution

This program has transaction implemented. If the process fails for any reason, the transaction should be rolled back, and the process can be restarted.

2.1.10 Create Warehouse/Inventory Location Table Entries

Job Name	Create Warehouse/Inventory Location Table Entries
Recommended Frequency	One Time. This conversion process should be executed to create newly added warehouse and Inventory tables in Advantage Financial 3.11
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes

Overview

In order to allow existing clients to use their current Primary Bin, Bin 1, and Bin 2 location setup, this batch job is needed to create data on the appropriate new tables to represent those locations.

This batch job creates the following Warehouse Location Type (WLOCTYP) values. These values are created for each Warehouse combination present on the Warehouse (WHSE) table, so the batch performs a lookup to WHSE to find every Warehouse combination available, and perform these updates for each one:

Warehouse	Warehouse Location Type	Name	Description
All warehouses on WHSE	PRIMBIN	Primary Bin	This is the primary bin the stock item is stored in.
All warehouses on WHSE	BIN1	Bin 1	This represents a secondary location that a given stock item may be stored at.
All warehouses on WHSE	BIN2	Bin 2	This represents a third location that a given stock item may be stored at.

The batch job subsequently creates the following Warehouse Location Structure (WLOCST) values. These values are created for each Warehouse present on the WHSE table, so the batch performs a lookup to WHSE to find every Warehouse available, and performs these updates for each one:

Warehouse	Location Structure	Description	Location Type 1	Location Structure Description
All warehouses on WHSE	PRIMBIN	This structure captures items stored at the Primary Bin location	PRIMBIN	Primary Bin
All warehouses	BIN1	This structure captures items stored at the Bin 1	BIN1	Bin 1

Warehouse	Location Structure	Description	Location Type 1	Location Structure Description
on WHSE		location		
All warehouses on WHSE	BIN2	This structure captures items stored at the Bin 2 location	BIN2	Bin 2

The batch job then begins the creation of Warehouse Inventory Location (WILOC) records to capture all existing Primary Bin, Bin 1, and Bin 2 locations. The batch performs a lookup across all Inventory Maintenance (INVN) records and identifies each unique Primary Bin, Bin 1, and Bin 2 value for each Warehouse WHSE table, and creates WILOC records to represent those values. When creating the WILOC records, the system should use the Department of the specified Warehouse on the INVN record.

Select the Inventory Location (ILOC) record for each INVN record where the Location Priority = 1 and set it as the default if there is not already an existing default record. If any ILOC record is already set as the default, then skip that INVN record.

While creating WILOC records to capture all existing Primary Bin, Bin 1, and Bin 2 locations, the batch simultaneously creates corresponding Inventory by Location (ILOC) records depending on the Track Quantity By Location flag value on the INVN record:

- The script creates an ILOC record for each WILOC record being created. On the ILOC record, the On Hand Quantity on the Primary Bin location is set equal to the On Hand Quantity of the INVN record and the Location Priority is set to 1. The On Hand Quantity for any Bin 1 and Bin 2 locations is set to zero and the Location Priority is set to 2 and 3, respectively. Note: If a Bin 1 value does not exist on the existing INVN record and a Bin 2 does exist, the Location Priority is set to 2.

Example:

For the below INVN records, multiple WILOC records are created as follows:

INVN records:

Warehouse	Stock Item	On Hand Quantity	Primary Bin	Bin 1	Bin 2
WHSE1	11111 001	100	132	55	24
WHSE2	11111 001	50	65	24	13
WHSE1	22222 001	75	12	11	10
WHSE2	22222 001	200	12	11	10
WHSE1	33333 001	400	15	22	
WHSE2	33333 001	1000	1444		
WHSE2	44444 001	1000	A1000		B12

WILOC records created by batch:

Warehouse	Location Structure	Inventory Location	Location Type 1	Location ID 1
WHSE1	PRIMBIN	132	PRIMBIN	132
WHSE1	BIN1	55	BIN1	55
WHSE1	BIN2	24	BIN2	24
WHSE1	PRIMBIN	12	PRIMBIN	12
WHSE1	BIN1	11	BIN1	11
WHSE1	BIN2	10	BIN2	10
WHSE2	PRIMBIN	65	PRIMBIN	65
WHSE2	BIN1	24	BIN1	24
WHSE2	BIN2	13	BIN2	13
WHSE2	PRIMBIN	12	PRIMBIN	12
WHSE2	BIN1	11	BIN1	11
WHSE2	BIN2	10	BIN2	10
WHSE1	PRIMBIN	15	PRIMBIN	15
WHSE1	BIN1	22	BIN1	22
WHSE2	PRIMBIN	144	PRIMBIN	144
WHSE2	PRIMBIN	A1000	PRIMBIN	A1000
WHSE2	BIN2	B12	BIN2	B12

ILOC records created by batch (assuming Track Quantity By Location = No for all above INVN records):

Warehouse	Stock Item	Location Structure	Inventory Location	On Hand Quantity
WHSE1	11111 001	PRIMBIN	132	100
WHSE1	11111 001	BIN1	55	0
WHSE1	11111 001	BIN2	24	0
WHSE1	22222 001	PRIMBIN	12	75
WHSE1	22222 001	BIN1	11	0

Warehouse	Stock Item	Location Structure	Inventory Location	On Hand Quantity
WHSE1	22222 001	BIN2	10	0
WHSE2	11111 001	PRIMBIN	65	50
WHSE2	11111 001	BIN1	24	0
WHSE2	11111 001	BIN2	13	0
WHSE2	22222 001	PRIMBIN	12	200
WHSE2	22222 001	BIN1	11	0
WHSE2	22222 001	BIN2	10	0
WHSE1	33333 001	PRIMBIN	15	400
WHSE1	33333 001	BIN1	22	0
WHSE2	33333 001	PRIMBIN	144	1000
WHSE2	44444 001	PRIMBIN	A1000	1000
WHSE2	44444 001	BIN1	B12	0

Restartability Information

This job does not support restartability. If the job fails for any reason, all updates are backed out and a new job must be scheduled.

Process Steps	Details
1. Processing WHSE records	Each of the WHSE records are processed to create WLOCTYP and WLOCST records.
2. Processing INVN records	Each of the INVN records are processed to create appropriate unique WILOC records for warehouses used on the records and then ILOC records for the Stock Items.

Major Input

- Warehouse (R_WHSE) table
- Inventory (R_INVN) table

Parameters

Parameter	Description	Default Value
Create Single ILOC record Y/N (Y= Yes N=No)	If the parameter is set to *Y*, create only one ILOC record using concatenated value of PRIMBIN/BIN1/BIN2.	<blank>
	If the value is set to *N*, create separate ILOC records for PRIMBIN/BIN1/BIN2.	
Set PRIMBIN as Default Y/N (Y= Yes N=No)	If the parameter is set to *Y*, set PRIMBIN ILOC record as the default record.	<blank>
	If the value is set to *N*, no ILOC record is set as the default record.	

Major Output

New entries in tables listed below:

- Warehouse Location Type (WLOCTYP)
- Warehouse Location Structure (WLOCST)
- Warehouse Inventory Location (WILOC)
- Inventory by Location (ILOC)

Job Return Code

The following table shows the potential job Return Codes for the Create Warehouse/Inventory Location Table Entries job.

Return Code	Condition
Successful (1)	All of the records are processed successfully without errors.
Warning (4)	N/A
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: • Entered 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.

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

If the process was discontinued for any reason the job cannot be restarted. A new job must be scheduled.

2.1.11 Customer Account Data Fix

Job Name	Customer Account Data Fix
Recommended Frequency	Once. Refer to the "Steps Before Running Job" and "Steps After Running Job" sections for important instructions that need to be followed before and after the execution of the batch process.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes

Steps Before Running Job

No other transactions should take place while the System Assurance 9 or Customer Account Data Fix is in progress.

Unarchive any archived Intercept Transactions.

The System Assurance 9 job should be run with existing Event Type setup to determine whether any Customer Account (CUSTA) records are out-of-sync. Discrepancies due to intercept fees updates are expected to be resolved.

Each site must ensure that the event types used to record Intercept Fees have the proper Customer Account Update flag and Customer Account Update Type setup. Unless there are special requirements to track intercept fees as collected amounts on CUSTA, it is recommended that the Customer Account Update flag be set to false and the Customer Account Update Type be set to blank for these event types so that intercept fees will not be recorded on CUSTA. The day zero event types that record Intercept Fees are: IF01, IF02, IF03, and IF04.

Overview

When an Intercept was performed during the Automated Disbursement process, if a Fee was captured along with Intercepted Money, the Fee amount would update CUSTA, while it was not showing up inside of the referenced RE. The Fee Amount should not update the CUSTA table, as this money should not count towards the amount that has been collected. It was discovered that day Zero data setup for several Event Types (IF01, IF02, IF03, and IF04) used for Intercept Fees have the Customer Account Update Flag and the Customer Account Update Type fields incorrectly set, thereby causing incorrect updates to the CUSTA table.

The Customer Account Data Fix job will fix CUSTA records with incorrect Earned Collected or Unearned Collected updates resulting from intercept fees collection. In order to rectify the CUSTA update issue, each site must ensure that the event types used for recording Intercept Fees have the proper setup, then run the Customer Account Data Fix. The fix should be followed by a non-incremental run of the System Assurance 9 job to resolve any out-of-sync CUSTA updates resulting from intercept fees.

Process Steps	Messages
1. Parameter Validation	Validating Batch Parameters listed in the above Batch Parameter section The basic validations are:

	• RPT_MODE: ensures that it is 1 or 2. If not, an error is logged. • RPT_PATH: If no value is provided, it takes the root directory as the output directory • RPT_FL_NM_PFX: Ensures that it is not empty. If empty, it takes the default "CustAcctUpdt" • SELECT_BLOCK_SZ: If no value is provided, it defaults to 500. If value provided is not a number, an error is logged • COMMIT_BLOCK_SZ: If no value is provided, it defaults to 100. If value provided is not a number, an error is logged • INTRCPT_DOC_CDS: If no value is provided, it defaults to value of "IT" • Validates other values but they can be defaulted. Batch Parameter validation completed.
2. Selection of Records	This starts the logical looping through each CUSTA detail record with potential data problem.
3. Calculation and Updates	Determine if the final version of the Intercept transaction exists on the system. Determine the correct Earned Collected and Unearned Collected amounts resulting from the Intercept Transaction for this CUSTA detail record. If the calculated amounts differ from the CUSTA detail amounts, report them to the spreadsheet report. If the correct amounts cannot be determined, indicate the reason on the spreadsheet report. For report and update mode only, update the CUSTA detail record with the correct amounts and make corresponding adjustments to the parent CUSTA record.

Restartability information

The job cannot be restarted but can be rescheduled as only records with data problems will be eligible for the data fix. The reason of job failure needs to be investigated by looking at the job error log and error log files.

Major Input

- Customer Account Detail (R_CUST_ACCT_DET)
- Customer Account (R_CUST_ACCT)
- Internal Exchange Accounting (IET_DOC_ACTG)

Batch Parameters

Parameter Name	Description (Caption)	Default Value
INTRCPT_DOC_CDS	Intercept Transaction Codes. Required. This should be the set of Intercept Transaction Codes which may have Fees that need to be backed out of the Customer Account tables.	IT
RPT_PATH	Report Path. Required. This is the location that the Report will be located when generated.	$AMSEXPORT$
RPT_FL_NM_PFX	Report File Name. Required. This is the prefix for the name of the report that will be generated for the current run. It should not contain an extension; an extension of .xlsx will be applied when the Excel Spreadsheet is created. If a file with the file name already exists, it will be deleted. The file name will have the format: RPT_FL_NM_PFX+job ID_DateTimeStamp.xlsx.	CustAcctUpdt
RPT_MODE	Report Mode. Required. This mode will dictate whether it was Report Only Mode (value of 1) or Report and Update Mode (value of 2). Report Only mode will only generate a report with the changes that would be performed if updating of the records is performed.	1 (Report Only)
SELECT_BLOCK_SZ	Select block Size. If not entered then defaulted to 500. It is the number of records that can be fetched as a single block. Can be used for Performance tuning.	500
COMMIT_BLOCK_SZ	Commit block Size. If not entered then defaulted to 100. It is the number of records that can be changed before a commit is called. This value is a suggestion and not a requirement, as we may call commit sooner than this count. Can be used for Performance tuning.	100

Major Output

- Customer Account Detail (R_CUST_ACCT_DET)

- Customer Account (R_CUST_ACCT)
- An Excel workbook named RPT_FL_NM_PFX+jobID_DateTimeStamp.xlsx will be written into the folder location indicated by the RPT_PATH parameter.

Job Return Code

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

Return Code	Condition
Successful (1)	Corrupted Customer Account records were successfully reported / fixed.
Warning (4)	If no incorrect record is found in the Customer Account Detail table then the job continues execution but ends in a job return code of Warning. Necessary details of the line are logged in the job log.
Failed (12)	This return code will be issued under several conditions: • Parameter validation failed. • 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 Sequence

The records will be sorted by:

- Department Code
- Unit Code
- Vendor Customer Code
- Billing Profile Code
- Transaction Code
- Transaction Department Code
- Transaction ID

Selection Criteria

Records will be selected for processing based on the following criteria:

- Transaction Code from R_CUST_ACCT_DET matches Intercept Transaction Codes from Job Parameter
- Earned Collected does not equal Earned Liquidated Rec or Unearned Collected does not equal Unearned Liquidated Rec on R_CUST_ACCT_DET

Problem Resolution

The following table shows the possible return codes and recommendations.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	No account was found with discrepancy **Sample Message:** No result was returned from the current result set query for transaction Code …	If no discrepancies were found, no further action is required.	N/A
	No final IET accounting lines were found matching the CUSTA detail record Sample Message: This Detail record contained a discrepancy which could not be resolved because the corresponding IET transaction cannot be found on the system	The records will show up in the report and will have to be investigated on individual basis.	N/A
Failed (12)	Required Parameters are not entered. **Sample Message:** Run Mode Value is required.	Enter valid parameters and run the job again.	N/A
	Run time exceptions for unexpected situations.	Investigate the cause of the issue by looking at the job log and error log files.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	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. Schedule a new job.	N/A

Steps After Running Job

Once the job has been run in report and update mode, the client will not run it again.

The System Assurance 9 job should be run in non-incremental mode to reflect the correct Event Type updates. Discrepancies due to intercept fees updates should be resolved.

2.1.12 Customer Account History Conversion Process

Description

The Customer Account History Conversion Process is a data conversion process to populate the Accounts Receivable Correspondence History Table with data from the Statement Print History Table, Invoice Print History Table and Payment Plan Print History table.

This job will fetch all the records from the Statement Print History table and issue an update on them as the system automatically inserts a record in the Accounts Receivable Correspondence History Table. The same approach is repeated for the Invoice Print History Table and the Payment Plan Print History table.

On the Client site, if there are a large number of records in the input tables, the job may take a longer time to complete. The job does not have 'Restart' facility as 'Checkpoint Logic' is not implemented. Therefore, if the job fails to complete due to any reason, the user will have to reschedule the job.

When to Run

This conversion process should be executed at the client site running on builds prior to 3.6 after the installation of the 3.6 build. (One time)

Steps to Run the Customer Account History Conversion Process

1. Enter the values in the following job parameters:
   - No of Transactions in a single commit
   - Select Block Size
2. Schedule the job

Major Input

- Statement Print History Table (R_STMT_PRN_HIST),
- Invoice Print History Table (R_INV_PRN_HIST) &
- Payment Plan Print History Table (R_ PYM_SCH_PRN_HIST).

Output

All the Records on the Statement History table, Print History table and Payment Plan History table are populated on the Accounts Receivable Correspondence History table (R_RCVB_CORS_HIST).

Parameters

Parameter	Description	Default Value
COMMIT_BLK	Number of transactions that will be in a single commit. If not entered then it is defaulted to 500. After	500

	processing the Commit block Size number of transactions a commit is issued. Can be used for Performance tuning. The value entered in this job parameter should be a positive integer greater than zero.	
SELECT_BLK	Select Block Size If not entered then it is defaulted to 1000. It is the number of records that can be fetched as a single block. Can be used for Performance tuning. The value entered in this job parameter should be a positive integer greater than zero.	1000

Sort Sequence

The records on R_STMT_PRN_HIST will be selected in the following order:

PRN_DT, VEND_CUST_CD, BPRO_CD, STMT_SEQ_NO, HIST_SEQ_NO

The records on R_INV_PRN_HIST will be selected in the following order:

DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, PRN_DT, INV_SEQ_NO,

VEND_CUST_CD, DEPT_CD, UNIT_CD, BPRO_CD

The records on R_PYM_SCH_PRN_HIST will be selected in the following order:

DOC_CD, DOC_DEPT_CD, DOC_ID, FORM_TYP, UNID, PNT_UNID

Selection Criteria

All the records on R_STMT_PRN_HIST, _INV_PRN_HIST and R_PYM_SCH_PRN_HIST get selected.

Problem Resolution

This job does not need ‘Restart’ facility. If the job fails then a new job should be scheduled. The new job will always first purge the Accounts Receivable Correspondence History Table and then the conversion will be started.

2.1.13 Encryption Conversion Utility for 3.6

Description

The Data Encryption Standard (DES) currently employed in Advantage applications will be replaced by the Advanced Encryption Standard (AES) for the Advantage Financial and VSS applications. The encryption standard will be applied to fields already encrypted within the application and secured fields. This batch process will encrypt the values of new fields that have been considered for encryption using AES and will also encrypt the existing encrypted Bank account number field (that was previously encrypted using DES) using AES.

The procedure for conversion is different for the Financial and VSS applications as detailed in the below section. The CSF.properties file should have properties set related to AES encryption, which should be the default setting in the provided software.

The conversion should be done in the following sequence:

Financial Data Conversion:

1. Run 'Encryption Conversion Utility for 3.6' on the Financial side. This will encrypt the specified fields on the specified Financial data objects.

DBA Intervention:

DBA intervention is required for converting data in the VSS application. The DBA should execute a SQL utility which will copy values of Shadow key and encrypted fields from the Financial table to the corresponding table on the VSS side by matching the primary keys. For example, the SQL script will copy the values of Shadow Key, ACCT_NO, and CC_NO fields from <financial schema>.R_VEND_CUST records to the corresponding records on <vss schema>.R_VEND_CUST. This should be done for all the tables involved in this process. This step will ensure that the encryption key and encrypted fields have the same values for corresponding table rows in Financial and VSS.

Following the last step will take care of encrypting the table rows that are unique on the VSS side.

VSS Section:

1. Run 'Encryption Conversion Utility for 3.6' on VSS side. This will encrypt the specified fields on the specified VSS data objects.

When to Run

This process will be run after Database Upgrade Scripts for the build are executed.

Major Input

Data objects that store one or more encrypted field(s). Note that the job may only be run for one data object at a time. The following data objects and columns were considered for encryption in this release:

Table Data Object	Column Name	Conversion needed on
R_VEND_CUST (Vendor Customer table)	CC_NO, ACCT_NO	FIN and VSS
R_PNT_VEND_CUST (Headquarters Account)	VSS_PASS, VEND_VER_HINT, VEND_PSWD_VERIFY	FIN and VSS
R_VEND_REG (Vendor Registration)	PSWD_TXT, VERIFY_PSWD_TXT, SEC_ANSWER, VEND_VER_PSWD, VEND_VER_HINT, VEND_PSWD_VERIFY	VSS
VCC_DOC_VCUST (VCC transaction)	CC_NO, ACCT_NO	FIN
VCM_DOC_HQ (VCM Transaction)	VEND_VER_HINT, VEND_PSWD_VERIFY, VSS_PASS	FIN
VCM_DOC_VCUST (VCM Transaction)	CC_NO, ACCT_NO	FIN
R_VSS_USER	VERIFY_PSWD_TXT, PSWD_TXT, SEC_ANSWER	FIN and VSS
AD_DOC_HDR	ACCT_NO	FIN
CR_DOC_VEND	CC_NO	FIN
R_AP_PRCU_CARD_ADM	PCARD_ID	FIN

Output

The Shadow column is populated on all records of the dataobject (listed as input) with a random number that will be applied by AES during the encryption and decryption process. The secured fields on the dataobject will be populated with the encrypted values.

Parameters

Job	Parameter	Description	Default Value
Encryption Conversion Utility for 3.6	DATAOBJECT_NAME	Name of the data object whose secured field needs to be encrypted based on AES	

Job	Parameter	Description	Default Value
	ATTRIBUTE_LIST	List of secured attributes of the data object specified in the DATAOBJECT_NAME parameter - comma delimited for multiple values	

Sort Sequence

N/A

Selection Criteria

The process will select all records of the data object specified in the DATAOBJECT_NAME parameter whose shadow column is null.

Problem Resolution

It is a good practice to look at the log of each job for errors even if the job has run successfully. In case of a failure the process can be restarted.

2.1.14 Encryption Standardization Conversion Utility for 3.6

Chain or Job Name	Encryption Standardization Conversion Utility for 3.6
Recommended Frequency	This job should be run once for every table that contains encrypted attributes if upgrading from 3.6 to 3.6.0.1. This job should not be run for 3.5 to 3.6 security conversion. Before beginning the process of converting a table from 3.6 to 3.6.0.1, make backups of all data objects intending to be run through this process.
Single Instance Required	The Encryption Standardization Conversion Utility for 3.6 does not support multiple instance of the job running in parallel.
Can be restarted?	The Encryption Standardization Conversion Utility for 3.6 can be restarted.
Reports generated	No Reports are generated for this process.

Overview

- The Encryption Standardization Conversion Utility is used to convert the encryption changes made in 3.6 from their platform dependent encryption process to a platform independent counterpart.
- The process begins by making sure that the input Data Object to be converted is valid. It will then select all of the rows in this data object, and for each row it will check to see whether the row uses the platform dependent encryption used in 3.6. It will then execute an update on that row if it was encrypted with the platform dependent encryption and change it to the platform independent encryption.

Process Steps	Messages
1. Validating Parameters	• 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: "Invalid Dataobject name: <data object name>" • If no encrypted columns for the entered data object returns 0, then the following message will be issued, "No encrypted columns were found for <data object>" • Validation Complete, is issued upon completion of the validation process
2. Selecting Records	• Selecting Records
3. Updating Records	• After selecting all of the rows in the table, the following message will be issued,

Process Steps	Messages
	"Updating Records" • While selecting rows to update, depending on the value of the Progress Block Size, the following message will appear in the log: "<Progress Block Size Multiple> rows processed" • When the updating of records is complete, the message, "Updated <number of records updated> Records" will be issued.

- The Encryption Standardization Conversion Utility for 3.6 can be restarted. It will again select all of the records in the data object name parameter provided; however, it will ignore rows that have already been converted to the new encryption scheme from any previous runs.
- This job should be run once completely for each data object that contains encrypted fields if upgrading from 3.6 to 3.6.0.1. This job does not support multiple instances that run in parallel.

Major Input

- Data objects that store one or more encrypted field(s). Note that the job may only be run for one data object at a time. Although not required to contain encrypted fields, this job does nothing to the provided Data objects that have none.
- Progress Block Size represents the amount of rows you would like the job to process before reporting this in the job log. The default value is 1000.

Major Output

- The Shadow column is populated on all records of the data object with a random number that will be applied by AES during the encryption and decryption process. The secured fields on the data object will be populated with the encrypted values. This will only be done for rows that use the platform dependent encryption as opposed to the platform independent.

Return Code	Condition
Successful (1)	All of the selected records are updated successfully. It will also return Successful if the provided Data Object contains no encrypted columns.
Failed (12)	The job will fail under the following conditions: • Parameter is invalid (Data Object is not found or not provided, or Progress Block Size was not positive) • Run time exceptions for unexpected situations
Terminated (16)	This return code will be issued when the job is terminated by the user.

Return Code	Condition
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

Selects all of the rows of the data object specified.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the parameters are validated successfully.	N/A	N/A
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following three conditions. 1) Encounters any runtime exceptions. 2) Data object parameter was not provided. 3) Data object parameter was invalid. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and reschedule the job.	If another job has already been scheduled and ran successfully for the same data object, then this job does not need to be re-scheduled.
	Failed because the data object parameter was not provided. **Sample Message**: Enter Data Object Name job parameter	**Recommendation:** Schedule a new job, providing the Data Object Name job parameter.	
	Failed because the data object parameter was not valid.	**Recommendation:** Schedule a new job, providing a valid data object	

Possible Return Codes	Condition	Recommendation	Other Instructions
	Sample Message: Invalid data object name: R_SOME_BAD_TABLE	Name job parameter.	
	Failed because the Progress Block Size was not positive. **Sample Message**: Progress Block Size must be a positive integer.	**Recommendation**: Schedule a new job, providing a positive Progress Block Size.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can be re-scheduled as a new job or restarted.	If another job has already been scheduled and ran successfully for the same data object, then this job does not need to be re-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 job has already been scheduled and ran successfully for the same data object, then this job does not need to be re-scheduled.

2.1.15 EV Transaction Conversion Process

The EV Transaction Conversion Process is a data conversion process for the EV transaction. This process will apply conversion to all the existing EV Transactions (drafts and final) to update the Awarded Items page, Award By Line Page and Solicitation Commodity Line page with the new information and bring them in sync with the new look and feel of the EV Transactions.

Description

This conversion process updates the following pages and data objects depending on the value of Job Parameter, 'Run Mode'.

If Run Mode = 1, which means Full Conversion, the following pages and related data objects will get updated.

- Awarded Items page (EV_DOC_MAPO),
- Award by Line page (EV_DOC_SRCOMMLN)
- Solicitation Commodity Line page (EV_DOC_SOCOMMLN)
- Award By Group page (EV_DOC_SRCOMMGP)

If Run Mode = 2, which means only Commodity Group Conversion, the following pages and related data objects will get updated.

- Award By Group page (EV_DOC_SRCOMMGP)
- Award by Line page (EV_DOC_SRCOMMLN)

When to Run

This conversion process should be executed at the client site running on builds prior to 3.6 after installation of the 3.6 build.

Awarded Items Page: (EV_DOC_MAPO):

This process will first read all records on EV_DOC_MAPO (Awarded Items). For each EV_DOC_MAPO record it will get the created Award transaction record and will check the commodity lines on Award Transaction.

If the Award transaction has only one Commodity Line, then for existing Awarded Items records, the following fields will be updated on EV_DOC_MAPO:

- SO Commodity line number
- Vendor Code
- Vendor Name
- Alias/DBA
- Quantity
- Contract Amount
- Discount

If the Award transaction has more than one commodity line, then the existing record on EV_DOC_MAPO will be updated as explained above and for the rest of the commodity lines new records will get inserted in EV_DOC_MAPO.

These new records will be created by copying the corresponding existing record.

- The values of the SO Commodity line number field will be retrieved from the Award Transaction's Commodity Line.
- The values of the Vendor Code, Quantity, Contract Amount, and Discount fields will be retrieved from the Response transaction's Commodity Line (SR_DOC_COMMLN).
- The values of the Vendor Name and Alias/DBA fields will be retrieved from the Vendor Customer table (R_VEND_CUST).

Award By Line Page (EV_DOC_SRCOMMLN):

This conversion process will update the following fields:

- Award Quantity
- Contract Amount
- Quantity Awarded (This field is hidden on the EV transaction. The value of this field is used to calculate Open Quantity on the SO Commodity Line on the EV transaction)

The values of these fields will be retrieved from the Commodity Line of the Response transaction (SR_DOC_COMMLN)

- Commodity Group Line No

The value of this field will be set by Solicitation Commodity Group Line No of Award By Line (EV_DOC_SRCOMMLN)

Solicitation Commodity Line Page (EV_DOC_SOCOMMLN):

The Open Quantity field will get calculated as per the following formula:

Open Quantity = Total Quantity – Awarded Quantity

Award By Group Page (EV_DOC_SRCOMMGP):

- Response Commodity Group Line No

The value of this field will be retrieved from the Commodity Group Line of the Response transaction (SR_DOC_COMMGP)

Steps to Run the EV Transaction Conversion Process

1. Enter the values in the following job parameters:
    - No of EV Transactions in a single commit
    - Select Block Size
    - Run Mode
2. Schedule the job

Major Input

1. Awarded Items (EV_DOC_MAPO) records on the EV transaction

2. Award transaction commodity line records (PO_DOC_COMM/MA_DOC_COMM)
3. Response transaction commodity line records (SR_DOC_COMMLN)
4. Vendor Customer table (R_VEND_CUST)
5. Response transaction commodity group line records (SR_DOC_COMMGP)

Output

1. Awarded Items (EV_DOC_MAPO) records on the EV transaction

The following fields will be updated on EV_DOC_MAPO:

- SO Commodity line number
- Vendor Code
- Vendor Name
- Alias/DBA
- Quantity
- Contract Amount
- Discount

2. Award By Line (EV_DOC_SRCOMMLN) records on the EV transaction

The following fields will be updated on EV_DOC_SRCOMMLN

- Award Quantity
- Contract Amount
- Quantity Awarded (This field is hidden on the EV transaction. The value of this field is used to calculate Open Quantity on the SO Commodity Line on the EV transaction)
- Commodity Group Line No

3. Solicitation Commodity Line (EV_DOC_SOCOMMLN) records on the EV transaction

The following fields will be updated on EV_DOC_SOCOMMLN

- Open Quantity

4. Award By Group (EV_DOC_SRCOMMGP) records on the EV transaction

The following fields will be updated on EV_DOC_SRCOMMGP

- Response Commodity Group Line No

Parameters

Parameter	Description	Default Value
No of EV Transactions	Required.	100

in a single commit	The value entered in this job parameter should be a positive integer greater than zero.	
Select Block Size	Required. The value entered in this job parameter should be a positive integer greater than zero.	1000
Run Mode	Required. Run Mode. For Full Conversion enter 1. For only Commodity Group Conversion enter 2.	1

Sort Sequence

The records on EV_DOC_MAPO will be selected in the following order:

DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, DOC_MAPO_NO

The records on EV_DOC_SRCOMMLN will be selected in the following order:

DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, RESP_DOC_CD, RESP_DOC_DEPT_CD, RESP_DOC_ID, RESP_DOC_VERS_NO, DOC_SOCOMMLN_LN_NO, DOC_SOCOMMLN_GP_NO, DOC_SOCOMMLN_NO, DOC_SRCOMMLN_NO

The records on EV_DOC_SRCOMMGP will be selected in the following order:

DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, DOC_SRCOMMGP_LN_NO, RESP_DOC_CD, RESP_DOC_DEPT_CD, RESP_DOC_ID, RESP_DOC_VERS_NO

Selection Criteria

All EV_DOC_MAPO (Awarded Items on EV transaction) where the Vendor Customer Code and SO Commodity Line No is null gets selected.

All EV_DOC_SRCOMMLN (Award By Line on EV transaction) gets selected.

All EV_DOC_SRCOMMGP (Award By Group on EV transaction) gets selected.

Problem Resolution

This job does not need ‘Restart’ facility. If the job fails then a new job should be scheduled. The new job will not pick up the EV_DOC_MPAO records that are already processed, since the selection criteria of the Job for EV_DOC_MAPO picks up records where the Vendor Customer Code and SO Commodity Line No is null on EV_DOC_MAPO.

2.1.16 FDT Special Conversion Routine

Chain or Job Name	FDT Special Conversion Routine
Recommended Frequency	One time conversion routine run after any and all necessary setup has been done to the Future Transaction Triggering Options page.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

The FDT Special Conversion Routine is a one-time conversion process to update existing FDT data for un-triggered records based on setup on the FDTO page, while also reporting on un-triggered FDT records in the form of a text file that violate FDTO setup and that need to be corrected manually or through custom SQL to conform to FDTO records.

The parameter validation will be done for the commit size, output text file and output directory if the commit size parameter is entered wrongly then the default commit size (100) will be taken into consideration and if the output file or the output directory is entered wrongly then the job will log the proper job log message and then the job gets moved to a Failed state.

Once parameter validation is successful, then the job selects the records from the Future Transaction Triggering Options (FDTO) table. If there are no records available in FDTO then the job gives a warning message; otherwise, the job selects records from the Future Transaction Triggering (FDT) table that are 'Active' and have a status of 'Ready'.

FDT record processing is as follows.

1. Select records from FDT that are Active and have a status of *Ready*.
2. For the FDT record selected in Step 1 select the FDTO record by using the same Transaction Code and Department Code or by using the Department Code of *ALL*.
3. If the FDTO record exists then verify that the Bypass Approval Option field on FDTO is *Do Not Allow* or *Require*.
    1. If the Bypass Approval Option is *Do Not Allow* on the FDTO record and the Bypass Approval Flag is checked on the FDT record then update the FDT record by un-checking the Bypass Approval Flag.
    2. If the Bypass Approval Option is *Require* on the FDTO record and the Bypass Approval Flag is not checked on the FDT record then update the FDT record by checking the Bypass Approval Flag.
    3. If the Transaction Prefix field on the FDTO record is not null then update the Transaction Prefix of the FDT record with the Transaction Prefix from the FDTO record.
4. If the Copy Final Phase Only flag is checked on the FDTO record and the corresponding record on FDT is a Recurring Type record then verify the transaction phase of the record on the transaction catalog and if the transaction Phase is not Final or Historical Final then log the record to the output text file.

Process Steps	Messages
1. Parameter	• Validating Batch Parameters Parameters are valid or invalid

Process Steps	Messages
Validation	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	• Selecting eligible records. • If the selection returns 0 records, then the following message will be issued: "No Future Transaction Trigger Option records found for processing".

Major Input

- Commit Size - The number of records need to be committed.
- Text File – The output file maintains records that need user attention after completion of the job.

Parameter	Description	Default Value
AMSEXPORT	Export Import Directory Required Parameter. The export import directory is the place where the output text file generated will be maintained.	$$AMSROOT$$/ExportImport
COMMIT_SIZE	COMMIT SIZE Used to commit a set of records.	100
FDT_TEXT_FILE	Output Text File The file shows the records that are mismatched in between the FDT and FDTO table. For example: The FDT entry has a transaction record entry that does not have a Final or Historical Final status in the Transaction Catalog; however, the corresponding record on FDTO has the Copy Final Phase Only flag checked. In this case the coma separated values of transaction code, department, transaction id, transaction version and transaction status will be logged to the output file.	FDTTrigger.txt

Major Output

- The Text File contains the records that are inconsistent between FDT and FDTO and need user attention.
- The Bypass Approval Flag and Transaction Prefix on FDT Records get updated depending on the corresponding FDTO record.

Job Return code

Return Code	Condition
Successful (1)	All of the selected Future Transaction Triggering Option records are processed successfully.
Warning (4)	No eligible records found. This could be because of the following reasons: • There are no records available in the Future

Return Code	Condition
	Transaction Triggering Options table.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Unable to process FDT records • Run time exceptions for unexpected situations. When this job ends with a return of code Failed.

Sort Criteria

The selected records get sorted depending on the Transaction Code and Department Code.

Selection Criteria

The job selects FDT records that are Active and have the Status of Ready on the Future Transaction Triggering table.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	The job ended with a Warning because there are no records available in the Future Transaction Triggering Options table.	Enter records into the Future Transaction Triggering Options (FDTO) table and then restart the job.	N/A
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following three conditions. 1) Encounters any runtime exceptions. 2) Failed during restart. 3) Parameters entered are not valid. If the job fails because of the runtime exceptions, investigate	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.

Possible Return Codes	Condition	Recommendation	Other Instructions
		the exception reported by the process, resolve the error and restart the job. If job fails because of invalid parameters then enter proper parameters and then restart 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.	**Recommendation:** Schedule a new 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 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.
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 a new job can be scheduled.	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.

2.1.17 Fix Order Amt on MA (Recalculate Master Agreement Balances)

Chain or Job Name	Fix Order Amt on MA (Recalculate Master Agreement Balances)
Recommended Frequency	Run this job after completing an upgrade that includes Advantage Financial release 3.7, but before data entry begins on the upgraded release. This job is a conversion routine that only needs to run once. Do not run this job while users are processing transactions. This process recalculates Master Agreement Amounts and Quantities, so if Master Agreements are not used at your site this job may not be necessary.
Single Instance Required	Run this job in a single process mode.
Can be restarted?	This job cannot be restarted. If the process fails, it should be re-run. Each time it runs, the process recalculates all amounts from scratch.
Reports generated	This job will generate a report listing Master Agreements that have data issues that you must resolve before converting Master Agreement balances.

Overview

This conversion process recalculates pre-existing amount and quantity fields on the Master Agreement (MA) and initializes newly added amount and quantity fields. Because the 3.7 logic differs from Master Agreement logic in prior releases, expect the Master Agreement balances to have different values after the conversion. These new values will be more correct.

The pre-existing fields to be recalculated are:

- Ordered Amount
- Ordered Quantity
- Service Contract Amount

New fields to be initialized are:

- Expended Amount
- Expended Quantity
- Expended Contract Amount

The Quantity and Contract Amount fields appear only on the MA Commodity Line. The Ordered Amount and Expended Amount fields appear on every MA component and page listed below:

- Master Agreement Components
  - Header

  - Authorized Department
  - Authorized Unit
  - Vendor
  - Commodity Line
- Renewal History (RNHIST) page
- MA Reference for Authorized Departments (AUTHREF) page
- MA Reference for Authorized Units (AUTHREFU) page.

Note: There are many aspects to this conversion covered by the run sheet. Please review it carefully before starting the conversion. The sections of the Run Sheet are organized as follows:

- **Assumptions**: Key notes and limitations about the Conversion
- **Background**: Information about the new logic in 3.7 that requires a conversion.
  - Control and Reporting on Authorized Department and Authorized Unit Amounts: A special section that discusses these two MA components in detail.
- **Prerequisites**: Critical information about the steps that should occur prior to the conversion.
- **Performance Suggestions**: Details on the database indices that will speed up the process significantly.
- **SQL Validation Details:** Further details about executing the SQL scripts that identify data issues. These issues should be addressed prior to the conversion process.
- **Conversion Process Details:** The largest section of this transaction, it contains information about the process logic and the job log messages output at each step of the logic.
- **Restart Capability:** A standard section of the run sheet that notes that this process has no restart capability. (The job can be re-run as needed.)
- **Major Input:** Details on the process Inputs.
- **Major Output:** Details on the process Outputs.
- **Job Return Codes:** A standard section that describes possible job return codes.
- **Sort Criteria**: Notes that very little sorting occurs in this process.
- **Selection Criteria:** Identifies that all records on the Transaction Catalog are selected.
- **Post-Processing Tasks:** Steps to take after the conversion completes.
- **Problem Resolution:** Suggestions for resolving various job return codes.
- **Correcting Data Identified by the SQL Validations:** A special section that describes the suggested resolution for the four data validation scenarios that could be identified by the SQL Validations.

Assumptions

The successful recalculation of the Master Agreement balances depends upon the following:

- Ordering transactions, and the transactions that reference them, must be on the Transaction Catalog (they cannot be archived), because they are needed to calculate the new amounts and quantities.

- You must resolve all SQL Validation issues according to the instructions in the section "Correcting Data Identified by the SQL Validations."
- There are some limitations to the conversion updates:
- The conversion updates the Renewal History (RNHIST) page amounts for the **current** renewal period only. Current balances are calculated as the difference between the amounts in prior renewal periods, which are not updated, and the newly updated Master Agreement header balances. In some situations, the current renewal period balance could be reduced (and possibly go negative) due to corrective entries for prior periods. The RNHIST amounts are not controlling amounts, so there are no impacts from these calculations.
- The process calculates amounts for the current "Final" version of the Master Agreement and then automatically carries these amounts to any "Pending" or "Draft" versions of the MA. Transactions in "Historical" and "Conflict Draft" phases are not updated, nor are MA transactions that were canceled or archived.
- The conversion routine updates tables with direct SQL update statements. No business rules are fired during these updates. For example, the conversion will not validate Department and Unit codes to see if they are still valid and active.

Background

This conversion job is required due to three changes to the Master Agreement balance logic that are all included in release 3.7.

- The first change is a redefinition of update logic for the Ordered Amount, Quantity and Service Contract Amount fields on the Master Agreement.
    - Prior to release 3.7, these fields reflected the amount and quantity ordered against the MA. The logic did not take into account if a Payment Request referenced the Order for less or more than the original Order's Amount and Quantity.
    - In the 3.7 version of Advantage Financial, update logic will adjust the Ordered Amount and Quantity when the Payment Request performs a Final over or short reference to the Purchase Order (DO or PO) transaction. If more or less was requested for Payment, then that difference is adjusted on the Master Agreement balances. The same updates will occur for Encumbrance Corrections and Lapsing Transactions.
- The second change was the introduction of new Expended fields on the Master Agreement. These fields are not used for balance controls, but are useful for reporting purposes.
    - Payment Requests referencing an Order will update the Expended fields, in addition to any adjustment to the Ordering Amount. A special scenario is Payment Requests that directly reference the Master Agreement (with no Ordering transaction). These Payment Requests will update both Ordered and Expended fields with the same amount.
    - "Expended" fields did not exist before release 3.7, except on the Renewal History (RNHIST) page. However, updates to that field were not consistent, so it had little value.
- The third change is the new logic for Authorized Department and Authorized Unit controls.
    - Prior to release 3.7, when a transaction, say a Purchase Order, referenced a Master Agreement, the Department and Unit used to verify Authorization was the Issuer ID's Department and Unit.
    - This logic changed in 3.7 to use the Transaction Department and Unit on the transaction referencing the Master Agreement. This change has significant impact on the Authorized Department and Unit amounts, which requires this data conversion. This change makes for a more accurate update for Advantage sites where centralized buyers (the "Issuer") process orders for multiple departments.

Control and Reporting on Authorized Department and Authorized Unit Amounts

The process calculates the Authorized Department and Unit amounts on the Master Agreement as follows:

- Explicitly Authorized Departments/Units – the amount ordered by that Department/Unit.
- Authorized Department or Unit "ALL" – the amount ordered by all Departments except those explicitly excluded or explicitly included.
- Explicitly Excluded Departments – the amount is set to zero.

Example - Assume that an MA has these Authorized Departments:

Department	Spending Limit
100	$1,000
ALL	$5,000

Assume that orders to date have been as follows:

Transaction Department	Actual Orders
100	$600
300	$1,200
400	$900

After the conversion script the MA's Authorized Department Ordered Amounts will be:

Department	Spending Limit	Ordered Amount
100	$1,000	$600
ALL	$5,000	$2,100

Note: Master Agreement logic for Authorized Departments and Units updates two separate pages (AUTHREF and AUTHREFU) when activity occurs for a Department or Unit that is authorized by an "ALL" line on the Master Agreement. In the example above, the amounts ordered by Departments 300 and 400 would display on AUTHREF.

Insufficient controls in releases before 3.7 could mean that one of four problematic scenarios exist with the Authorized Departments and Units. ***Important*****:** *Before running the conversion routine, you should run the validation SQL scripts (see Prerequisite #6) and perform any resultant data cleanup.* Instructions on the failure conditions and recommended resolutions appears later in this run sheet.

The goal of the conversion program and data cleanup is to make these fields as accurate as possible. However, even after the SQL scripts run cleanly, there are still scenarios that could cause Authorized Department and Unit amounts to not match actual activity since the MA was in

effect. (Amounts **will** be correct at other MA components: header, vendor, and commodity line.) There are four reasons that the converted amounts will not match actual historical usage:

1. The batch process recalculates amounts using the Authorized Departments and Units from the **current "Final"** version of the MA. It is possible that other Departments or Units were allowed to use the MA at some point in time (and placed orders) but are now not on the list of Allowable Departments, either as an explicit Department code or under the "ALL" line. In this scenario, the Department and Unit amounts will not show the activity for Departments or Units that have historically used the MA but are not authorized on the current Final version.
2. The script uses the *Transaction* Department and *Transaction* Unit from orders to calculate amounts for Authorized Departments and Units, which is the new 3.7 logic. It is possible that these values on any given order are different from the *Issuer's* Department and Unit, which the pre-3.7 logic used for calculations. For example, a scenario could occur where the Issuer's Department was authorized to order from an MA, but the Transaction's Department is not authorized. In this scenario, the ordered amount would not appear in the Master Agreement's Authorized Department balance because the Transaction Department is not an Authorized Department value.
3. The batch process will reset the Ordered Amount on MA Authorized Departments and Units marked as "Exclude" to $0. The updates for Units are inherited from their Department. If a Department is excluded, then all of its Units will be considered excluded, even if their Authorized Unit records are not marked "Excluded".
4. If an order was created and then cancelled before it was paid, Conversion will skip that order when creating Authorized Department and Unit records.

To report on what Departments and Units have ordered against the MA, reports should be run against Purchase Order activity; not against the AUTHREF and AUTHREFU tables. Amounts on these pages will be properly set to control activity from this point forward – not for historical reporting.

Prerequisites

There are a set of steps that to perform before running this job in update mode:

1. Determine if Master Agreement Orders (generally DOs and POs, but also PRs starting with release 3.6) or the transactions that reference them (generally Payment Requests and lapsing entries) have been archived. *Archived transactions will not be included in the conversion calculations*. The suggested resolution is to unarchive those transactions and then re-archive the transactions after the conversion is over. There is no problem if only the Historical versions of transactions were archived.
2. Determine if Historical versions of *current* Master Agreements were archived at your site. If they were, the suggested resolution is to unarchive those historical versions and then re-archive the transactions after the conversion is over. Historical versions of Master Agreements must be on the Transaction Catalog for the validation SQLs to warn you about possible data issues in the Authorized Department and Unit components. If Historical versions are not restored the conversion routine will still work, but it will only use the current versions of the Master Agreement on the Transaction Catalog for the validations.
3. Execute the database scripts provided to create indexes before the batch process is run because those indexes significantly improve the performance of the batch process. Please see the next section on "Performance Suggestions." Once conversion completes, those added indexes can be dropped.
4. Run the provided set of SQL Validations running the conversion script.

a. The SQLs may be found on Disk 05 for your DB Type and OS Type under a folder entitled MABalances.
b. Run the SQL scripts as instructed in the section below “SQL Validation Details”
c. Resolve any issues found according to the instructions in this Run Sheet - see the section “Correcting Data Identified by the SQL Validations.”

5. To update the Unit name of Authorized Reference Unit a synonym needs to be created for the R_UNIT table of the ADMIN schema in the Financial Schema so that Financial schema can access R_UNIT. Without this synonym, the job will fail.
6. Schedule the conversion process to run when there is no transaction processing going on in the system and no batch processes are being run concurrently that update Master Agreement, Purchase Order and Payment Request transaction tables.
7. A backup is recommended before this process since this job commits after completing each of the component sections. If the job fails in the middle of Updating, some of the components will have been changed and others not.

Performance Suggestions

There are six non-unique indices that should be added to their respective tables before this job is run in order to maximize performance, and should be dropped after the conversion is complete. The indices should be created with the fields in the specified order. The examples are shown with a suggested index name, but any index name that the database will find valid can be used.

1. PO_DOC_HDR: This index links the Purchase Order Header to the Master Agreement.

   - AGREE_DOC_CD
   - AGREE_DOC_DEPT_CD
   - AGREE_DOC_ID

   Example (Oracle syntax):

   CREATE INDEX NT_PODOCHDR_MABAL ON PO_DOC_HDR
   (AGREE_DOC_CD, AGREE_DOC_DEPT_CD, AGREE_DOC_ID)

2. PO_DOC_COMM: This index links the Purchase Order Commodity Line to the Master Agreement.

   - AGREE_DOC_CD
   - AGREE_DOC_DEPT_CD
   - AGREE_DOC_ID
   - AGREE_VEND_LN_NO
   - AGREE_COMM_LN_NO

   Example (Oracle syntax):

   CREATE INDEX NT_PODOCCOMM_MABAL ON PO_DOC_COMM
   (AGREE_DOC_CD, AGREE_DOC_DEPT_CD, AGREE_DOC_ID, AGREE_VEND_LN_NO,
   AGREE_COMM_LN_NO)

3. PO_DOC_ACTG: This index links the Purchase Order Accounting Line to the Master Agreement.

   - AGREE_DOC_CD

- AGREE_DOC_DEPT_CD
- AGREE_DOC_ID
- AGREE_VEND_LN_NO
- AGREE_COMM_LN_NO

Example (Oracle syntax):

CREATE INDEX NT_PODOCACTG_MABAL ON PO_DOC_ACTG
(AGREE_DOC_CD, AGREE_DOC_DEPT_CD, AGREE_DOC_ID, AGREE_VEND_LN_NO,
AGREE_COMM_LN_NO)

4. PR_DOC_COMM: This index links the Purchase Request Commodity Line to the Master Agreement.

   - AGRMT_DOC_CD
   - AGRMT_DOC_DEPT_CD
   - AGRMT_DOC_ID
   - AGRMT_VEND_LN_NO
   - AGRMT_COMM_LN_NO

   Example (Oracle syntax):

   CREATE INDEX NT_PRDOCCOMM_MABAL ON PR_DOC_COMM
   (AGRMT_DOC_CD, AGRMT_DOC_DEPT_CD, AGRMT_DOC_ID, AGRMT_VEND_LN_NO,
   AGRMT_COMM_LN_NO)

5. PR_DOC_ACTG: This index will be used to assist in calculating Amounts to be set on the Master Agreement.

   - AGRMT_DOC_CD
   - AGRMT_DOC_DEPT_CD
   - AGRMT_DOC_ID
   - AGRMT_VEND_LN_NO
   - AGRMT_COMM_LN_NO

     Example (Oracle syntax):

     CREATE INDEX NT_PRDOCACTG_MABAL ON PR_DOC_ACTG
     (AGRMT_DOC_CD, AGRMT_DOC_DEPT_CD, AGRMT_DOC_ID,
     AGRMT_VEND_LN_NO, AGRMT_COMM_LN_NO)

6. MA_DOC_COMM: This index will be used to assist in updating the Master Agreement.

   - DOC_PHASE_CD
   - DOC_FUNC_CD

     Example (Oracle syntax):

     CREATE INDEX NT_MADOCCOMM_MABAL ON MA_DOC_COMM
     (DOC_PHASE_CD, DOC_FUNC_CD)

- SQL Validation Details

  Since this data conversion uses database SQLs to perform the necessary work to reset the desired values, it must ensure that a series of conditions do not exist which will cause the SQLs to fail. To resolve these issues before conversion time, the Advantage 3.7 release includes separate SQL scripts that may be run at any time *(even before the 3.7 upgrade!!)* to search for these failure conditions. See Prerequisite step 6 for the location of these scripts.

- It is strongly recommended that you run the SQL Validation scripts prior to conversion and that you identify and resolve any necessary errors. Some errors may be identified but do not require any steps to resolve because some scenarios will be corrected automatically by the Conversion process.

- The Validation scripts perform the same validation logic discussed below in the section called "Identify Known Failure Conditions." The steps for resolution are also the same. The only difference is that the SQL Validation scripts do not produce a formatted error report, they only write output to a spooled file.

The SQL scripts must be executed as appropriate for the database operating system that you use. Regardless of the scenario, please consider these notes:

1. You must spool the output of the scripts to retain a list of transactions that have problems.
2. If you are running the validation multiple times, consider that new runs will overwrite old spooled output. If the spooled output previously created still exists and should be kept, then back up that prior run's file before the script is run this again.
3. Some tools (e.g. SQL Plus) have a limitation as to the size of a spooled file. If this is exceeded then the SQL script will fail. If this happens, you must break up the SQLs into individual files / spool files so as not to exceed the maximum size of SQL Plus's spooling feature.
4. Each script will send output to the spool indicating if problems were found. Problem records will be identified by Transaction ID and Authorization line (Department or Unit). Here is an example:

```
DOC_CD   DOC_ DOC_ID                DOC_AUTHDEPT_LN_NO
-------- ---- --------------------- ------------------
MA       010  01270400000000000033                   1
MA       010  06300500000000000153                   1
MA       TB01 03040600000000000013                   1
```

5. When run as a single script, the output does not identify which of the validations found a problem. If problems are found then the script should be run one validation at a time to determine which one found the issue.

Conversion Process Details

There are a series of high level steps that are performed inside of the batch job. The steps are:

1. Validate and Initialize values
2. Identify known failure conditions
3. Remove duplicate Authorized Departments / Authorized Units

4. Reset fields and tables

The following sections discuss each of the high level steps listed above.

Validate and Initialize Values

This step has two parts; the first part is the validation of the parameters for this job. The second step sets a series of internal values based upon the environmental variables for the Advantage installation.

If there is a problem in either of these parts the job will halt with a failure condition. The job must be re-run with the problem corrected (e.g. with valid parameters).

The following are the steps performed and log messages generated.

Process Steps	Messages
1. Parameter Validation	• Validating batch parameters. • Batch parameters are valid. [When the parameters are valid.] • Batch parameters are not valid. [When the parameters are not valid, which means the process will fail.]
2. Set Internal Values	• No messages

Identify Known Failure Conditions

Since this data conversion uses database SQLs to perform the necessary work to reset the desired values, it must ensure that a series of conditions do not exist which will cause the SQLs to fail.

Although the conversion can bypass some of these conditions during conversion, it is strongly suggested that you run the SQL scripts and resolved as many conditions as possible. For resolution instructions, see the section below on “Correcting Data Identified by the SQL Validations.”

The set of conditional checks are:

D1. Duplicate Departments in the Master Agreement Authorized Department Component [will cause a problem inside of the Data Conversion]

D2. Master Agreement Authorized Department Code changed from one version to another [may not capture some of the Orders inside of the Authorized Department Ordered Amount updates]

D3. Master Agreement Authorized Department changed Exclude Flag from one version to next [may not capture some of the Orders inside of the Authorized Department Ordered Amount updates]

D4. Two or more Authorized Department records with a non-zero Spending Limit [will cause a problem inside of the Data Conversion]

Authorized Units are the same basic queries:

U1. Duplicate Units in the Master Agreement Authorized Unit Component mapped to the same Department Code [will cause a problem inside of the Data Conversion]

U2. Master Agreement Authorized Unit Code changed from one version to another [may not capture some of the Orders inside of the Authorized Unit Ordered Amount updates]

U3. Master Agreement Authorized Unit changed Exclude Flag from one version to the next [may not capture some of the Orders inside of the Authorized Unit Ordered Amount updates]

U4. Two or more Authorized Unit records with a non-zero Spending Limit mapped to the same Authorized Department record [will cause a problem inside of the Data Conversion]

If any of these problems are encountered the report will list the set of errant transactions to resolve. This table summarizes the conditions, their severity, and remediation steps:

Condition	Fatal/ Non-Fatal	Remediation
D.1: Duplicate Authorized Departments	Fatal	See section entitled “Correcting Data Identified by the SQL Validations”, Validation 1. This failure condition is rectified by the job when requested to bypass all errors by deleting Duplicate Authorized Departments. See the section entitled “Remove duplicate Authorized Departments / Authorized Units” for a brief discussion.
U.1: Duplicate Authorized Units	Fatal	See section entitled “Correcting Data Identified by the SQL Validations”, Validation 1. This failure condition is rectified by the job when requested to bypass all errors by deleting Duplicate Authorized Departments. See the section entitled “Remove duplicate Authorized Departments / Authorized Units” for a brief discussion.
D.2: Authorized Department changed from previous version	Non-Fatal	See section entitled “Correcting Data Identified by the SQL Validations”, Validation 2.
U.2: Authorized Unit changed from previous version	Non-Fatal	See section entitled “Correcting Data Identified by the SQL Validations”, Validation 2.
D.3: Authorized Department Exclude flag changed from previous version	Non-Fatal	See section entitled “Correcting Data Identified by the SQL Validations”, Validation 3.
U.3: Authorized Unit Exclude flag changed from previous version	Non-Fatal	See section entitled “Correcting Data Identified by the SQL Validations”, Validation 3.
D.4: More than one Excluded Authorized Department with Spending Limit	Fatal	See section entitled “Correcting Data Identified by the SQL Validations”, Validation 4. This condition may cause the job to fail even if a request is made to bypass all validation failures.
U.4: More than one	Fatal	See section entitled “Correcting Data Identified by the

Condition	Fatal/ Non-Fatal	Remediation
Excluded Authorized Unit with Spending Limit per Authorized Department		SQL Validations", Validation 4. This condition may cause the job to fail even if a request is made to bypass all validation failures.

The following are the steps performed and log messages generated.

Process Steps	Messages
1. Report Aspect	• Validation report started. [Signifies the section when we are starting a report for possible failed validation check in SQLs.] • Rendering report started. [When we have finished all of the validation SQL checks and we need to write out the report.] • Rendering report completed. [When the report has been completed.]
2. Running Validation SQLs	• Performing validation ##. [Signifies the validation number being processed.] • Validation checks complete. [Issued when we have finished all Validation SQL checks.]

Remove duplicate Authorized Departments / Authorized Units

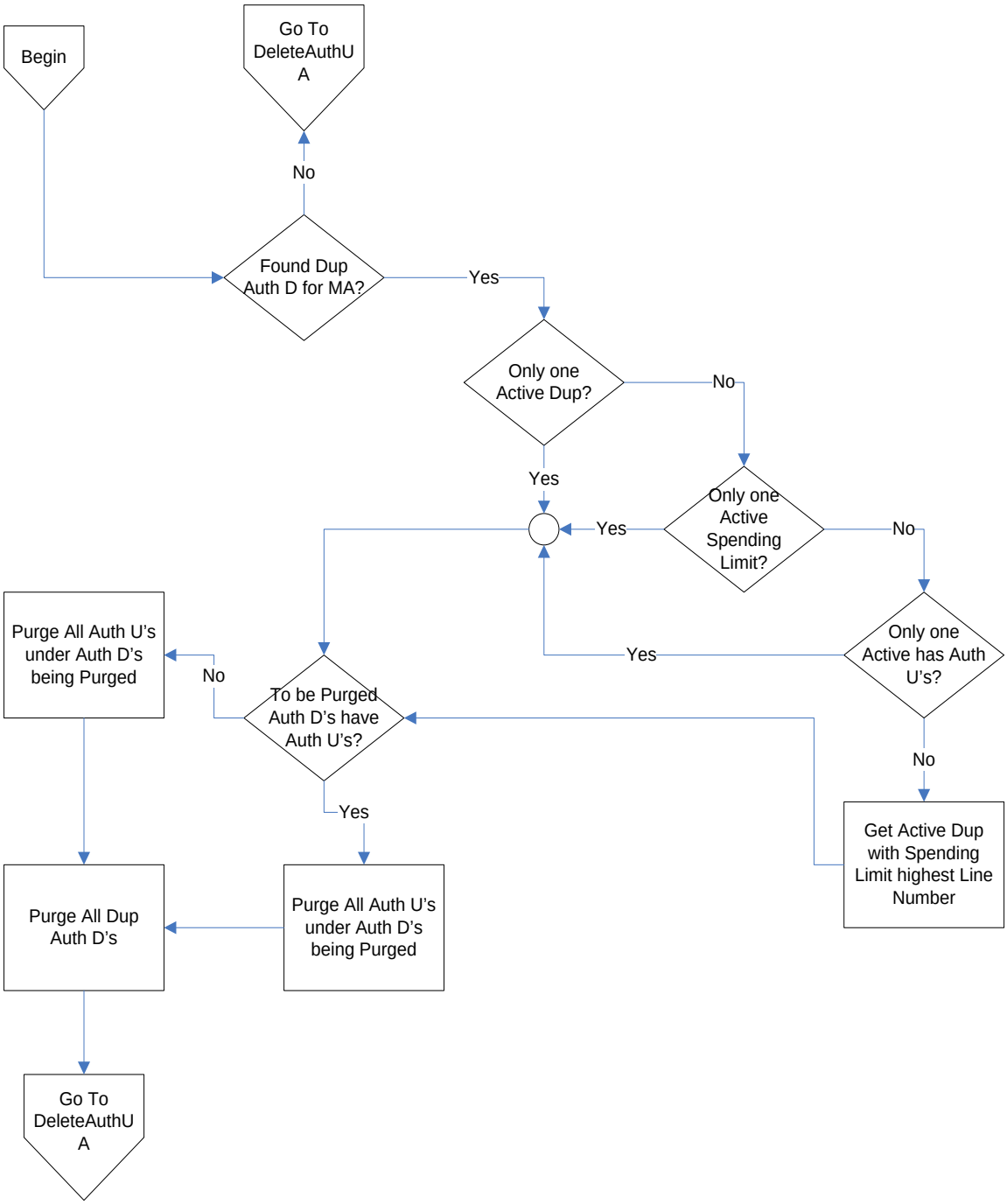
This step has no impact if the previous step found no duplicate Authorized Departments or Authorized Units. (Error D1 or U1 indicates that duplicates *do* exist.)

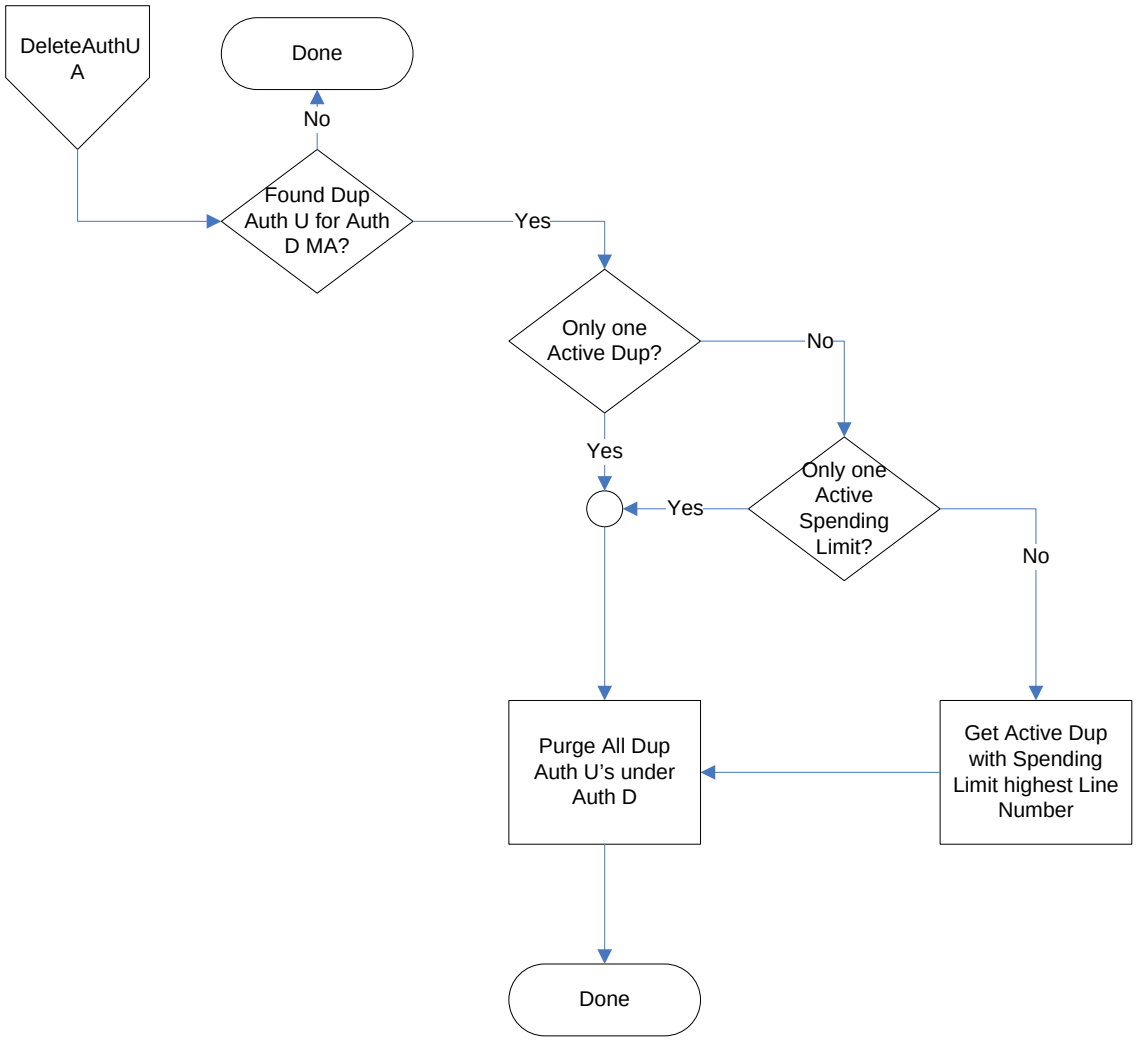
If duplicate Authorized Departments or Units were found, then the Validation Mode parameter controls whether this step runs or not. If the Validation Mode is set to 2 (Bypass All), all duplicate Authorized Departments and Authorized Units are automatically deleted from Master Agreements. If the Validation Mode is "0" or "1" then the job will fail in the prior step, so duplicate entries will not be automatically deleted.

This diagram shows, in a pictorial flow, how the routine decides which duplicate department(s) / unit(s) to delete. Generally, the process deletes duplicate Departments and Units in this order:

1. Deletes any duplicate inactive lines are deleted and retains active lines.
2. Deletes duplicate lines without spending limits and retains lines with non-zero spending limits.
3. Finally, all remaining duplicate lines are removed in order of line number, keeping only the highest line number (the one most recently added).

When the process deletes a duplicate Authorized Department record, it also deletes all of the Authorized Unit records defined as "children" to the record.


Begin
Go To DeleteAuthUA
No
Found Dup Auth D for MA?
Yes
Only one Active Dup?
No
Yes
Only one Active Spending Limit?
Yes
No
Purge All Auth U's under Auth D's being Purged
No
To be Purged Auth D's have Auth U's?
Yes
Only one Active has Auth U's?
Yes
No
Get Active Dup with Spending Limit highest Line Number
Purge All Dup Auth D's
Purge All Auth U's under Auth D's being Purged
Go To DeleteAuthUA



The following are the steps performed and log messages generated. (Note: "Auth" is used throughout as an abbreviation for "Authorized" and "Dept" as an abbreviation for "Department".)

Process Steps	Messages
1. Indication of process start	• Starting MA Transaction fixes. [To mark the time when the process started.]
2. Deleting duplicate Authorized Department lines (and any dependent Authorized Unit lines)	• Number of rows deleted from Auth Unit table (D1A): ##. [The number of Auth Unit rows deleted for which the Auth Dept is inactive and is a duplicate of any active Auth Dept (all child Auth Units of Auth Depts to be deleted in the next step).] • Number of rows deleted from Auth Dept table (D1B): ##. [The number of deleted inactive Auth Dept rows that are duplicates of any active Auth Dept.] • Number of rows deleted from Auth Unit table (D1C): ##. [The number of Auth Unit rows deleted for which the Auth Dept has no spending limit and is a duplicate of any Auth Dept with a

Process Steps	Messages
	spending limit (all child Auth Units of Auth Depts to be deleted in the next step).] • Number of rows deleted from Auth Dept table (D1D): ##. [The number of deleted Auth Dept rows with no spending limit that are duplicates of any Auth Dept with a spending limit.] • Number of rows deleted from Auth Unit table (D1E): ##. [The number of Auth Unit rows deleted for which the Auth Dept is a duplicate of any Auth Dept with a higher Auth Dept Line Number (all child Auth Units of Auth Depts to be deleted in the next step).] • Number of rows deleted from Auth Dept table (D1F): ##. [The number of deleted Auth Dept rows that are not the duplicate with the highest Auth Dept Line Number.]
3. Deleting duplicate Authorized Unit lines	• Number of rows deleted from Auth Unit table (U1A): ##. [The number of deleted inactive Auth Unit rows that are duplicates of any active Auth Unit.] • Number of rows deleted from Auth Unit table (U1B): ##. [The number of deleted Auth Unit rows with no spending limit that are duplicates of any Auth Unit with a spending limit.] • Number of rows deleted from Auth Unit table (U1C): ##. [The number of deleted Auth Unit rows that are not the duplicate with the highest Auth Unit Line Number.]
4. Indication of process completion	• MA Transaction fixes complete. [When the process has been completed.]

Reset Fields and Tables

The last section inside of this job will only run if all errors were resolved or if errors occurred and the Validation Mode parameter is “1” or “2.” (However, if the Validation Mode parameter is “2,” and fatal errors occur – such as with condition 4, this step will still fail.) This section updates Master Agreement Components and recreates the AUTHREF and AUTHREFU tables.

The order of SQL Updates is as follows:

1. Update of Master Agreement Commodity Component records
   1.1. Initialize all final, non-cancelled versions of MA Commodity records by setting the fields that will be updated to zero.
   1.2. Set the final, non-cancelled version of the MA Commodity record’s Ordered Amount and Quantity fields to the current value (zero) + the sum of all final, non-cancelled PO’s that reference this MA Commodity line.
   1.3. Set the final, non-cancelled version of MA Commodity record’s Expended Amount and Quantity fields to the current value (zero) + the sum of all final, non-cancelled PR’s that reference this MA Commodity line.
   1.4. Set all pending and draft versions of MA Commodity records with the amounts and quantities from the associated final version MA Commodity record.
   1.5. Commit all of the work performed on the MA Commodity line records.

Process Steps	Messages
Update Master Agreement Commodity Records	• Number of MA_DOC_COMM records updated to initialize balances: ### • Number of MA_DOC_COMM records updated from PO_DOC_COMM for Final version: ### • Number of MA_DOC_COMM records updated from PR_DOC_COMM for Final version: ### • Number of MA_DOC_COMM records updated from PR_DOC_ACTG for Final version: ### • Number of MA_DOC_COMM records updated for Modified Draft/Pending version: ###

2. Update of Master Agreement Vendor Component records

   2.1. Update all final, non-cancelled versions of the MA Vendor records' Ordered and Expended Amounts by performing a summation of all of the MA Commodities hanging from the vendor.

   2.2. Update all of the draft and pending modified versions by copying the Ordered and Expended Amounts for the Vendor component from the final version into the draft and pending versions.

   2.3. Commit all of the work performed on the Vendor Component records.

Process Steps	Messages
Update Master Agreement Vendor Records	• Number of MA_DOC_VEND records updated for Final version: ### • Number of MA_DOC_VEND records updated for Modified Draft/Pending version: ###

3. Update of Master Agreement Header Component records

   3.1. Update all final, non-cancelled versions of the MA Header records' Ordered and Expended Amounts by performing a summation of all of the MA Vendors hanging from the Header.

   3.2. Update all of the draft and pending modified versions by copying the Ordered and Expended Amounts from the Final version of the Header into the Draft and Pending versions.

   3.3. Commit all of the work performed on the Header Component records.

Process Steps	Messages
Update Master Agreement Header Records	• Number of MA_DOC_HDR records updated for Final version: ### • Number of MA_DOC_HDR records updated for Modified Draft/Pending version: ###

4. Update Master Agreement Authorized Department Component records

   4.1. Update final, non-cancelled version of MA Authorized Departments for excluded records, setting the Ordered and Expended Amounts to be zero.

   4.2. Update final, non-cancelled version of MA Authorized Departments for Included Specific Department records, setting the Ordered Amount equal to the Sum of Final, non-cancelled Purchase Order and Payment Request Commodity Lines that **directly** reference the MA where the Purchase Order or Payment Request Transaction Department is the same as the Included Department.

   4.3. Update final, non-cancelled version of MA Authorized Departments for Included Specific Department records, setting the Expended Amounts equal to the Sum of Final, non-cancelled Payment Request Commodity Lines that reference the MA directly or reference MA Orders. Matching records are found using the Transaction Department of the Payment Request (if it is a direct purchase) or the referenced Purchase Order's Transaction Department (if the PR references a Purchase Order).

   4.4. Update of the final, non-cancelled Version of MA Authorized Departments for Included ALL Department record by setting the field to zero. (These amounts will be updated in a later step of the logic.)

   4.5. Update draft and pending versions from the final version.

   4.6. Commit all of the work performed on the Authorized Department records.

Process Steps	Messages
Update Master Agreement Authorized Department Records	• Number of MA_DOC_AUTHDEPT records updated for Final version for Specific Department Exclude: ### • Number of MA_DOC_AUTHDEPT records updated for Final version from PO_DOC_COMM for Specific Department Include: ### • Number of MA_DOC_AUTHDEPT records updated for Final version from PR_DOC_ACTG for Specific Department Include: ### • Number of MA_DOC_AUTHDEPT records updated for Final version from PO_DOC_COMM for ALL Department Include: ### • Number of MA_DOC_AUTHDEPT records updated for Final version from PR_DOC_ACTG for ALL Department Include: ### • Number of MA_DOC_AUTHDEPT records updated for Modified Draft/Pending version: ###

5. Update Master Agreement Authorized Unit Component records

   5.1. Update final, non-cancelled version of MA Authorized Units for excluded records mapped to Specific Included Departments, setting the Ordered and Expended Amounts to zero.

   5.2. Update final, non-cancelled version of MA Authorized Units for Included ALL Unit, setting the Ordered / Expended Amounts to zero. (These amounts will be updated in a later step of the logic.)

   5.3. Update final, non-cancelled version of MA Authorized Units for Included Specific Unit mapped to an Included Specific Authorized Department, setting the Ordered Amounts

equal to the Sum of final, non-cancelled Purchase Order and Payment Request Commodity Lines that **directly** reference the MA where the Purchase Order or Payment Request Transaction Unit is the same as the Included Unit.

5.4. Update final, non-cancelled version of MA Authorized Units for Included Specific Unit mapped to an Included Specific Authorized Department, setting the Expended Amounts equal to the Sum of final, non-cancelled Payment Request Commodity Lines that reference the MA directly or reference MA Orders. Matching records are found using the Transaction Unit of the Payment Request (if it is a direct purchase) or the referenced Purchase Order's Transaction Unit (if the PR references a Purchase Order).

5.5. Update draft and pending versions from the final version..

5.6. Commit all of the work performed on the Authorized Unit records.

Process Steps	Messages
Update Master Agreement Authorized Unit Records	• Number of MA_DOC_AUTHUNIT records updated for Final version, Specific Include Department and Specific excluded Unit: ### • Number of MA_DOC_AUTHUNIT records updated for Final version from PO_DOC_COMM, ALL Include Department and ALL Include Unit: ### • Number of MA_DOC_AUTHUNIT records updated for Final version from PR_DOC_ACTG, ALL Include Department and ALL Include Unit: ### • Number of MA_DOC_AUTHUNIT records updated for Final version from PO_DOC_COMM, Specific Include Department and Specific Include Unit: ### • Number of MA_DOC_AUTHUNIT records updated for Final version from PR_DOC_ACTG, Specific Include Department and Specific Include Unit: ### • Number of MA_DOC_AUTHUNIT records updated for Final version from PO_DOC_COMM, Specific Include Department and ALL Include Unit: ### • Number of MA_DOC_AUTHUNIT records updated for Final version from PR_DOC_ACTG, Specific Include Department and ALL Include Unit: ### • Number of MA_DOC_AUTHUNIT records updated for Final version, Units under Department Exclude: ### • Number of MA_DOC_AUTHUNIT records updated for Modified Draft/Pending version: ###

6. Rebuild MA Reference for Authorized Departments (R_AUTHREF) table

6.1. Purge the R_AUTHREF table of all records

6.2. Acquire all final, non-cancelled Transaction Department Codes for transaction types "PO" and "PR" that contain a reference (direct or indirect) to an MA where the MA is final and non-cancelled. From this group, insert one new record on R_AUTHREF for each department if both of the following are true:

- The MA contains an "ALL" entry for Authorized Department or an Excluded Department.

- The Transaction Department of a specific record (for transactions that reference the MA directly) or the referenced Order's Transaction Department Code (for PRs that reference a PO) is not specifically Included or Excluded on the MA Authorized Department list.

6.3. Update all of the R_AUTHREF records by acquiring the Spending Limit either from the Excluded record with a spending limit or from the ALL Include record with a spending limit.

6.4. Update the Ordered Amounts by summing all of the final, non-cancelled Purchase Order and Payment Request Commodity Lines that **directly** reference the MA where the Purchase Order or Payment Request Transaction Department is the same as the R_AUTHREF Department code.

6.5. Update the Expended Amounts by summing all of the final, non-cancelled Payment Request Commodity Lines that reference the MA directly or reference MA Orders. Match R_AUTHREF records using the Transaction Department of the Payment Request (if it is a direct purchase) or the referenced Purchase Order's Transaction Department (if the PR references a Purchase Order).

6.6. Update the Department Name field by acquiring the Department Name from the Department table.

6.7. Commit all of the work performed on the R_AUTHREF records.

Process Steps	Messages
Rebuild AUTHREF table	• Number of AUTHREF entries deleted: ### • Number of AUTHREF entries inserted: ### • Number of AUTHREF entries updated: ###

7. Rebuild MA Reference for Authorized Units (R_AUTHREFU) table

7.1. Purge the R_AUTHREFU table of all records

7.2. Acquire all final, non-cancelled Transaction Unit Codes for transaction types "PO" and "PR" that contain a reference (direct or indirect) to an MA where the MA is final and non-cancelled. From this group, insert one new record on R_AUTHREFU for each unit if the following are true:

7.2.1. If the Authorized Department is "ALL" then the Authorized Department must be "ALL" so each Unit that had orders will update AUTHREFU.

7.2.2. If the Authorized Department is specific and included, then if Authorized Unit is "ALL" each Unit that had orders will update AUTHREFU.

7.3. Update all of the R_AUTHREFU records by acquiring the Spending Limit either from the Excluded record with a spending limit or from the ALL Include record with a spending limit.

7.4. Update the Ordered Amounts by summing all of the final, non-cancelled Purchase Order and Payment Request Commodity Lines that **directly** reference the MA where the Purchase Order or Payment Request Transaction Department and Unit are the same as the R_AUTHREF Department and Unit codes.

7.5. Update the Expended Amounts by summing all of the final, non-cancelled Payment Request Commodity Lines that reference the MA directly or reference MA Orders. Match R_AUTHREFU records using the Transaction Department and Unit of the

Payment Request (if it is a direct purchase) or the referenced Purchase Order's Transaction Department and Unit (if the PR references a Purchase Order).

7.6. Update the Department Name field by acquiring the Department Name from the Department table.

7.7. Update the Unit Name field by acquiring the Unit Name from the Unit table.

7.8. Commit all of the work performed on the R_AUTHREFU records.

Process Steps	Messages
Rebuild AUTHREFU table	• Number of R_AUTHREFU entries deleted: ### • Number of R_AUTHREFU entries inserted: ### • Number of R_AUTHREFU updated: ###

8. Reset the Total Spending Limit on Master Agreement Header (**Note**: this field is not displayed, but exists on the Database.)

Process Steps	Messages
Reset Total Spending Limit	• Number of records for Spending limit on MA_DOC_HDR for Final Version updated to 0: ### • Number of records for Spending limit on MA_DOC_HDR for Final Version updated: ### • Number of records for Spending limit on MA_DOC_HDR for Modified Draft/Pending Version updated to 0:: ### • Number of records for Spending limit on MA_DOC_HDR for Modified Draft/Pending Version updated: ###

9. Update Renewal History (RN_HIST) table

9.1. The Renewal History record for the current Renewal Period, as listed on the Final version of the MA Header, will have the Ordered and Expended amounts set from the Final version of the MA. The calculation will be:

- Current Renewal Period = Final Version MA Header Amount [Ordered / Expended] – the Sum Amount [Ordered / Expended] of all non-current Renewal Periods mapped to the current MA Header.

Process Steps	Messages
Update Renewal History table	• Number of current RN_HIST records refreshed from MA_DOC_HDR: ### • Number of current RN_HIST updated from non-current RN_HIST records: ###

10. Purge the Master Agreement Pending table

Process Steps	Messages
Purge the Master	• Number of R_MA_PEND_BAL entries deleted: ###

Process Steps	Messages
Agreement Pending table	

A critical item here is that a commit inside of the database is performed at the end of each of the above main sections, to ensure that we do not have a memory problem in the database.

Restart Capability

This job does not have restart capability. If the job fails, the problem must be identified and rectified. The job can be re-run again because it recalculates amounts with each run. Because the last step performs intermittent commits, a failure in the last step will created inconsistent data until the job is successfully completed.

Major Input

This section discusses the major inputs into this Data Conversion. It contains two main sections:

- Parameters
- Tables

Parameters

The following are the parameters for this job.

Parameter Name	Description	Default Value
CLIENT_NM	Optional Field. This is the name of the client that should show up on the Report for Validation Check.	-
VALIDATION_MODE	Required. ‘0’ is the default value, which performs all validations. ‘1’ specifies to bypass non-fatal checks. ‘2’ signifies that all validations can be bypassed.	0
COMMIT_BLOCK	Required. This is used for the reporting aspect in the event that there are a large number of values to be reported upon.	1000

The VALIDATION_MODE is an important parameter. Generally, the job should be run with value ‘0’. The main reason is that if the validations are bypassed, some of the SQL statement updates will fail under certain conditions. This parameter exists to give flexibility, in the chance that there is a need to bypass some or all of the validations for some unforeseen reason.

The COMMIT_BLOCK is also important, but is only used in the validation aspect of this job. Commits are performed after the main sections of the reset logic is performed, which does not take into consideration how many records were modified. However, if we are reporting on the validations and there are a large number of records to be reported upon, if we do not perform an intermittent commit, we may have memory problems. Thus, this parameter is only used in reporting on the validation items.

Tables

This following list includes the main tables used for input into this Data Conversion.

- Purchase Order Transaction
    - Commodity Line (PO_DOC_COMM)
    - Accounting Line (PO_DOC_ACTG)
- Payment Request Transaction
    - Commodity Line (PR_DOC_COMM)
    - Accounting Line (PR_DOC_ACTG)
- Master Agreement Transaction
    - Header (MA_DOC_HDR)
    - Vendor (MA_DOC_VEND)
    - Authorized Department (MA_DOC_AUTHDEPT)
    - Authorized Unit (MA_DOC_AUTHUNIT)
- MA Reference for Authorized Departments (R_AUTHREF)
- MA Reference for Authorized Units (R_AUTHREFU)
- Renewal History (RN_HIST)
- Department (R_DEPT)
- Unit (R_UNIT)

Major Output

There are two forms of major output:

- Report - The report is a listing of any validations that failed. The report helps identify the Master Agreements that must be corrected before the conversion can run successfully. The report is stored in the RTFiles folder, ReportOutput\FixOrderedAmtForMA\...\ FixOrderedAmtForMA.pdf for the pdf file. Appropriate staff should use the report's listing of failed validations and make corrections following the guidelines listed below (see "Correcting Data Identified by the SQL Validations")
- Updated and Rebuilt Tables

    The tables updated or rebuilt are:

    - Master Agreement
    - Header (MA_DOC_HDR)
        - Expended Amount (EXPND_AM)
        - Inception to Date Closed Amount – a hidden field (ITD_CLSD_AM)
        - Inception to Date Orders – a hidden field (ITD_ORD_AM)
        - Ordered Amount (ORD_TOT_AM)
        - Total Spending Limit – a hidden field (TOT_SPND_LMT_AM)
    - Vendor (MA_DOC_VEND)

        - Expended Amount (EXPND_AM)
        - Ordered Amount (VEND_ORDERED_AM)
    - Authorized Department (MA_DOC_AUTHDEPT)
        - Expended Amount (EXPND_AM)
        - Ordered Amount (ORD_AM)
        - Possible deletion of duplicate Authorized Departments
    - Authorized Unit (MA_DOC_AUTHUNIT)
        - Expended Amount (EXPND_AM)
        - Ordered Amount (ORD_AM)
        - Possible deletion of duplicate Authorized Units under a single Authorized Department
        - Possible deletion of Authorized Units hanging from a deleted Authorized Department
    - Commodity (MA_DOC_COMM)
        - Expended Amount – a hidden field (EXPND_AM)
        - Ordered Amount – a hidden field (RFED_ORD_AM)
        - Total Expended Item Quantity (EXPND_QTY)
        - Total Expended Service Contract Amount (EXPND_CNTRC_AM)
        - Total Ordered Item Quantity (RFED_QTY)
        - Total Ordered Service Contract Amount (RFED_CNTRC_AM)
        - MA Reference for Authorized Departments (R_AUTHREF)
- Purged and rebuilt
    - MA Reference for Authorized Units (R_AUTHREFU)
- Purged and rebuilt
    - Renewal History (RN_HIST)
        - Ordered Amount (ORD_AM)
        - Expended Amount (CLSD_AM)
    - MA Pending Balances (R_MA_PEND_BAL)
- Purged

Job Return Codes

The following job return codes are the expected outcome of the Data Conversion Job.

Return Code	Condition
Successful (1)	Tables validated successfully. Successfully updated / inserted / deleted records as desired.
Warning (4)	Currently nothing in this job would return this code.

Return Code	Condition
Non Fatal Error (8)	Currently nothing in this job would return this code.
Failed (12)	The job fails under one of the following conditions: • Parameters are invalid • The validations performed were not successful • Failure was encountered trying to update / insert / delete the tables listed above • Fatal error was experienced trying to perform some block of work
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

There is no sorting on the Inserting / Updating / Deleting of records. The only sorting performed is going to surround the return of records inside of the validation step. The records to be returned from all of the validation queries are sorted by:

- Transaction Code
- Transaction Department Code
- Transaction ID

Selection Criteria

The Insert / Update / Delete conversion process selects all Master Agreement, Purchase Order, and Payment Request records in the Transaction Catalog. There is specific selection logic for the validation SQL, which is described in the section above “Identify Known Failure Conditions”.

Post-Processing Tasks

The temporary indices should be dropped upon successful completion of the conversion process.

Problem Resolution

The series of high level steps that are performed inside of the batch job are:

1. Validate and Initialize values
2. Identify known failure conditions
3. Reset fields and tables

Any one of these steps could result in a return code of Successful, Failure, Terminated, or System Failure. (This job does not produce Warning or Non-Fatal Error return codes.) If the job returns any return code besides Successful, review the Job Log and the Failure Condition. The Failure Condition may indicate the problem by itself - see the table below. If the Failure Condition is not enough, compare the job log messages that appear to the ones listed above for each step

of the job's logic. Based on the messages produced just before failure, you should be able to determine which step of the job failed.

The following table provides the set of possible return codes and steps to be taken.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully. None of the failure conditions were found. Everything inserted / updated / deleted successfully and all was committed into the Database.	N/A	N/A
Warning (4)	N/A (This return code will not occur for this step.)	N/A	N/A
Non Fatal Error (8)	N/A (This return code will not occur for this step.)	N/A	N/A
Failed (12)	Job failed due to Fatal conditions.	If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and schedule a new job.	This job will rebuild and reset all of the values, so it can be run many times.
	Job Parameter failed validation.	**Recommendation:** Schedule a new job supplying the correct values or required values as listed in the Failed Job's Log.	N/A
	Job failed due to Validation Condition 1, the message will be: "MA Authorized Department with duplicate Department values", or "MA Authorized Unit with duplicate Unit values".	See section entitled "Correcting Data Identified by the SQL Validations" Validation 1.	After performing these changes and rectifying any other issue, re-run the job.

Possible Return Codes	Condition	Recommendation	Other Instructions
	Job failed due to Validation Condition 2, message will be: "MA Authorized Department with Department values changed from previous version", or "MA Authorized Unit with Unit values changed from previous version"	See section entitled "Correcting Data Identified by the SQL Validations" Validation 2.	After performing these changes and rectifying any other issue, re-run the job.
	Job failed due to Validation Condition 3, message will be: "MA Authorized Department with Department values with Exclude Flag changed from previous version", or "MA Authorized Unit with Unit values with Exclude Flag changed from previous version"	See section entitled "Correcting Data Identified by the SQL Validations" Validation 3.	After performing these changes and rectifying any other issue, re-run the job.
	Job failed due to Validation Condition 4, message will be: "MA Authorized Department with 2 Department values that have Exclude Flag checked and Spending Limit > 0", or "MA Authorized Unit with 2+ Unit values that have Exclude Flag checked and Spending Limit > 0"	See section entitled "Correcting Data Identified by the SQL Validations" Validation 4.	Re-run the job after rectifying this problem.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated; schedule a new job.	N/A
System Failure (20)	When the job is terminated because of database server	The reason for the System Failure needs to be investigated; schedule a new	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
	or network issues.	job.	

Correcting Data Identified by the SQL Validations

The resolutions suggested here can be involved. For some cases (validations D2, D3, U2, and U3) the resolution may not be worth the effort since the problems will not stop the updates from occurring. Careful consideration and planning is suggested for these fixes, as the possibility exists that an improper fix could cause further errors!

The resolutions are consistent for the same validations at the Department and Unit levels, so for each validation condition they are discussed together.

Validation 1 – Duplicate entries on Authorized Department/Unit lines

D1. Duplicate Departments in the Master Agreement Authorized Department Component

U1. Duplicate Units in the Master Agreement Authorized Unit Component mapped to the same Department Code

Resolution Steps:

Note: There are two options for resolution. One option is to perform these steps manually, with someone who is familiar with the Master Agreement making decisions about which records to delete. The other option is to allow the job to automatically delete duplicate records as described earlier in the run sheet in the section "Remove duplicate Authorized Departments / Authorized Units." *You only need to follow these steps if you choose to do the manual deletion.*

1. Review the Master Agreement online. Since there are multiple records with the same Department or Unit, someone must identify which one is the "correct" line that should be kept. Some suggestions for determining this:
    a. Do the lines have separate values for the "active" flag? One may be correct.
    b. Do the lines have different values for the spending limits? Is one of those values correct?
    c. Do the Department lines have Authorized Units defined underneath them? (Authorized units are tied to an Authorized Department line at the database level).
2. Once someone has identified the "correct" line, the other lines must be deleted. Note that the Authorized Department lines have a visible line number (online) but the Authorized Unit does not.
    a. The lines to be deleted must be deleted at the database level – it is not possible to delete them online.
    b. The Departments must be deleted from MA_DOC_AUTHDEPT.
    c. The Units must be deleted from MA_DOC_AUTHUNIT.

Validation 2 – code values for the same authorized line changed from one version to another

D2. Master Agreement Authorized Department Code changed from one version to another

U2. Master Agreement Authorized Unit Code changed from one version to another

Resolution Steps:

Note: Data correction is not necessarily required. When the SQL process runs to recalculate the Authorized Department and Unit amounts, it will calculate balances for only the Department and Unit codes that appear on the latest "Final" version of the transaction. If the codes on the latest current version of the Master Agreement are accurate, then no corrective steps are needed. Balances will not reflect the "historical" use of the Master Agreement, but the amounts will be correct for controlling future use of the Master Agreement.

If corrections are desired, then the following steps apply:

1. If the Department/Unit that was previously on the line did not perform any orders while they were allowed to, then no action is needed.
2. If the Department/Unit that was previously on the line did execute orders and those orders can not be canceled, then one of two actions are needed, depending on whether Authorized Departments and Units are defined by inclusion or exclusion logic. (Each Master Agreement can only use one method.)
    a. If the Master Agreement uses the "include" method then:
        i. Add the Department/Unit as a new Authorized line
        ii. Set the Spending Limit to the amount that was ordered
        iii. Verify that the No Limit flag is unchecked. (This will stop that Department/Unit from ordering any additional items from the MA.)
    b. If the Master Agreement uses the "exclude" method, then the MA must be changed to use the "include" method. In this case, manual modification of the MA is needed to enter Authorized lines for all of the Departments and Units that are authorized (after deleting the Exclusion lines). For those Departments/Units that ordered in the past but should not be allowed to order any more, follow the same steps listed in the prior bullet:
        i. Add the Department/Unit as an "included" Authorized line
        ii. Set the Spending Limit to the amount that was ordered
        iii. Verify that the No Limit flag is unchecked.

Validation 3 – exclusion flags have changed from one version to another

D3. Master Agreement Authorized Department changed Exclude Flag from one version to next

U3. Master Agreement Authorized Unit changed Exclude Flag from one version to next

Resolution Steps:

Note: Data correction is not necessarily required. When the SQL process runs to recalculate the Authorized Department and Unit amounts, it will calculate balances for only the Department and Unit codes that are *included* on the latest "Final" version of the transaction. If the exclusion flags on the latest current version of the Master Agreement are correct, then no corrective steps are needed. Balances will not reflect the "historical" use of the Master Agreement, but the amounts will be correct for controlling future use of the Master Agreement.

If corrections are desired, then the following steps apply:

1. If the record changed from an include to exclude, but orders were previously placed by that Department/Unit that is now excluded, that means that the Master Agreement was changed at one point from an "inclusion" MA to an "exclusion" MA. For the amounts to update properly, the MA must be changed back to an "inclusion" MA.
    a. In this case, manual modification of the MA is needed to enter Authorized lines for all of the Departments and Units that are authorized (after deleting the Exclusion lines).
    b. Once the entries are changed to "inclusion" records, you can stop further use by that Department or Unit by doing the following:

   i. Set the Spending Limit to the amount that was already ordered
   ii. Verify that the No Limit flag is unchecked. (This will stop that Department/Unit from ordering any additional items from the MA.)

2. If the flag was changed from exclude to include, and the Department/Unit should truly be included, then no action is required.

Validation 4 – Two ore more excluded Authorized Department/Unit lines have non-zero spending limits

D4. Two or more Authorized Department records with non-zero Spending Limit

U4. Two or more Authorized Unit records with non-zero Spending Limit mapped to the same Authorized Department record

Resolution Steps:

1. If a line is excluded and has a spending limit, then the system meaning is that the limit listed applies to all other departments besides the one excluded. *The resolution is to reset the spending limits for all excluded lines except the one that has the correct limit amount to zero.*

2.1.18 Generate Ledger Hash Code Batch Process

Description

A new field Concat Keys Hash Code (CONCAT_KY_HSH_CD) was added to all Ledger tables during Parallelization enhancement to Ledger Engine. This field will store Hash Code generated for the combination of data in Concat key fields (CONCAT_KYn). The Generate Ledger Hash Code Conversion program populates Hash Code for Concat Keys for the existing records on Ledger tables. This conversion program should not be run when any other processes that can potentially perform insert/update/delete actions on Ledger tables are running.

When to Run

This process will be run after Database Upgrade Scripts for the build are executed.

Major Input

All the Ledger Tables will be the input to this process.

- General Use Ledger #1 (LDGR_001)
- General Use Ledger #2 (LDGR_002)
- General Use Ledger #3 (LDGR_003)
- General Use Ledger #4 (LDGR_004)
- General Use Ledger #5 (LDGR_005)
- General Use Ledger #6 (LDGR_006)
- General Use Ledger #7 (LDGR_007)
- General Use Ledger #8 (LDGR_008)
- General Use Ledger #9 (LDGR_009)
- General Use Ledger #10 (LDGR_010)
- Accounting by Accounting Period Ledger (LDGR_APD_ACTG)
- Cost Accounting by Accounting Period Ledger (LDGR_APD_CA)
- Accounting by Budget Fiscal Year Ledger (LDGR_BFYTD_ACTG)
- Cost Accounting by Budget Fiscal Year Ledger (LDGR_BFYTD_CA)
- Accounting by Fiscal Year Ledger (LDGR_FYTD_ACTG)
- Cost Accounting by Fiscal Year Ledger (LDGR_FYTD_CA)
- Accounting Inception-to-Date Ledger (LDGR_ITD_ACTG)
- Cost Accounting Inception-to-Date Ledger (LDGR_ITD_CA)
- Ledger for Budget Systems Assurance (LDGR_SA_BUD)
- Full Detail Accounting Ledger (LDGR_FYDAD)

Peripheral Input

- Journal/Ledger Control detail (R_JRNL_LDGR_CTRL)

Output

The field Concat Keys Hash Code (CONCAT_KY_HSH_CD) is populated on all Records of all Ledger Tables(listed as input) with a generated Hash Code based on the combined data of Concat key fields (CONCAT_KYn).

Parameters

Description (Caption)	Parameter Name	Default Value
Commit block Size. If not entered then defaulted to 1000. After processing Commit block Size number of Ledger records a commit is issued. Can be used for Performance tuning.	COMMIT_BLOCK	1000
Select block Size. If not entered then defaulted to 1000. It is the number of Ledger records that can be fetched as a single block. Can be used for Performance tuning	SELECT_BLOCK	1000

Information on Parameters

- Commit Block Size should be equal to or less than Select Block Size.
- Values of Select and Commit blocks sizes should be tuned as per the system processing and memory capacities.

Sort Sequence

Ledger tables retrieved to be updated are sorted by Journal/Ledger Id that is assigned to Ledger tables on Journal/Ledger Control detail table.

Selection Criteria

- Get all the Ledger tables sorted by Journal/Ledger Id from Journal/Ledger Control detail table.
- For each Ledger table
  - Select REC_NO, CONCAT_KY, CONCAT_KY2, CONCAT_KY3 from the Ledger table ordered by REC_NO.
  - Combine data in CONCAT_KY, CONCAT_KY2, CONCAT_KY3 fields and generate Hash Code for them.
  - Update Concat Keys Hash Code (CONCAT_KY_HSH_CD) with this generated Hash Code.

Problem Resolution

This program has checkpoint implemented. Information stored regularly in Checkpoint is the Ledger Table name and the last Record number(REC_NO) of last successful commit. If the

process fails for any reason (such as the network is down or the server is down, other than any technical errors like NullPointerException), then the process can be restarted and the process can restart from the last point where it stopped with the information retrieved from Checkpoint.

2.1.19 Posting Line Group Counter Conversion Process

Job Name	Posting Line Group Counter Conversion Process
Recommended Frequency	Once. Refer to the Steps Before Running Job and Steps after Completing the Job sections for important instructions that need to be followed before execution of the batch process.
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	No

Steps Before Running the Job

No transaction processing should occur when the Data Fix is in progress. The indices given in the next section should to be created.

Performance Suggestions

There are five non-unique indices that should be added to their respective tables before this job is run in order to maximize performance, and should be dropped after the conversion is complete. The indices should be created with the fields in the specified order. The examples are shown with a suggested index name, but any index name that the database will find valid can be used.

1. DOC_ACTG: This index will link Posting Line Group Counter to the Transaction. Information.

- DOC_TYP
- DOC_VERS_NO
- DOC_PHASE_CD
- DOC_FUNC_CD
- PSTNG_LN_GRP_CTR
- BFY

Example (Oracle syntax):

CREATE INDEX N1_DOCACTG_PSTNGLNGRP ON DOC_ACTG (DOC_TYP, DOC_VERS_NO, DOC_PHASE_CD, DOC_FUNC_CD, PSTNG_LN_GRP_CTR, BFY)

2. DOC_ACTG: This index is created for retrieving the transaction information.

- DOC_CD
- DOC_DEPT_CD
- DOC_ID
- DOC_PHASE_CD

- DOC_FUNC_CD

Example (Oracle syntax):

CREATE INDEX N2_DOCACTG_PSTNGLNGRP ON DOC_ACTG (DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_PHASE_CD, DOC_FUNC_CD)

3. PSTNG_LN_GRP: This index will link Posting Line Group Counter to the Transaction Information. This is used to find out whether a Posting Line exists for the problematic Accounting Line.

- DOC_CD
- DOC_DEPT_CD
- DOC_ID
- DOC_VERS_NO
- DOC_VEND_LN_NO
- DOC_COMM_LN_NO
- DOC_ACTG_LN_NO
- LN_FUNC_CD

Example (Oracle syntax):

CREATE INDEX N1_PSTNGLNCAT_PSTNGLNGRP ON PSTNG_LN_CAT (DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, DOC_VEND_LN_NO, DOC_COMM_LN_NO, DOC_ACTG_LN_NO, LN_FUNC_CD)

4. PSTNG_LN_GRP: This index helps in getting maximum Posting Line Group number from the posting line catalog for FINAL / DRAFT / PENDING versions to be set on the Accounting Line.

- DOC_CD
- DOC_DEPT_CD
- DOC_ID
- DOC_VEND_LN_NO
- DOC_COMM_LN_NO
- DOC_ACTG_LN_NO
- DOC_PHASE_CD

Example (Oracle syntax):

CREATE INDEX N2_PSTNGLNCAT_PSTNGLNGRP ON PSTNG_LN_CAT (DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VEND_LN_NO, DOC_COMM_LN_NO, DOC_ACTG_LN_NO, DOC_PHASE_CD)

5. PSTNG_LN_GRP: This index links the referenced transaction information for the setting of posting line group counter on the liquidation posting lines.

- RFED_DOC_CD
- RFED_DOC_DEPT_CD
- RFED_DOC_ID
- RFED_VEND_LN_NO
- RFED_COMM_LN_NO
- RFED_ACTG_LN_NO
- LN_FUNC_CD
- DOC_PHASE_CD
- DOC_FUNC_CD

Example (Oracle syntax):

CREATE INDEX N3_PSTNGLNCAT_PSTNGLNGRP ON PSTNG_LN_CAT (RFED_DOC_CD, RFED_DOC_DEPT_CD, RFED_DOC_ID, RFED_VEND_LN_NO, RFED_COMM_LN_NO, RFED_ACTG_LN_NO, LN_FUNC_CD, DOC_PHASE_CD, DOC_FUNC_CD)

Overview

In release 3.3 fields were added to the Accounting and Posting Lines in order to track the 'newer' Posting Lines from 'older' ones. This aging was not to track simple dollar increases or decreases where the Line Amount field was the only field changed on a modification. The aging was to track changes in a modification where the system generated one or more posting lines to back out 'old accounting' that had been posted and then created one or more posting lines to book 'new accounting.' Examples of such modifications were changes to the commodity code of a parent Commodity Line, vendor code of a parent Vendor Line, or the changing of any COA values (seen or inferred). An additional Accounting Line change that creates 'new accounting' is the changing of BFY by one of the Open Activity Roll processes. This last item is the most common cause of 'new accounting.'

Not all document types use this tracking feature. Those without Accounting Lines (Budget, Invoice, and Receiver for example) and those that cannot be modified (Journal Vouchers for example) will never have a back out Posting Line and one to post 'new accounting.' For this reason the program will only process against certain transaction types: ABS, RQS, PO, ARE, and RE.

A defect was found where a modification transaction was processed for one Accounting Line, but not all Accounting Lines. In this case the field added for this tracking at the Accounting Line reset to zero and in doing so changed the value at the posting line too. At this point any future modifications to an Accounting Line reset to zero would start re-numbering. All aging to that point would be lumped into a single age group. That defect was fixed in 3.7, 3.4.1.2, 3.6.0.2, and 3.5.0.9 releases. ** If your site went live on any of these or a later release, then the defect will not have impacted any of your transactions.

The main impact the defect only became evident when an inverse reference was done to re-open some or all of what had been closed on the Accounting Line. If that Accounting Line had been rolled forward from a prior BFY to the current, the inverse reference would not know that the posting line for the current BFY should be opened first. The system would then apply the inverse reference to all posting lines that it thought were the newest, thus re-opening a posting line in a prior BFY. Proper BFY Staging setup would have likely prevented such an increase when it should not happen. For example, an inverse reference on a Commodity Encumbrance Correction

(CEC) transaction in BFY/FY 2008 re-opening a BFY 2007 Encumbrance in a late stage of 2007 (for example – the stage for August in a year that ended the previous June).

Others have reported incidents where a partial or final reference was not allowed against an Accounting Line where the aging process had been corrupted.

Thus the objective of this Data Fix is to clean up Final, Pending and Draft versions of transactions in the Accounting Line Catalog [DOC_ACTG] which experienced the corruption in aging. The Accounting Line field tracking age is the Posting Line Group Counter. The field starts off as 1 on the first version and then increments by one for each modification that created 'new accounting.' The Posting Line field tracking age is the Posting Line Group Number. This value is set from the Accounting Line which triggered the initial creation of the Posting Line. The value in this field starts as 1 on the first version. When a modification to change some value on the Accounting Line, say the BFY (or some other 'new accounting' field changed), happens, the Posting Line Group Counter of that new transaction version increases. The Posting Line Group Counter of the previous version remains unchanged. The Posting Line(s) on the modification that back out 'old accounting' from the prior version retain the old Posting Line Group Number. The Posting Line recording 'new accounting' has a Posting Line Group Number equal to the Posting Line Group Counter that was copied from the prior version Accounting Line and incremented by one. The following is a simple example:

Version 1: BFY 2007 & COA String 1
Accounting Line: Posting Line Group Counter = 1 $100
Posting Line 1: Posting Line Group Number = 1 $100

Version 2: BFY 2007 & COA String 1+ (Sub Object entered where left blank in version 1)
Accounting Line: Posting Line Group Counter = 2 $100
Posting Line 1: Posting Line Group Number = 1 ($100) w/COA String 1
Posting Line 2: Posting Line Group Number = 2 $100 w/COA String 1+

Version 3: BFY 2008 & COA String 1+ (rolled)
Accounting Line: Posting Line Group Counter = 3 $100
Posting Line 1: Posting Line Group Number = 1 ($100) w/COA String 1 BFY 2007 (this line would actually disappear if nothing was closed against it)
Posting Line 2: Posting Line Group Number = 2 ($100) w/COA String 1+ BFY 2007
Posting Line 3: Posting Line Group Number = 3 $100 w/COA String 1+ BFY 2008

Processing logic by the program is as follows:

1. **Accounting Catalog records' Posting Line Group Counter**

   Change of any number of the COA elements on the modification transaction affects the Posting Line Group Counter. Apart from the COA elements the other fields which may affect the Posting Line Group Counter (based upon the Event Category) are: BFY, VEND_CUST_CD, COMM_CD, BANK_ACCT_CD, DISB_CAT, BPRO_CD, WHSE_CD and STK_ITM_SFX. If none of these fields change from one version to the next in all versions of the Accounting Line, the Posting Line Group Counter remains 1 throughout all the versions. Even if one of these fields change, the resulting version will have Posting Line Group Counter as 2. Thus Posting Line Group counter increases for every version if the version had undergone a change in the combination of the COA and the above mentioned fields.

2. **Standard Posting Lines' Posting Line Group Number**

   The Posting Line Group on the Posting Lines also behaves similar to that of the Accounting Lines. If the existing combination of the COA elements and the above

mentioned fields change in a modification transaction, a new Posting Line is created apart from the existing one. The existing line will have Posting Line Group Counter as 1 and the other line will have Posting Line Group Counter as 2. Going further, in a newer version of the transaction, if the combination of the COA elements and the above mentioned fields change, 1 new Posting Line will be generated apart from the existing 2 which will have Posting Line Group Counter as 3.

3. Liquidation Posting Lines' Posting Line Group Number

The liquidation posting line will have the same Posting Line Group Counter as the one on the Standard Posting Line mapped to it. Thus if the standard Posting Line's Group counter changes, accordingly its liquidation posting line Group Counter will also change.

Process Steps	Messages
4. Parameter Validation	Validating Batch Parameters listed in the above Batch Parameter section The basic validations are: • BFY: If no value is provided, all the Budget Fiscal Years are processed. • PRE_33_DATA: Ensures that it is 0 or 1. If not an error is logged. • DOC_CDS: If no value is provided, process all DOC_CDS entered must belong to one value entered DOC_TYP • DOC_TYP: Ensures that at least 1 DOC_TYP is provided. If not, an error is logged. If a value is entered and if it is not of the DOC_TYP provided in batch parameters section, an error is logged. • SELECT_BLOCK_SZ: If no value is provided, it defaults to 500. If value provided and is not a number, an error is logged • PROG_CTR_SZ: If no value is provided, it defaults to 500. If value provided is not a number, an error is logged Batch Parameter validation completed.
5. Selection of Records whose FINAL Version needs to be modified	This selects the final version Accounting Catalog [DOC_ACTG] records with potential data problem whose posting line group counter is not set correctly.
6. Calculation and Updates of the FINAL Version Accounting and Posting Lines	The final standard Posting Lines corresponding to the selected problematic Accounting lines are updated with correct posting line group counter. Later on, final Accounting Lines are also updated
7. Updates of the Referencing Liquidation Posting Lines of the current select Errant Accounting Line	The liquidation Posting Lines corresponding to the selected problematic Accounting Lines are updated with correct posting line group counter. Later on Accounting Lines are also updated

8. Selection of Records whose PENDING Version needs to be modified	This selects the problematic Accounting Lines [DOC_ACTG] pending version records with potential data problem whose posting line group counter is not set correctly.
9. Calculation and Updates of the PENDING Version Accounting and Posting Lines	The Posting Lines corresponding to the selected problematic Accounting Lines are updated with correct posting line group counter. Later on, pending Accounting Lines are also updated
10. Selection of Records whose DRAFT Version needs to be modified	This selects the problematic Accounting Lines [DOC_ACTG] draft version records with potential data problem whose posting line group counter is not set correctly.
11. Calculation and Updates of the DRAFT Version Accounting and Posting Lines	The draft Posting Lines corresponding to the selected problematic Accounting Lines are updated with correct posting line group counter. Later on, draft Accounting Lines are also updated

Restartability information

The job commits after each problematic Accounting Line is processed. On restart, the job will start processing from the record on which it was terminated. The reason of job failure needs to be investigated by looking at the job error log and error log files.

Major Input

Transaction Accounting Line table (DOC_ACTG)

Posting Line Catalog table (PSTNG_LN_CAT)

Batch Parameters

Multiple runs are suggested to limit the workload and provide better selection criteria by transaction type. For example, if all open Pre Encumbrances are lapsed at year end so that only one BFY has any open activity, then a run with that BFY value transaction type RQ are suggested. The same can be said for Encumbrances as well, however if they are left open in the BFY which they originated and not rolled, then the BFY parameter would be blank and transaction type PO entered.

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

Parameter Name	Description	Default Value
BFY	Starting BFY Optional field. If left blank, then BFY will not be used in selection. If values are entered, they should only be for active budget years that still allow activity.	No default

PRE_33_DATA	Converted from Pre-3.3 Required field. This parameter should be set to 1 if this data was begun in a pre-3.3 version of the application and then converted through 3.3 and beyond. 0 if the data set was first begun in 3.3 or later version.	0
DOC_CDS	Transaction Codes Optional field. This should be a comma separated value list, if more than one transaction code supplied. This set of transaction codes must map to the supplied transaction types parameter value. If left blank will assume all transaction codes listed on Transaction Control (DCTRL) for supplied transaction types will be processed.	No default
DOC_TYP	Transaction Type Required field. This parameter is a list of the transaction types to be processed for this instance of the run. Only one transaction type should be supplied. If transaction codes are supplied this transaction type parameter must be associated with all of the transaction codes inside of Transaction Control (DCTRL). Values can only be one of the following: ABS, ARE, PO, RE, or RQ.	No default
SELECT_BLOCK_SZ	Select block Size Optional field. If not entered then defaulted to 10000. It is the number of records that can be fetched as a single block. Can be used for performance tuning.	500
PROG_CTR_SZ	Progression Counter Size Optional field. If not entered then defaulted to 10000. It is the number of records after which the progression counter message suggesting the number of records completed is to be written in the job log. It helps the user to track the number of records processed.	500

Major Output

1. Accounting Catalog records' Posting Line Group Counter is correctly populated.
2. Standard Posting Line Catalog records' Posting Line Group Counter is correctly populated.
3. Liquidation Posting Line Catalog records' Posting Line Group Counter is correctly populated.

Note that only FINAL / PENDING / DRAFT versions of the above tables are corrected.

Main Query to Identify the Problematic Accounting Lines:

The main query which is executed to fetch the problematic Accounting Lines is discussed in this section. The job parameter values are assumed for defining the query. As the job parameters change, the query should also be changed accordingly. The values given in the job parameters are assumed to be:

BFY : 2004

DOC_TYP : PO

PRE_33_DATA: either 0 or 1. Both the possibilities are discussed below.

Assuming that the job parameter for BFY is 2004 and DOC_TYP is 'PO', the purpose of this query is to fetch all the problematic Accounting Lines which are in FINAL / PENDING / DRAFT phase whose budget fiscal year is greater than 2004 and transaction type is 'ARE'. The query selects the Account Lines which has the Posting Line Group Counter field as either 0 or NULL and the associated posting lines are Standard Posting Lines.

The query contains Oracle Specific SQL Function calls, these must be changed so that they can run on the client specific Database Type.

- Query to fetch all the problematic Accounting Lines which are in FINAL Phase with the job parameter PRE_33_DATA as 1.
- Query to be run if the client went live before 3.3 and ran FES Posting Data Conversion to upgrade to newer version of application.

Example (Oracle syntax):

SELECT DISTINCT B.DOC_CD, B.DOC_DEPT_CD, B.DOC_ID, B.DOC_VEND_LN_NO, B.DOC_COMM_LN_NO, B.DOC_ACTG_LN_NO FROM DOC_ACTG B WHERE B.DOC_TYP IN ('PO') AND B.DOC_VERS_NO > NVL((

-- This will allow us to see which is the first version that a Group Counter was
-- introduced which is important for clients that had to run the FEC Posting Data
-- conversion Job

```
        SELECT MIN(C.DOC_VERS_NO) FROM DOC_ACTG C WHERE
        B.DOC_CD = C.DOC_CD AND B.DOC_DEPT_CD = C.DOC_DEPT_CD
        AND B.DOC_ID = C.DOC_ID AND B.DOC_VEND_LN_NO =
        C.DOC_VEND_LN_NO AND B.DOC_COMM_LN_NO =  C.DOC_COMM_LN_NO
        AND B.DOC_ACTG_LN_NO = C.DOC_ACTG_LN_NO AND C.DOC_PHASE_CD IN
        (3/*Final*/, 5/*Hist Final*/) AND C.DOC_FUNC_CD <> 3 AND
        C.PSTNG_LN_GRP_CTR > 0
    ), 1)
    AND B.DOC_PHASE_CD IN (3/*Final*/, 5/*Hist Final*/) AND B.DOC_FUNC_CD <>  3
    /*Cancelled*/ AND (B.PSTNG_LN_GRP_CTR IS NULL OR B.PSTNG_LN_GRP_CTR =
    0) AND B.BFY >= 2004 AND NOT EXISTS (
        -- This will find out if we are inside of a Transaction that is cancelled, which we should
        -- ignore cancelled transactions
        SELECT D.DOC_CD FROM DOC_ACTG D WHERE B.DOC_CD = D.DOC_CD AND
        B.DOC_DEPT_CD = D.DOC_DEPT_CD AND
        B.DOC_ID = D.DOC_ID AND D.DOC_PHASE_CD = 3 /*Final*/ AND
        D.DOC_FUNC_CD = 3 /*Cancelled*/)
    AND EXISTS(
        -- We only care about ALs that have Standard PLs, since Non-Standard will
        -- not be referenced
        SELECT E.DOC_CD FROM PSTNG_LN_CAT E WHERE B.DOC_CD =
        E.DOC_CD AND B.DOC_DEPT_CD = E.DOC_DEPT_CD AND B.DOC_ID
```

```
    = E.DOC_ID AND B.DOC_VERS_NO = E.DOC_VERS_NO AND
    B.DOC_VEND_LN_NO = E.DOC_VEND_LN_NO AND
    B.DOC_COMM_LN_NO = E.DOC_COMM_LN_NO AND
    B.DOC_ACTG_LN_NO = E.DOC_ACTG_LN_NO AND E.LN_FUNC_CD =1
     /*Standard*/
)
ORDER BY B.DOC_CD,
B.DOC_DEPT_CD,B.DOC_ID,B.DOC_VEND_LN_NO,B.DOC_COMM_LN_NO,B.DOC_
ACTG_LN_NO
```

Note: The Accounting and Posting Lines belonging to the HISTORICAL FINAL phase will not be updated by this process. Thus, the query may return results after this process is run to fix the corrupt records.

- Query to fetch all the problematic Accounting Lines which are in FINAL Phase with the job parameter PRE_33_DATA as 0.
- Query to be run if a client first went live in 3.3 and beyond.

Example (Oracle syntax):

```
SELECT DISTINCT B.DOC_CD, B.DOC_DEPT_CD, B.DOC_ID,
B.DOC_VEND_LN_NO, B.DOC_COMM_LN_NO, B.DOC_ACTG_LN_NO FROM
DOC_ACTG B WHERE B.DOC_TYP IN ('PO') AND B.DOC_VERS_NO > 1 AND
B.DOC_PHASE_CD IN (3/*Final*/, 5/*Hist Final*/) AND B.DOC_FUNC_CD <> 3
/*Cancelled*/ AND (B.PSTNG_LN_GRP_CTR IS NULL OR B.PSTNG_LN_GRP_CTR =
0) AND B.BFY >= 2004 AND NOT EXISTS (
    -- This will find out if we are inside of a Transaction that is cancelled, which we should
    --- ignore cancelled transactions.
    SELECT D.DOC_CD FROM DOC_ACTG D WHERE B.DOC_CD = D.DOC_CD AND
    B.DOC_DEPT_CD = D.DOC_DEPT_CD AND B.DOC_ID = D.DOC_ID AND
    D.DOC_PHASE_CD = 3 /*Final*/ AND D.DOC_FUNC_CD = 3 /*Cancelled*/
 ) AND EXISTS (
    -- We only care about ALs that have Standard PLs, since Non-Standard will not be
    -- referenced
    SELECT E.DOC_CD FROM PSTNG_LN_CAT E WHERE B.DOC_CD = E.DOC_CD
    AND B.DOC_DEPT_CD = E.DOC_DEPT_CD AND B.DOC_ID = E.DOC_ID AND
    B.DOC_VERS_NO = E.DOC_VERS_NO AND B.DOC_VEND_LN_NO =
    E.DOC_VEND_LN_NO AND B.DOC_COMM_LN_NO = E.DOC_COMM_LN_NO
    AND B.DOC_ACTG_LN_NO = E.DOC_ACTG_LN_NO AND E.LN_FUNC_CD = 1
    /*Standard*/
) ORDER BY B.DOC_CD,
B.DOC_DEPT_CD,B.DOC_ID,B.DOC_VEND_LN_NO,B.DOC_COMM_LN_NO,B.DOC_
ACTG_LN_NO
```

Note: The Accounting and Posting Lines belonging to the HISTORICAL FINAL phase will not be updated by this process. Thus, the query may return results after this process is run to fix the corrupt records.

- Query to fetch all the problematic Accounting Lines which are in PENDING phase.

Example (Oracle syntax):

```
SELECT DISTINCT B.DOC_CD, B.DOC_DEPT_CD, B.DOC_ID,
B.DOC_VEND_LN_NO, B.DOC_COMM_LN_NO, B.DOC_ACTG_LN_NO,
DOC_VERS_NO FROM DOC_ACTG B WHERE B.DOC_TYP IN ('PO') AND
B.DOC_VERS_NO > 1 AND B.DOC_PHASE_CD IN (2/*Pending*/) AND
B.DOC_FUNC_CD <> 3 /*Cancelled*/ AND (B.PSTNG_LN_GRP_CTR IS NULL OR
B.PSTNG_LN_GRP_CTR = 0) AND B.BFY >= 2004 AND NOT EXISTS (
    -- This will find out if we are inside of a Transaction that is cancelled, which we should
    --- ignore cancelled transactions.
    SELECT D.DOC_CD FROM DOC_ACTG D WHERE B.DOC_CD = D.DOC_CD AND
    B.DOC_DEPT_CD = D.DOC_DEPT_CD AND B.DOC_ID = D.DOC_ID AND
    D.DOC_PHASE_CD = 2 /*Pending*/ AND D.DOC_FUNC_CD = 3 /*Cancelled*/
) AND EXISTS (
    -- We only care about ALs that have Standard PLs, since Non-Standard will not be
    -- referenced
    SELECT E.DOC_CD FROM PSTNG_LN_CAT E WHERE B.DOC_CD = E.DOC_CD
    AND B.DOC_DEPT_CD = E.DOC_DEPT_CD AND B.DOC_ID = E.DOC_ID AND
    B.DOC_VERS_NO = E.DOC_VERS_NO AND B.DOC_VEND_LN_NO =
    E.DOC_VEND_LN_NO AND B.DOC_COMM_LN_NO = E.DOC_COMM_LN_NO
    AND B.DOC_ACTG_LN_NO = E.DOC_ACTG_LN_NO AND E.LN_FUNC_CD = 1
    /*Standard*/
) AND B.DOC_COMP_MOD_FL=1
ORDER BY B.DOC_CD,
B.DOC_DEPT_CD,B.DOC_ID,B.DOC_VEND_LN_NO,B.DOC_COMM_LN_NO,B.DOC_
ACTG_LN_NO, DOC_VERS_NO
```

- Query to fetch all the problematic Accounting Lines which are in DRAFT phase.

Example (Oracle syntax):

```
SELECT DISTINCT B.DOC_CD, B.DOC_DEPT_CD, B.DOC_ID,
B.DOC_VEND_LN_NO, B.DOC_COMM_LN_NO, B.DOC_ACTG_LN_NO,
DOC_VERS_NO FROM DOC_ACTG B WHERE B.DOC_TYP IN ('PO') AND
B.DOC_VERS_NO > 1 AND B.DOC_PHASE_CD IN (1/*Draft*/) AND B.DOC_FUNC_CD
<> 3 /*Cancelled*/ AND (B.PSTNG_LN_GRP_CTR IS NULL OR
B.PSTNG_LN_GRP_CTR = 0) AND B.BFY >= 2004 AND NOT EXISTS (
    -- This will find out if we are inside of a Transaction that is cancelled, which we should
    --- ignore cancelled transactions.
    SELECT D.DOC_CD FROM DOC_ACTG D WHERE B.DOC_CD = D.DOC_CD AND
    B.DOC_DEPT_CD = D.DOC_DEPT_CD AND B.DOC_ID = D.DOC_ID AND
    D.DOC_PHASE_CD = 1 /*Draft*/ AND D.DOC_FUNC_CD = 3 /*Cancelled*/
) AND EXISTS (
    -- We only care about ALs that have Standard PLs, since Non-Standard will not be
    -- referenced
    SELECT E.DOC_CD FROM PSTNG_LN_CAT E WHERE B.DOC_CD = E.DOC_CD
    AND B.DOC_DEPT_CD = E.DOC_DEPT_CD AND B.DOC_ID = E.DOC_ID AND
    B.DOC_VERS_NO = E.DOC_VERS_NO AND B.DOC_VEND_LN_NO =
    E.DOC_VEND_LN_NO AND B.DOC_COMM_LN_NO = E.DOC_COMM_LN_NO
    AND B.DOC_ACTG_LN_NO = E.DOC_ACTG_LN_NO AND E.LN_FUNC_CD = 1
    /*Standard*/
) AND B.DOC_COMP_MOD_FL=1
ORDER BY B.DOC_CD,
B.DOC_DEPT_CD,B.DOC_ID,B.DOC_VEND_LN_NO,B.DOC_COMM_LN_NO,B.DOC_
ACTG_LN_NO, DOC_VERS_NO
```

Job Return Code

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

Return Code	Condition
Successful (1)	Corrupted Accounting Line Catalog and Posting Line Catalog records were successfully fixed.
Failed (12)	This return code will be issued under several conditions: 1. Parameter validation failed. 2. Run time exceptions for unexpected situations. 3. There is no Standard Posting Line or Accounting Line to alter the selected Errant Accounting Record. 4. More than 1 Accounting Line map to the current Errant Record.
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 Sequence

The records will be sorted by:

- Transaction Code
- Transaction Department Code
- Transaction ID
- Transaction Vendor Line Number
- Transaction Commodity Line Number
- Transaction Accounting Line Number
- Transaction Version Number (only for select DRAFT / PENDING version problematic Accounting Lines)

Selection Criteria

The program selects records from the Accounting Catalog (DOC_ACTG) and Posting Line Catalog (PSTNG_LN_CAT) table that have Transaction Type and Transaction Code matching the ones given in the job parameters. It selects those records whose Budget Fiscal Year is greater than the one specified in the job parameter.

Steps After Completing the Job

The indices added for this job should be dropped from the database.

1. Accounting Catalog indices[DOC_ACTG]:

   Example (Oracle syntax):
   DROP INDEX N1_DOCACTG_PSTNGLNGRP
   DROP INDEX N2_DOCACTG_PSTNGLNGRP

2. Posting Line Catalog Indices [PSTNG_LN_CAT]:

   Example (Oracle syntax):
   DROP INDEX N1_PSTNGLNCAT_PSTNGLNGRP
   DROP INDEX N2_PSTNGLNCAT_PSTNGLNGRP
   DROP INDEX N3_PSTNGLNCAT_PSTNGLNGRP

Problem Resolution

The following table shows the possible return codes and recommendations.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Failed (12)	Required Parameters are not entered. **Sample Message:** Transaction Type is required.	Enter valid parameters and run the job again.	N/A
	Run time exceptions for unexpected exceptions. **Sample Message:** Failed to commit Draft Accounting Lines.	The transaction reported in the job log could not be committed. Please investigate the transaction.	The record reported in the report could not be fixed.
	There are no Posting Lines or Accounting Lines updated for the selected Errant record **Sample Message:** We did not find any Standard Posting Lines to alter for this current Errant Accounting Line	The record reported in job log did not map any Accounting or Posting Lines. So updates could not occur. Please investigate the record.	The record reported in the report could not be fixed.
	More than 1 Accounting Lines map to current Errant record. **Sample Message:** We experience a problem where 2 Accounting Lines are getting updated mapping to current AL. However we expect only 1 update.	The record reported in the job log mapped more than 1 Accounting Line which is incorrect. Please investigate the record.	The record reported in the report could not be fixed.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	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. Schedule a new job.	N/A

2.1.20 R311 Encryption Utility

Chain or Job Name	R311 Encryption Utility
Recommended Frequency	This job should be run only once when your site upgrades to 3.11 from a prior release.
Single Instance Required	The R311 Encryption Utility does not support multiple instance of the job running in parallel.
Can be restarted?	The R311 Encryption Utility cannot be restarted. This job can be re-run again with the same set of parameters.
Reports generated	No reports are generated for this process.

Overview

This batch job should be run only if your site intends to be FIPS 140-2 compatible. If FIPS 140-2 compatibility is not needed then this batch job should be ignored.

Federal Information Processing Standards (FIPS) are standards that are developed by two government bodies. One is the National Institute of Standards and Technology in the United States. The other is the Communications Security Establishment in Canada. FIPS are standards that are either recommended or mandated for use in federal (either United States or Canadian) government-operated IT systems.

FIPS 140-2 is a statement of the Security Requirements for Cryptographic Modules. It specifies which encryption algorithms and which hashing algorithms can be used and how encryption keys are to be generated and managed. Some hardware, software, and processes can be FIPS 140-2 validated by an approved validation lab. Some of them can also be described as FIPS 140-2-compliant as the term is defined in this article.

Advantage uses symmetric encryption to encrypt sensitive columns like Bank Account Number, Credit Card Number, and so on. The data symmetric key used for encryption is stored in the table as a separate column alongside with the encrypted column. This data symmetric key in prior releases was encrypted using PBE (Password Based Encryption). PBE used in the application is currently not supported by the FIPS 140-2 standard. This batch job converts the existing encrypted data for all the sensitive columns like Bank Account Number, Credit Card Number, and so on to an encryption standard which is FIPS 140-2 compatible. The process uses a technique where a Master symmetric key is used to encrypt the data key using symmetric encryption. This Master key is usually generated by the site prior to running this batch job.

This process starts with transaction validation. This batch job utilizes the 3 parameters as listed below.

- COMMIT_BLOCK_SIZE
- MODE (Report -1 , Update -2)
- PARM_FILE(Parms\DataObjectNames.txt)

If validation is successful, then it searches for the file DataObjectNames.txt inside the Parms directory. This file has a list of data objects which has encrypted columns. It goes through all the data objects listed in the file and for each data object it performs a select query to get all records from the Data Object. It loops through all the encrypted columns in the list and for each encrypted column it decrypts the data using the old symmetric encryption standard and it re-

encrypts using the new encryption standard. It saves the re-encrypted value in the encrypted column and stores the encrypted data key inside the data key column.

This job can be run in Report Mode or Update Mode. It's advisable that you run the batch job first in Report Mode. In Report Mode, when the batch job is run, it indicates all the errors found. By doing this, you will know beforehand if you need to do any corrections prior to running the job in Update Mode. After resolving the errors in Report Mode, the job should be run in Update Mode. In Update Mode, the job does the same thing as in Report Mode, but is also commits to the database.

It is also advisable to keep a backup of the tables listed below in the Major Input section prior running this batch job.

Process Steps	Messages
1. Validating Parameters	• Validating Parameters • Parameters are valid or invalid depending on the validation. If the parameter is invalid, the invalid value will be displayed in the log: o "Invalid Dataobject name: <data object name>" o Validation Complete is issued upon completion of the validation process.
2. Selecting Records	• Conversion for <data object> started.
3. Updating Records	• "<Progress Block Size Multiple> rows processed" • When the updating of records is complete, the message, "Conversion for <data object> ended" will be issued.

The R311 Encryption Utility cannot be restarted. It can be rerun and will again select all of the records in the data object name parameter provided; however, it will ignore rows that have already been converted to the new encryption scheme from any previous runs.

Major Input

Data objects that store one or more encrypted field(s). This value is taken as a list of all the Data objects in the application in a parameter file (DataObjectNames.txt). This is the list of Data Objects which will be impacted with this conversion process. It is recommended that you keep a backup of the Data Objects in the list. This data object list can be edited and it can also be run for a subset of the list below or all the data objects.

- R_VEND_CUST
- ABS_DOC_VEND
- CR_DOC_VEND
- AD_DOC_HDR

- AD_DOC_VEND
- DRM_DOC_VEND
- PR_DOC_VEND
- R_AD
- R_AP_CW_PYMT
- R_AP_DISB_RQST
- R_AP_DISB_RQST_LN
- R_AP_PRCU_CARD_ADM
- R_AP_UPROC_PRCU
- R_PNT_VEND_CUST
- R_VSS_USER
- TRVL_DOC_VEND
- VCC_DOC_AD
- VCC_DOC_VCUST
- VCM_DOC_AD
- VCM_DOC_HQ
- VCM_DOC_VCUST

Mode (1- Report, 2- Update) - This job can be run in Report Mode or Update Mode. It is advisable that you run the batch job in Report Mode first and then in Update mode. In Report Mode, the batch job is run and indicates all the necessary errors. You will know beforehand if you need to do any corrections prior to running the job in Update Mode. No database commit is issued when the job is run in Report Mode.

After resolving the errors found in Report Mode, the job should be run in Update Mode for the actual encryption to happen. When the job is run in Update Mode, it does the same thing as Report Mode but it commits to the database.

Commit Block Size – This represents the amount of rows you would like the job to process before committing to the database.

Parameter	Description	Default Value
COMMIT_BLOCK_SIZE	Commit Block size	1000
MODE	Mode (1 - Report , 2 - Update)	1
PARM_FILE	Parameters file for Data Objects	$$AMSPARM$$\DataObjectsName.txt

Major Output

The encrypted columns are updated with the FIPS 140-2 compatible Symmetric Encryption algorithm. The data key column (PARITY_STR) is updated with the encrypted data key.

Return Code	Condition
Successful (1)	All of the selected records are updated successfully.
Warning	This means that there were some error which occurred during encryption and decryption. There may be some manual intervention required to convert the data. It is recommended that the job is first run in Report Mode so that you can review the job log for errors. There could be errors as shown below which means that it was not able to decrypt the particular Vendor from the VCUST page. This could be an issue with the data or setup. Error found on table R_VEND_CUST for record VEND_CUST_CD = 'VS0000000184' If you see this error repeated for all the records then it means it is a setup issue and the system administrator needs to be called to resolve the setup issue. Logging may need to be turned on to see details about the error in the job logs. If you see this error only for a few records then it could be bad data which would need to be manually corrected by opening the page for the respective table and selecting the right record based on the primary key.
Failed (12)	The job will fail under the following conditions: • Parameter is invalid (Data Object is not found or not provided, or Commit Block Size was not positive) • Run time exceptions for unexpected situations There could be a setup issue if there are runtime exceptions and the system administrator needs to be called to resolve the setup issue. Logging may need to be turned on for security to see details about the error in the job logs.
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

Selects all of the rows of the data object specified where PARITY_STR is not null.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the parameters are validated successfully.	N/A	N/A
Warning(2)	The job went through successfully but need to look at the logs to address the records which were not encrypted successfully.	This means that there were some errors which occurred during encryption and decryption. Manual intervention may be required to convert the data. It is recommended that the job is first run in Report Mode so that you can review the job log for errors. There could be errors as shown below in the example: Error found on table R_VEND_CUST for record VEND_CUST_CD = 'VS0000000184' This means that it was not able to decrypt the particular Vendor from the VCUST page. This could be an issue with the data or setup. If you see this error for all the records then it means it is a setup issue and the system administrator needs to be called to resolve the setup issue. Log4j may need to be turned on to see details about the error in the job logs. If you see this error only for a few records then it could be bad data which would need to be manually corrected opening the page for the respective table and selecting the right record based on the primary key.	
Failed (12)	Job failed due to fatal conditions.	In this step, the job can fail under the following conditions: • Encounters any runtime exceptions	

Possible Return Codes	Condition	Recommendation	Other Instructions
		due to bad setup. • Data object parameter was not provided. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and reschedule the job. Log4j may need to be turned on for Security to see detailed logging for the issue.	
	Failed because the Parameters file for Data Objects was not provided. **Sample Message**: Parameter List for the data object is required.	**Recommendation:** Schedule a new job, providing the valid Data Object Name names in the DataObjectNames.txt file.	
	Failed because the data object parameter was not valid. **Sample Message**: Invalid data object name: R_SOME_BAD_TABLE	**Recommendation:** Schedule a new job, providing the valid Data Object Name names in the DataObjectNames.txt	
	Failed because the Commit Block Size was not positive. **Sample Message**: Commit Block Size must be a positive integer.	**Recommendation**: Schedule a new job, providing a positive Commit Block Size.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can be re-scheduled as a new job or restarted.	

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. The job can either be restarted or schedule a new job.	

2.1.21 Receivable History and Reference Data Fix

Job Name	Receivable History and Reference Data Fix
Recommended Frequency	Once. Refer to the "Steps Before Running Job" and "Steps After Running Job" sections for important instructions that need to be followed before and after the execution of the batch process.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes

Steps Before Running Job

No other transactions should take place while the Receivable History and Reference Data Fix is in progress.

Unarchive any archived Intercept Transactions.

Overview

When an Intercept was performed during the Automated Disbursement process, if a Fee was captured along with Intercepted Money, the Fee amount would show up as part of the Liquidated Amount inside of the Receivable History and Reference table incorrectly, even though the Fee amount does not liquidate the referenced RE. The Fee Amount should not count towards the amount that has been liquidated.

The Receivable History and Reference data fix will fix the Liquidated Amount field for Intercept transactions to back out any Intercepted Fee Amounts which may have been included in the Liquidated Amount field incorrectly.

Process Steps	Messages
4. Parameter Validation	Validating Batch Parameters listed in the above Batch Parameter section The basic validations are: • RPT_MODE: ensures that it is 1 or 2. If not, an error is logged. • RPT_PATH: If no value is provided, it takes the root directory as the output directory • RPT_FL_NM_PFX: Ensures that it is not empty. If empty, it takes the default "PymtDetInq" • SELECT_BLOCK_SZ: If no value is provided, it defaults to 500. If value provided is not a number, an error is logged • COMMIT_BLOCK_SZ: If no value is provided, it defaults to 100. If value provided is not a number, an error is logged

	• INTRCPT_DOC_CDS: If no value is provided, it defaults to value of “IT” Batch Parameter validation completed.
5. Selection of Records	This starts the logical looping through each RDET detail record with potential data problem.
6. Calculation and Updates	Determine the total posting line liquidation amount resulting from this Intercept Transaction version referencing the RE transaction for this RDET record. If the calculated amount differs from the RDET detail Liquidated amount, report them to the spreadsheet report. For report and update mode only, update the liquidated value of RDET detail record with the calculated amount.

Restartability information

The job cannot be restarted but can be rescheduled as only records with data problems will be eligible for the data fix. The reason of job failure needs to be investigated by looking at the job error log and error log files.

Major Input

- Receivable History and Reference Detail table (R_PYMT_DET_INQ_DET)
- Posting Line Catalog table (PSTNG_LN_CAT)

Batch Parameters

Parameter Name	Description (Caption)	Default Value
INTRCPT_DOC_CDS	Intercept Transaction Codes. Required. This should be the set of Intercept Transaction Codes which may have Fees that need to be backed out of the Receivable History and Reference Detail table.	IT
RPT_PATH	Report Path. Required. This is the location that the Report will be located when generated.	$AMSEXPORT$
RPT_FL_NM_PFX	Report File Name. Required. This is the prefix for the name of the report that will be generated for the current run. It should not contain an extension; an extension of .xlsx will be applied when the Excel Spreadsheet is created. If a file with that file name already exists, it will be deleted. The file name will have the format: RPT_FL_NM_PFX+Job Id_DateTimeStamp.xlsx.	PymtDetInq

Parameter Name	Description (Caption)	Default Value
RPT_MODE	Report Mode. Required. This mode will dictate whether it was Report Only Mode (value of 1) or Report and Update Mode (value of 2). Report Only mode will only generate a report with the changes that would be performed if updating of the records is performed.	1 (Report Only)
SELECT_BLOCK_SZ	Select block Size. If not entered then defaulted to 500. It is the number of records that can be fetched as a single block. Can be used for Performance tuning.	500
COMMIT_BLOCK_SZ	Commit block Size. If not entered then defaulted to 100. It is the number of records that can be changed before a commit is called. This value is a suggestion and not a requirement, as we may call commit sooner than this count. Can be used for Performance tuning.	100

Major Output

- Receivable History and Reference Detail table (R_PYMT_DET_INQ_DET)
- An Excel workbook named RPT_FL_NM_PFX_jobID_DateTimeStamp.xlsx will be written into the folder location indicated by the RPT_PATH parameter

Job Return Code

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

Return Code	Condition
Successful (1)	Corrupted Receivable History and Reference Detail records were successfully reported / fixed.
Warning (4)	If no incorrect record is found in the Receivable History and Reference Detail table then the job ends in a job return code of Warning. Necessary details are logged in the job log.
Failed (12)	This return code will be issued under several conditions: 1. Parameter validation failed. 2. 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 Sequence

The records will be sorted by:

- Transaction Code
- Transaction Department Code
- Transaction ID
- Accepted Date
- Reference Transaction Code
- Reference Transaction Department
- Reference Transaction ID
- Reference Transaction Version

Selection Criteria

The program selects records from the Receivable History and Reference Detail table (R_PYMT_DET_INQ_DET) that have Reference transaction Code matching the Intercept Transaction Codes supplied in the job parameter.

Problem Resolution

The following table shows the possible return codes and recommendations.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	No account was found with discrepancy Sample Message: No result was returned from the current result set query for transaction Code	If no discrepancies were found, no further action is required.	N/A
Failed (12)	Required Parameters are not entered. **Sample Message:** Run Mode Value is required.	Enter valid parameters and run the job again.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	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. Schedule a new job.	N/A

Steps After Running Job

Once the job has been run **in report and update mode**, the client will not run it again.

2.1.22 Remittance Advice Email Conversion

Job Name	Remittance Advice Email Conversion
Recommended Frequency	This job is required to run only once to complete the data conversion expected under the enhancement Remittance Advice Email Attachment.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

Enhancement Remittance Advice (RA) Email Attachment provides the ability to email RA to vendors by adding few new attributes and related functionality on the EFT transaction, Vendor Customer Page and transactions, Address Page and Disbursement Format Page.

The RA Format and RA Transmission Mode fields are being added to the Accounting component of the EFT Transaction. Data conversion is required on the Accounting component of the EFT Transaction.

An EFT transaction with the transaction phase of final and transaction function not equal to Cancellation is considered during the conversion.

RA Transmission Mode is set to Postal for all of the selected records.

The RA Format will be derived from the Address table or Location level of the Vendor based on the setting of Allow Prenote/EFT on the Application Parameter (APPCTRL) table.

- If Allow Prenote/EFT is True on Application Parameter (APPCTRL):

  If the EFT is for a Payee the RA Format will be inferred from the Address table for the Address ID in the transaction. If RA Format is not defined on the address ID then it will be inferred from the location level of the Payee vendor. Similarly if the EFT does not have a Payee then the RA Format will be inferred from the Address table for the Address ID of the Vendor If RA Format is not defined on the address ID then it will be inferred from the location level of the Vendor.

- If Allow Prenote/EFT is False on Application Parameter (APPCTRL):

  If the EFT is for a Payee the RA Format will be inferred from the location level of the Payee vendor. Similarly if the EFT does not have a Payee then the RA Format will be inferred from the location level of the Vendor.

The following table shows the various steps that the RA Email Conversion Job goes through and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	The conversion job does not have any Job Parameters.
2. Selection and Evaluation of Records	• Selecting EFT Accounting Line with DOC_CD: xxxx DOC_DEPT_CD : xxxx DOC_ID : xxxx DOC_VERS_NO : xxx DOC_ACTG_LN_NO : xxx

Process Steps	Messages
	• Conversion of EFT Accounting Line completed. • Total EFT Accounting Line processed are xxxx

Restartability Information

This job cannot be restarted. If the job failed due to any reason, a new job should be scheduled after correcting the errors that caused the job to fail.

Major Input

Tables

The following is list of tables that are used as input:

- AD Transaction Header (AD_DOC_HDR)
- AD Transaction Vendor (AD_DOC_VEND)
- AD Transaction Accounting (AD_DOC_ACTG)
- Vendor Customer (R_VEND_CUST)
- Address (R_AD)
- Disbursement Format(R_AP_DISB_FRMT)

Batch Parameters

- None

Major Output

- Table Updates
    - Update the RA Transmission Mode and RA Transmission Format on AD_DOC_ACTG

Job Return Code

The following table shows the potential job return codes for the Disbursement Printing job.

Return Code	Condition
Successful (1)	All of the selected EFT records are processed successfully.
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

- Select Records from AD_DOC_ACTG

  Where DOC_PHASE_CD = 3 (FINAL)

  and DOC_FUNC_CD <> 3 (CANCELLED)

  and REMT_ADV_FRMT IS NULL

  and DOC_CD = (SELECT DOC_CD from R_GEN_DOC_CTRL Where DOC_SUBTYP = 'EFT')

2.1.23 Sync ILOC Quantities with INVN

Job Name	Sync ILOC Quantities with INVN
Recommended Frequency	One Time
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	Yes. Sync ILOC Quantities with INVN Exception Report

Overview

This Conversion data fix job should be run while upgrading to 4.x releases to set the various quantity fields on the Inventory by Location (ILOC) table to be in sync with the quantities on the Inventory Maintenance (INVN) table.

The following quantity fields on the ILOC table will be corrected by this data fix job:

- On Hand Quantity
- Reserved Quantity
- Released Quantity
- In Transfer Quantity
- Available Quantity

This data fix job fetches the quantity fields from the Inventory Maintenance (R_INVN) and Inventory by Location (R_INVN_BY_LOC) tables and compare the values of R_INVN with sum of quantities on the R_INVN_BY_LOC table for each Warehouse (WHSE_CD) and Stock Item (STK_ITM) combination. If there is a difference between quantities, then this job will update the R_INVN_BY_LOC table to match the quantities on the R_INVN table.

Assumption

- Quantities on the Inventory (INVN) table are considered correct and it will be used to update the quantities on the Inventory by Location (ILOC) table.
- Run this job before using the new format Pick and Issue (PI) transaction, which has the new Pick Bin lines.
- It is recommended to take a backup of the Inventory by Location (ILOC) table.

Major Input

The following is list of tables that are used as input:

- R_INVN (Inventory)
- R_INVN_BY_LOC (Inventory by Location)

Major Output

Table Updates

- Update the On Hand, Released, Reserved, In transfer and Available quantity on R_INVN_BY_LOC.

Exception report

- If any record with the following fields as a negative value on the INVN table, it will be added to the Exception report:
    - On Hand
    - Reserved
    - Released
    - In Transfer

    Also, if the On Hand – Reserved – Released is less than zero on INVN, the record will be added to the Exception report.

Job Return Code

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

Return Code	Condition
Successful (1)	If the Incorrect quantities (On hand, Released, Reserved, In transfer and available) on the Inventory by Location table are updated .
Warning (4)	If no incorrect records are found for correction.
Failed (12)	If any exception occurred, while updating the records.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Selection Criteria

The job selects records from ILOC where (all statements are OR):

- Sum of On Hand Quantity (ON_HAND_QTY) on ILOC is not equal to On Hand Quantity on INVN.
- Sum of Reserved Quantity (RS_QTY) on ILOC is not equal to Reserved Quantity on INVN.
- Sum of Released Quantity (REL_QTY) on ILOC is not equal to Released Quantity on INVN.
- Sum of In Transfer Quantity (IN_XFER_QTY) on ILOC is not equal to In Transfer Quantity on INVN.
- Available Quantity (AVAIL_QTY) on ILOC is not equal to On Hand – Reserved – Released on ILOC.

Processing Logic

Notes: When updating ILOC records, the sequence should be as follows:

1. Update records with the Default flag checked.

2. Update records with the highest priority (1 to 99, 1 being highest) and make sure to skip the default record as its already updated in #1 above.

3. Compare On Hand Quantity of the INVN record with the Sum of On Hand Quantity on the ILOC record for the WHSE/STK_ITM combination. If it is not equal, then updates are needed to the ILOC table.

    - For any record on ILOC where the On Hand Quantity field is a negative value, the system will set it to Zero before performing the calculations.
    - Positive value (INVN is greater than ILOC): Add quantity to Default location, if no default then the highest priority record.
    - Negative value (ILOC is greater than INVN): Subtract till it goes to zero, if balance remains, move to the next ILOC record.

4. Compare Released Quantity of the INVN record with the sum of Released Quantity on ILOC records for the WHSE/STK_ITM combination. If it is not equal, then updates are needed.

    - For any record on ILOC where the Released Quantity field is a negative value, the system will set it to Zero before performing the calculations.
    - When updating REL_QTY on ILOC make sure REL_QTY is not updated to a value greater than On-Hand Quantity.
    - Positive value (INVN is greater than ILOC): Add and make sure it is not going beyond On Hand. If balance remains, move to the next ILOC record.
    - Negative value (ILOC is greater than INVN): Subtract till it goes to zero, if balance remains, then move to the next ILOC record.

5. Compare Reserved Quantity of the INVN record with the sum of Reserved Quantity on the ILOC records for the WHSE/STK_ITM combination. If it is not equal, then updates are needed.

6. When updating RS_QTY on ILOC make sure that Available Quantity is not going negative, that is, RS_QTY + REL_QTY =< ON_HAND_QTY.

    - For any record on ILOC where the Reserved Quantity field is a negative value, the system will set it to Zero before performing the calculations.
    - Positive value (INVN is greater than ILOC): Add and make sure Reserved + Released does not exceed On Hand.
    - Negative value (ILOC is greater than INVN): Subtract and make sure Reserved + Released does not exceed On Hand. If balance remains, then move to the next ILOC record.

7. Compare In Transfer Quantity of the INVN record with the sum of In Transfer Quantity on ILOC records for the WHSE/STK_ITM combination. If it is not equal, then updates are needed.

    - For any record on ILOC, if In Transfer Quantity field is a negative value, the system will set it to Zero before performing the calculations.
    - Positive value (INVN is greater than ILOC): Add quantity to the Default location, if no default then the highest priority record.
    - Negative value (ILOC is greater than INVN): Subtract till it goes to zero, if balance remains, then move to the next ILOC record.

8. Select ILOC records where AVAIL_QTY is not equal to ON_HAND_QTY – RS_QTY – REL_QTY. If it is incorrect, set it to the correct value by using the formula: AVAIL_QTY = ON_HAND_QTY – RS_QTY – REL_QTY.

Problem Resolution

This job does not support restartability. If the job fails for any reason, a new job must be scheduled.

2.1.24 TIN Conversion Process

Job Name	TIN Conversion Process
Recommended Frequency	One Time. This conversion process should be executed while upgrading to release 3.8 from a prior Financial release.
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	Yes

Overview

This conversion job should be run while upgrading to Financial 3.8 to initialize new Taxpayer Identification Number (TIN) and Taxpayer Identification Number Type (TIN Type) fields. This job updates Headquarters records with the TIN and TIN Type from the Location section of its Vendor/Customer (VCUST) record. This job generates two reports. The Headquarters Account Code Mismatch Report lists all Vendor/Customer records that share the same TIN and TIN Type but do not share the same Headquarters Account. This is an optional report and is generated only if the Generate Report Parameter for this job is supplied as *Y*. It is recommended to run this report if the Require Unique TIN/TIN Type across HQ account parameter is true on the Application Parameters (APPCTRL) table. The other report is the Exception Report which lists those Vendor/Customer Locations that do not share the same TIN and TIN Type as the other Vendor/Customer Locations with the same Headquarters Account Code. This report is useful for the user that wants to set the Require Consistent TIN/TIN Type between HQ and Locations parameter to true on the Application Parameters (APPCTRL) table.

Restartability Information

This job does not support restartability. If the job fails for any reason, a new job must be scheduled.

Process Steps	Messages
1. Parameter validation	Validating Batch Parameters • If the parameter is invalid, the log message will show the error statement. Batch Parameter validation completed
2. Generate Headquarters Account Code Mismatch Report	Before and after the report generation log messages are written.
3. Update the Headquarters table and generate the Exception Report.	Progression messages when the Update and Exception Report generation is completed: • Started with update logic. • Rendering Reports started. • Run Ended

Major Input

- Headquarters Account Table (R_PNT_VEND_CUST) – Vendor/Customer Location records for every Headquarters Account are searched to identify the TIN and TIN Type values to be updated on the Headquarters Account records.
- Vendor/Customer Location Table (R_VEND_CUST)

Parameters

Batch Parameters

Description (Caption)	Parameter Name	Default Value
Generate Report (Y/N) Optional Field. If *Y*, the Headquarters Account Code Mismatch Report is generated. If *N*, this report is not generated. In both cases the Exception Report is generated.	GEN_REPRT	(blank)

Major Output

- Headquarters Account Code Mismatch Report showing Vendor/Customer records sharing the same TIN and TIN Type but not the same Headquarters Account Code.
- Updates the Headquarters Account records with TIN and TIN Type from their Vendor/Customer Locations.
- Generates an Exception Report.

Job Return Code

The following table shows the potential job return codes for the TIN Conversion job.

Return Code	Condition
Successful (1)	All of the records are processed successfully and reports are generated without errors.
Warning (4)	N/A
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: • Entered 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 Sequence

Vendor/Customer Record and Parent (Headquarters) Account Code

Selection Criteria

Headquarters Account Code Mismatch Report: All Vendor/Customer Location records that share the same TIN and TIN Type but not the same Headquarters Account Code.

Update Headquarters Account Table:

- For every Headquarters Account Code, search all of the related Vendor/Customer Location records.
    - If all related Vendor/Customer Location share the same TIN and TIN Type, then copy TIN and TIN Type to the Headquarters Account.
    - If the related Vendor/Customer Location records share different TIN and TIN Type, then write these records to the Exception Report.

Problem Resolution

If the process was discontinued for any reason the job cannot be restarted. A new job must be scheduled.

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:** Parameter is not valid	Schedule a new job.	
	Failed because of runtime exceptions for an unexpected situation.	The failure reason needs to be investigated before scheduling a new job	
Terminated (16)	Job is terminated manually by the user.	The 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.	The reason for the System Failure needs to be investigated before scheduling a new job.	

Step 2: Creating reports and updating Headquarters records - This step will be performed only when the parameters are valid.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the reports are generated successfully and updates are done.	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)	Failed because of runtime exceptions for an unexpected situation.	The failure reason needs to be investigated before scheduling a new job.	N/A
Terminated	Job is terminated manually by	The reason for the termination needs to be	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
(16)	the user.	investigated before scheduling 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 before scheduling a new job.	N/A

Troubleshooting

If the job fails due to some reason or due to power failure there is no other option but to restart the job.

2.1.25 Update Receivable Closed Date

Chain or Job Name	Update Receivable Closed Date
Recommended Frequency	On Demand. This job can also be scheduled in offline mod.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes

Overview

This Update Receivable Closed Date job updates the Transaction Closed Date (DOC_CLSD_DT) on RE_DOC_HDR and associated DOC_HDR records respectively, if the Transaction Closed Date on header is not populated but all of its accounting lines are closed, that is, Closed Date (AL_CLSD_DT) is populated on all of its accounting lines. The job takes the latest among the Closed Date of all its accounting line and uses that date to set the Transaction Closed Date on the RE transaction header, which will internally update the respective DOC_HDR record.

The following table shows the various steps that the Update Receivable Closed Date job goes through and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	The following messages are displayed in both online and offline mode. The basic validations are: • RUN_MODE: ensures that it is 1 or 2. If not, an error is logged. • Parameters are valid or invalid depending on the validation. If the parameter is invalid, the job will fail with proper message displayed in job logs.
2. Selection and Evaluation of Records	This starts the logical looping through each RE_DOC_ACTG record with all the data available in the database. The following messages are displayed only in offline mode. • Start processing the records. • If the selection returns zero records, then this step will be completed and the following message is issued: "No eligible records are found". • Number of Records Selected and Processed: <Count>
3. Calculation and Updates	The job takes the latest among the closed date of all its accounting line and uses that date to set the Transaction closed date on RE transaction header, which will internally update the respective DOC_HDR record.

Restartability information

The job cannot be restarted but can be rescheduled as only records with data problems will be eligible for the data fix. The reason of job failure needs to be investigated by looking at the job error log and error log files.

Major Input

- Receivable Transaction Accounting table (RE_DOC_ACTG)
- Receivable Transaction Header table (RE_DOC_HDR)

Batch Parameters

Offline printing process

Parameter	Description	Default Value
RUN_MODE	Run Mode. Required. This parameter indicates the run mode. Select Only Mode (value of 1) or Update Only Mode (value of 2). The Select Only mode will only generate a report with the selection performed and Update Only mode will update the records selected and generate a report with the updated value.	null

Major Output

- Receivable Transaction Header table (RE_DOC_HDR)
- Transaction Header table (DOC_HDR)

Job Return Code

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

Return Code	Condition
Successful (1)	Number of Records Selected and Processed: <Count>
Warning (4)	If no incorrect record is found, then the job ends with a job return code of Warning. Necessary details are logged in the job log.
Failed (12)	This return code will be issued under several conditions: 1. Parameter validation failed. 2. 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 Sequence

For report, the accounting lines for a particular RE transaction is sorted by:

- Transaction accounting line number.

Selection Criteria

The program selects records from the Receivable Transaction Header table (RE_DOC_HDR).

Problem Resolution

If the job fails for any reason, then a new job should only be scheduled. There is no need to back out any updates if this job fails in any of the steps since the user hold flag and hold payment reason on the corresponding table gets updated if all records are processed successfully.

Selection Mode:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	No eligible records found	N/A	N/A
Failed (12)	Required parameters are not specified. **Sample Message:** Run mode should be Report(1) or Report and Update(2) and should not be empty.	Modify the run mode as per instruction in the job logs.	.
	The job may fail in case of run time exception.	The reason needs to be investigated for run time exception. 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)	The job is terminated manually by the user.	The reason for the termination needs to be investigated and a new job should be scheduled.	
System Failure (20)	The job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated and a new job should be scheduled.	

Update Mode:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	No eligible records found	N/A	N/A
Failed (12)	Required parameter not specified **Sample Message:** Run mode should be select(1) or update(2) and should not be empty.	Modify the run mode as per instruction in the job logs.	.
	The job may fail in case of run time exception.	The reason needs to be investigated for run time exception. 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)	The job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling a new job.	.
System Failure (20)	The job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before scheduling a new job.	

2.1.26 Update Total Attachment Count

Job Name	Update Total Attachment Count
Recommended Frequency	On Demand. This conversion process should be executed to update the attachment count in the components given in the input file.
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	No

Overview

This batch job has been developed to prevent the out of synch issues for the attachment count of the components of the different transactions.

The attachment count is taken from the IN_OBJ_ATT_CTLG table based on the UNID attribute and if the UNID attribute's value is matched with that of the component's UNID then the attachment count is updated.

The input files (FileForPGTot.txt and FileForSGTot.txt) provide a transaction component listing as input to this batch job. When executed, the job reads each component present in the input text files (FileForPGTot.txt and FileForSGTot.txt) and passes the component to the update query present in the batch job file. This batch job updates the selected component's attachment count by taking the attachment count from the IN_OBJ_ATT_CTLG table.

This batch job updates the primary group total attribute and the secondary group total attribute of each component present in the input text files (FileForPGTot.txt and FileForSGTot.txt).

Restartability Information

This job does not support Restartability. If the job fails for any reason, a new job must be scheduled.

Process Steps	Messages
1. Parameter validation	If the parameter is invalid, the log message will show the error statement.
2. Update the records	Updates all the components given in the input text files with the correct attachment count from the IN_OBJ_ATT_CTLG table.
3. Job Complete	Job processing complete.

Major Input

- The input text files: FileForPGTot.txt and FileForSGTot.txt that has all the components whose attachment count has to be updated is placed in the Export/Import location.

Parameters

Parameter	Description	Default Value
Export Location where the text file or input text files (FileForPGTot.txt and FileForSGTot.txt) required for this batch job	Required. This parameter specifies the location of the input text file.	$$AMSEXPORT$$

Major Output

- Updates the component's primary group total attribute and the secondary group attribute based on the matching UNID attribute from the IN_OBJ_ATT_CTLG table.

Job Return Code

The following table shows the potential job return codes for the Update Total Attachment Count job.

Return Code	Condition
Successful (1)	Updates the component's primary group total and the secondary group total based on the matching UNID from the IN_OBJ_ATT_CTLG table.
Failed (12)	The job may fail under the following condition: • The batch parameter is not specified correctly.

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

If the job fails, submit a new job.

Troubleshooting

If the job fails due to some reason or due to power failure, there is no other option but to restart the job.

2.1.27 Vendor Verification Update

Job Name	Vendor Verification Update
Recommended Frequency	One Time. This conversion process should be executed while upgrading to post 3.8.0.1 releases.
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	No

Overview

This conversion job should be run while upgrading to releases post 3.8.0.1 to set the vendor verification details properly in R_VEND_CUST for location vendors. This batch job can also be run if there is a mismatch of verification details between any headquarters and its corresponding location vendors.

This batch job will fetch the following vendor verification fields from Headquarter Account (R_PNT_VEND_CUST): Verify My Locations by (VEND_VER_PSWD_TYP), Vendor Verification Based On (VEND_VER_HINT), Vendor Verification Password (VSS_PASS) and Confirm Verifications (VEND_PSWD_VERIFY). The batch job will then compare the values with the corresponding location vendor verification fields and update them if they do not match the headquarter account.

The batch job fetches all of the location vendor verification details, compares them with the headquarter account, and updates the location vendor verification as follows:

1. If the location vendor has a shadow key (PARITY_STR), then it will use the existing shadow key to encrypt the vendor verification details fetched from the headquarter account before the value is saved for the location vendor (R_VEND_CUST). The list of vendors that is updated through this step will be found in the VendorUpdatedWithExistingShdw.txt file and this can be found in the “XMLExportFileLocation” location mentioned in the ADV30Params.ini file.
2. If the location vendor does not have a shadow key (PARITY_STR) then a new key will be generated and used to encrypt the vendor verification details fetched from the headquarter account before the value is saved for the location vendor (R_VEND_CUST). The list of vendors that is updated through this step will be found in the VendorUpdatedWithNewShdw.txt file and this can be found in the “XMLExportFileLocation” location mentioned in the ADV30Params.ini file.

Restartability Information

This job does not support restartability. If the job fails for any reason, a new job must be scheduled.

Process Steps	Messages
1. Parameter validation	Validating Batch Parameters • If the parameter is invalid, the log message will show the error statement. Batch Parameter validation completed
2. Update Mismatch	Vendor verification details will be updated in location vendor to match the headquarter account, if the fields

Process Steps	Messages
Record	do not match.
3. Generate text files	VendorUpdatedWithExistingShdw.txt and VendorUpdatedWithNewShdw.txt will be created with the records that got updated through this process.

Major Input

- Headquarters Account Table (R_PNT_VEND_CUST): Vendor/Customer Location records for every Headquarters Account are searched to identify the mismatch in Verify My Locations by (VEND_VER_PSWD_TYP), Vendor Verification Based On (VEND_VER_HINT), Vendor Verification Password (VSS_PASS) and Confirm Verification (VEND_PSWD_VERIFY) fields on the Vendor/Customer Location (R_VEND_CUST) table.

Parameters

Batch Parameters

Description (Caption)	Parameter Name	Default Value
Commit block size	COMMIT_BLOCK	1
Select block size	SELECT_BLOCK	1

Major Output

- Updates the vendor verification details of Location Vendor/Customer records (R_VEND_CUST) with corresponding entries from Headquarter Account (R_PNT_VEND_CUST) if there is a mismatch.
- Generates the VendorUpdatedWithExistingShdw.txt and VendorUpdatedWithNewShdw.txt files.

Job Return Code

The following table shows the potential job return codes for the Vendor Verification Update job.

Return Code	Condition
Successful (1)	All of the records are processed successfully and reports are generated without errors.
Warning (4)	N/A
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: • Entered 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 Sequence

Headquarter Account Code

Selection Criteria

Location verification details of each Vendor/Customer record is compared with the corresponding headquarter account code and updated if there is a mismatch.

Problem Resolution

If the process was discontinued for any reason the job cannot be restarted. A new job must be scheduled.

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:** Parameter is not valid	Schedule a new job.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	Failed because of runtime exceptions for an unexpected situation.	The failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	The 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.	The reason for the System Failure needs to be investigated before scheduling a new job.	

Troubleshooting

If the job fails due to some reason or due to power failure there is no other option but to restart the job.

2.1.28 Verification Type 'No Password Required' Update

Job Name	Verification Type 'No Password Required' Update
Recommended Frequency	One Time. This conversion process should be executed to eliminate the No Password Required option from Vendor verification details.
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	No

Overview

Due to the potential security risks associated with the 'No Password Required' option, the No Password verification method and its associated functionalities is removed from VSS and the same will be removed from Financial if the Restrict VSS Access field is set to *No* on the R_VEND_CUST (VCUST) table in Financial.

This batch job will remove the 'No Password Required' option from R_VEND_CUST and R_PNT_VEND_CUST if the Restrict VSS Access field is set to *No* for the site using VSS. If the site is not using VSS, then the Restrict VSS Access field will be set to *Yes* for the vendor that has the 'No Password Required' option as the vendor verification method.

This batch job has two parts.

- Clients Who Do Not Use VSS:

  The data fix will set the Restrict VSS Access default to *True* (selected) on the DCTRL table for VCC and VCM transactions, as well as the newly created APPCTRL parameter, and also sets the Restrict VSS Access equal to *Yes* for all vendors that have *No Password Required* as the verification method.

- Clients Who Use VSS:

  1. When a parent vendor or headquarter account has the 'No Password Required' option, it will be updated to *Use My TIN Number*, with the Parent TIN being the verification. Any of the location vendors using the same Headquarter Account Code, with the Restrict VSS Access flag set as *False*, will have the vendor verification password set to the Parent TIN and the vendor verification method to *Use My TIN Number* and the vendor verification hint will be "Please verify that you are part of this organization by entering the TIN number of your Headquarters and hitting submit. If you are unsure of the TIN number, please contact the Account Administrator for your Headquarters." The list of parent account codes and the corresponding location vendors updated with this approach will be found in VendDetUpdWithTin.txt file.
  2. When a parent vendor or headquarter account has the 'No Password Required' option without Parent TIN and any of the location vendors using the same parent account code has the Restrict VSS Access flag as *False*, then the vendor verification password will be the custom password provided in the job parameter. If the job parameter value is NULL, then the job will use the parent account code as the custom password. In either case, the vendor verification method will be ''Create My Own'', the vendor verification hint will be the custom vendor verification hint provided, or if its empty, then "Please contact customer support for instructions to access your account" will be used. The list of parent account codes and their corresponding

location vendors updated with this approach will be found in VendDetUpdWithCstmDet.txt file.

3. If the 'No Password Required' option is required for the vendors then those vendor's parent account codes should be entered in a text file one by one and the Restrict VSS Access flag for those vendors using the parent account code provided in text file will be set to *True* without changing the vendor verification method.

Restartability Information

This job does not support restartability. If the job fails for any reason, a new job must be scheduled.

Process Steps	Messages
1. Parameter validation	If the parameter is invalid, the log message will show the error statement.
2. Update R_VEND_CUST and R_PNT_VEND_CUST Record	No Password required option will be removed from the R_VEND_CUST and R_PNT_VEND_CUST tables if the site is using VSS, if the site is not using VSS then Restrict VSS Access will be set to *Yes* on the R_VEND_CUST (VCUST) table.
3. Generate text files.	VendDetUpdWithTin.txt and VendDetUpdWithCstmDet.txt will be created with the records which got updated through this process.

Major Input

- Headquarters Account (R_PNT_VEND_CUST) table – The headquarter account which has vendor verification details set as "No Password Required" will be fetched initially and its corresponding location vendors that have "Restrict VSS Access" set to *False* will be retrieved.

Parameters

Batch Parameters

Description (Caption)	Parameter Name	Default Value
Export Location where the text file or input text file required for this batch job.	AMSEXPORT	$$AMSEXPORT$$
Name of the text file which has list of headquarter account codes for which "No Password required " option should not be removed.	FILE_NM	
Whether "No Password Required" option should be removed for all vendors (Yes/No).	REMV_FROM_ALL	
The custom vendor verification hint that the site wants to use. If left empty, then "Please contact customer support for instructions to access your account" will be used for vendors	VEND_VER_HINT	

without TIN.		
Whether the site is using VSS or not (Yes/No).	VSS_EXISTS	
Custom Password that the site wants to use. If left empty then corresponding vendor records parent account code will be used to set passwords in vendor verification.	VSS_PASS	

Major Output

- Updates the vendor verification details of the headquarter account code having vendor verification details set as *No Password Required* and the corresponding location vendor will be updated.
- Generates VendDetUpdWithTin.txt and VendDetUpdWithCstmDet.txt file.

Job Return Code

The following table shows the potential job return codes for the Vendor Verification Update job.

Return Code	Condition
Successful (1)	All of the records are processed successfully and reports are generated without errors.
Warning (4)	N/A
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: • Entered 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 Sequence

N/A

Selection Criteria

The Headquarter Account Code (R_PNT_VEND_CUST) having the vendor verification details set as *No Password Required* and the corresponding location vendors (R_VEND_CUST) will be selected.

Problem Resolution

If the process was discontinued for any reason the job cannot be restarted. A new job must be scheduled.

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:** Parameter is not valid.	Schedule a new job.	
	Failed because of runtime exceptions for an unexpected situation.	The failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	The 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.	The reason for the System Failure needs to be investigated before scheduling a new job.	

Troubleshooting

If the job fails due to some reason or due to power failure there is no other option but to restart the job.

2.2 Conversion Report Processes

The Conversion Processes report run sheet included in this section is:

- COGS Analysis Report

2.2.1 COGS Analysis Report

Job Name	COGS Analysis Report
Recommended Frequency	This job generates a report and can be run once when the client upgrades the Advantage product to the 3.7 Release.
Single Instance Required	Yes
Can be restarted?	Yes. The job can be restarted after error resolution. The job will start again from the beginning.
Reports generated	This job generates a report which helps in the analysis to determine COGS adjustment.

When to Run

This report job should be run if the inventory functionality is being used and event type changes were not made before production processing to change the S002 posting codes to D023 in posting pair C for event types ST10, ST11, ST20, ST21, ST30, and ST31. This report does not have to be run for upgrade/conversion to the 3.7 Release. It can be run at any point afterwards. Additionally, if updates to the Event Type table are not commonly taken when upgrading, the one mentioned in ADVFN00016948 should be taken as it changes these 4 records and others. Please see the release notes for more information on this defect.

After making adjustments, the report will still identify the records initially selected.

Overview

This job is part of the conversion process jobs and it generates a report which helps in the analysis to determine if any Cost of Goods Sold (COGS) adjustment needs to occur. A COGS adjustment would be needed if the delivered setup for event types ST10, ST11, ST20, ST21, ST30, and ST31 was using posting code S002 (Inventory Offset) instead of D023 (Cost of Goods Sold).

The report will total all postings found to S002 that should have been to D023 and will summarize by Warehouse, Budget Fiscal Year, Fiscal Year, and Accounting Period. Sites will then determine if they want to make any accounting adjustments using a Journal Voucher. There is no need to adjust the Inventory tables as updates there have been correct.

When adjustments are to be performed, users can retrieve from the Warehouse provided for each record, all COA needed from the WHSE table and create 1 line to back out S002. Then copy that line, change the posting code to D023, move amounts between the debit to credit fields, and enter the COGS object for that warehouse. At any point in time if the warehouse changed COA, a client will have to retrieve what the old COA were and use it for correcting accounts from those time periods.

The first step in the job is to read the “Client Name” parameter which is printed on the report. Providing a value for this parameter is optional and there is no validation for this parameter.

Records are then selected from the Accounting Journal Table (JRNL_ACTG) based on the following criteria: (Event Type is ST10, ST11, ST20, ST21, ST30, or ST31) and (Posting Pair C) and (Posting Code is S002).

The records are then summarized on warehouse (WHSE_CD), budget fiscal year (BFY), fiscal year (FY_DC), and accounting period (PER_DC) and written to the report.

The following table shows the various steps that the COGS Analysis Report Job goes through and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	• Currently Client Name is the only parameter for this job and since it is an optional parameter it is not being validated. For future use the standard parameter validation messages have been provided as follows: • Validating Batch Parameters • If the parameter/s are valid the message "Parameters are Valid" will be logged. • If the parameter/s is invalid the message "Parameter validation failed" will be logged.
2. Selection of Records and generating report	• Selecting eligible records from JRNL_ACTG. • Number of records (count) selected will be displayed.

Restartability Information

If the job fails in any of the above steps the job can be restarted after resolving the error. If the job is restarted, it will start from the beginning.

Major Input

Tables

- Accounting Journal (JRNL_ACTG)

Batch Parameters

Parameter	Description	Default Value
Client Name (CLIENT_NM)	Optional	No Default

Major Output

- COGS Analysis Report in HTML and PDF format.

Job Return Code

The following table shows the potential job return codes for the COGS Analysis Report job.

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Failed (12)	The job will fail under the following condition:

Return Code	Condition
	• 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 Sequence

N/A

Selection Criteria

The COGS Analysis Report job selects the records from the Accounting Journal Table (JRNL_ACTG) based on the following criteria:
(Event Type is ST10, ST11, ST20, ST21, ST30, or ST31) and (Posting Pair C) and (Posting Code is S002).

The records are then summarized on WHSE_CD, BFY, FY_DC, and PER_DC and written to the report.

Problem Resolution

The following tables show the possible return codes and recommendations for each processing step.

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	N/A	This step does not issue this return code.	This step does not issue this return code.
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	When the job is	The reason for the	

Possible Return Codes	Condition	Recommendation	Other Instructions
(20)	terminated because of database server or network issues.	System Failure needs to be investigated. The job can either be restarted or rescheduled.	

Step 2: Selection of records

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	No records were found.	No further action required.	N/A
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.	In this step the job can fail with fatal conditions only on encountering unknown exceptions. If that happens, investigate the exception reported by the process, resolve the error and restart 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.	