CGI Advantage® 4

Financial Utilities
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 Utilities Batch and Chain Processes

The processes that drive Utilities in CGI Advantage Financial are listed below in the order in which they are listed on the Job Manager page. Click on one of the process names to navigate to the corresponding run sheet.

- Automatic Transaction Correction
- Batch Output Purge
- Begin Day
- Batch Interface Pre-Processor
- BPRO To CACT Updates
- Budget Archiving
- COA Usage Verification
- Cost Accounting Archiving
- Data Conversion for Prior Bypass Functionalities
- Future Transaction Trigger
- Inbound Interface Reconciliation Report
- Interface Batch Demand Report
- Interface Transaction Exception Report
- Job Manager Output Archive
- Job Manager Output Restore
- Journal Ledger Archiving
- Matching Tables Archive
- Matching Tables Restore
- MRT Generation (Admin only)
- Multiple Transaction Submit
- Multi Process Approve
- Multi Process Submit
- Multi-threaded Table Loader
- Multi-threaded Transaction Loader
- New Year Initialization Table Process
- Online Forms Cleanup
- Output File Maintenance
- Paid Check Archiving
- Paid Check Table Restore
- Record-Specific Alerts
- Submit Ready Transactions

- Sync Transaction ECM Reference
- System Maintenance Utility
- Temporary Table Purge
- Transaction Archiving
- Transaction Message Update
- Transaction Unarchive
- Vendor Customer Archiving

2.1.1 Automatic Transaction Correction

Job Name	Automatic Transaction Correction
Recommended Frequency	On-Demand Condition: Setup 1 or more correction rules using the Automatic Transaction Correction Parameters (ADCP) page prior to running this job.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes - Automatic Transaction Correction Activity Report.

Overview

When a group of transactions is loaded from an interface or through a chain job, there is the potential that one or more of the accounting line fields are incorrect, missing, or has a field populated when prohibited. The correction of such transactions can be time-consuming, whether done manually or with SQL statements to change records. CGI Advantage Financial has the Automatic Transaction Correction process and the Automatic Transaction Correction Parameters (ADCP) page to make the correction process quicker, more accurate, and automatic.

Before running the process, you should gather the reasons for transaction rejections using exception reports or opening transactions to view the error messages online. As each problem is identified, a record is added to the ADCP page to select the rejected accounting line(s) and make the necessary correction(s). A single Process ID is assigned to each ADCP record and each record is defined to a unique Record Number. With all ADCP records in place, the batch process is then executed.

The batch process reads in the one or more ADCP records defined to a Process ID to first select accounting lines and then to change accounting line fields. After the process is run and the output report reviewed to ensure all rejecting accounting lines were corrected, the transactions are submitted by an on demand System Maintenance Utility (SMU) job with the Transaction Submit action or processed in the next nightly cycle by an SMU job.

After initial parameter validation, the job locates transactions that match the ADCP settings for Transaction Code, Transaction Department, Transaction ID, Transaction Status, and Transaction Function. In no case will the process select a transaction that has a Transaction Status of *Submitted* or a Transaction Phase other than *Draft*. These are important selection criteria in that multiple runs against a rejected batch of transactions is permitted as transactions corrected and submitted successfully will not be selected in a subsequent run. Please note that to run another batch job against the same Process ID, the Processed On Date must be cleared on the ADCP page.

When at least one draft transaction is selected, the process then goes into accounting line selection. The first selection rule is that an accounting line cannot be closed. Subsequent selection is based on the left-hand column of fields on the ADCP page. The process gathers those ‘selection field’ combinations for all Record Numbers under the Process ID and matches them to open accounting lines. When a match is made the value(s) in the right-hand column of fields is updated to the accounting line. Please be aware that the process does not make updates from Record Number 1 and then perform selection again with Record Number 2 as with the Chart of Account Crosswalk (COAX) feature for rolling open accounting activity. ADCP corrections do not ‘build’ upon earlier corrections.

Any field left empty in the left-hand column of ADCP will not be used for selection. Any field left empty in the right-hand column will not be used in the correction. Only fields with a valid code specified or one of the allowed wildcards will be used for selection or correction. The table below gives each of the selection and correction scenarios:

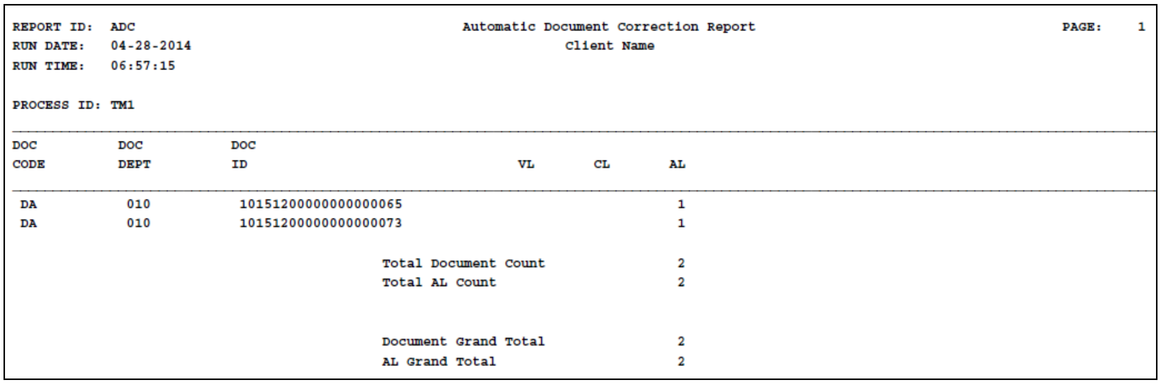
Selection Field (Left)	Correction Field (Right	Result
Code 1	Code 2	When accounting line is found with Code 1 in the field, the value in that field is replaced with Code 2.
BLNK	Code 2	When accounting line is found with no value in the field, Code 2 is placed in that field.
ANY	Code 2	When accounting line is found with a code in the field, Code 2 is placed in that field.
Code 1	BLNK	When accounting line is found with Code 1 in the field, that field is cleared.

When the transaction for correction does not have values for the Fiscal Year, Budget FY, or Period fields seen on the accounting line, selection for those three values will be done using the defaulting values for each field. If consideration is not given to what these defaulting values are (especially Fiscal Year as it is a required selection criteria), then the process will likely not select any records. Those defaults come from three different scenarios:

- Corresponding transaction header field has a value.
- Record Date is populated on the header so the accounting line field is defaulting the value for that date from the Calendar Date record.
- The Application Date is defaulting to the transaction and the field is defaulting the value for that date from the Calendar Date record.

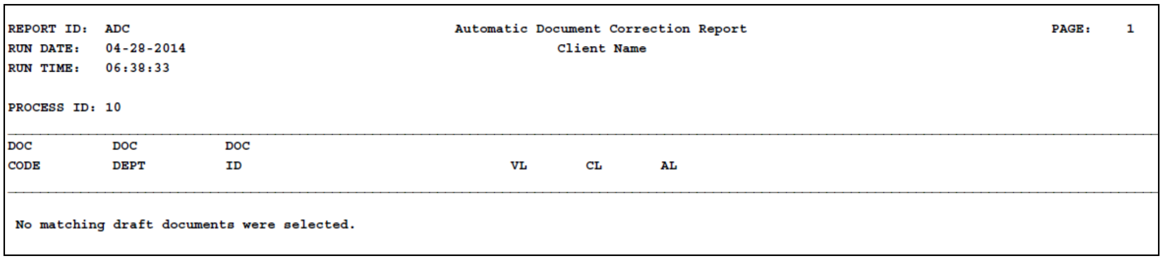
The Period field for selection and correction is critical in one specific scenario and should be specified on ADCP records: Time has passed since the loading of the rejected batch of transactions so that a new period has begun, but that accounting activity should be recorded in the previous period.

After making changes to the accounting lines, the process saves those transaction updates and then generates the Automatic Transaction Correction Activity report. The report lists out each transaction and accounting line combination corrected (see below). If an accounting line is corrected by more than one ADCP record, then it is listed for each record. The total count of transactions and accounting lines in the report should be compared to the count of rejected transactions and accounting lines to ensure all have been selected.



```
REPORT ID:   ADC                                         Automatic Document Correction Report                                             PAGE:     1
RUN DATE:    04-28-2014                                              Client Name
RUN TIME:    06:57:15

PROCESS ID: TM1
------------------------------------------------------------------------------------------------------------------------------------------------------
DOC          DOC           DOC
CODE         DEPT          ID                                   VL        CL        AL
------------------------------------------------------------------------------------------------------------------------------------------------------
 DA           010           10151200000000000065                                     1
 DA           010           10151200000000000073                                     1

                                              Total Document Count                   2
                                              Total AL Count                         2


                                              Document Grand Total                   2
                                              AL Grand Total                         2
```

If no change was made to an accounting line, it is not listed (see below).



```
REPORT ID:  ADC                                Automatic Document Correction Report                              PAGE:    1
RUN DATE:   04-28-2014                                      Client Name
RUN TIME:   06:38:33

PROCESS ID: 10
--------------------------------------------------------------------------------------------------------------------------
DOC          DOC           DOC
CODE         DEPT          ID                                 VL        CL        AL
--------------------------------------------------------------------------------------------------------------------------

 No matching draft documents were selected.
```

The following table contains the different processing steps and the job log messages for each.

Process Steps	Messages
1. Parameter Validation	• Validating batch parameters. • Each parameter is listed followed by any value: • If the Process ID parameter is blank then "Process ID is required and missing." • If the Process ID is not valid on ADCP then "Process ID is invalid." • If the Processed On field for the Process ID on ADCP is populated then "Process ID has already been processed." • If the Commit Block is blank or invalid, the job assumes a value of 100. • Parameter validation completed.
2. Selection of Matching Transactions	• If no matching transactions are found then "No matching draft transactions were found." • If matching transactions are found then "Total transactions updated: ###"
3. Selection of Matching Accounting Lines.	• If no matching accounting lines are found then "No matching draft transactions were found." • If matching accounting lines are found then "Total accounting lines updated: ###"
4. Generate Report	• Reports output folder mapped. • Rendering report started. • Rendering report completed.

Major Input

- Automatic Transaction Correction Parameters (R_ADC_PARM)
- Transaction Accounting Line table(s) (e.g. ABS_DOC_ACTG, PO_DOC_ACTG, etc…)

Batch Parameters

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

Parameter	Description	Default Value

Client Name CLIENT_NM	Optional Client Name to be printed on the report.	(blank)
Process ID PROC_ID	Required Process ID of the Automatic Transaction Correction Parameter record(s).	(blank)
Commit Size COMMIT_BLK	Optional Commit Block Size is a performance parameter to control the number of records saved.	100

Major Output

- Transaction Accounting Line table(s) (e.g. ABS_DOC_ACTG, PO_DOC_ACTG, etc…) and these will push values to the Common Accounting Line (DOC_ACTG) table upon submit/validate actions.
- Automatic Transaction Correction Activity report
- Automatic Transaction Correction Parameters (R_ADC_PARM) – Processed On (PROC_DT) is updated on processed records with current system time and date.

Job Return Code

The following table shows the potential job Return Codes for the Automatic Transaction Correction job:

Return Code	Condition
Successful (1)	Parameter validation was successful, at least one accounting line was selected, and the update to that accounting line was successful.
Warning (4)	• The process was unable to find an accounting line of a draft transaction to match the selection criteria.
Non Fatal Error (8)	• This job does not end with this return code.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid. • ADCP record(s) already processed. • 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 Criteria

- R_ADC_PARM records will be sorted by Transaction Code field.

Selection Criteria

The following are pre-defined selection criteria. All others come from the selected ADCP record(s).

Select where:

- DOC_TYPE <> JV and CLSD_DT = null (*open accounting lines only*)
- DOC_PHASE_CD = 1 *(draft)*

Problem Resolution

The following table shows the possible Return Codes and recommendations for each processing step.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated and records processed successfully.	N/A	N/A
Warning (4)	Job ended with a Warning because there are no matching draft transactions found. **Sample Message:** No matching draft transactions were found.	Review the draft transactions. Modify or add new set of ADCP records and schedule another instance of the job.	
Failed (12)	Job failed due to validations.	If job failed due to parameter validation, reschedule the job with valid parameters values. If job failed due to no ADCP record founds, add new records and reschedule the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job.	
System Failure (20)	When the job is terminated because of database server or network	The reason for the System Failure needs to be investigated before scheduling a new job.	

	issues.		

2.1.2 Batch Output Purge

Description

The Batch Output Purge batch process will utilize user-supplied parameters specified online within the job manager to define the specifics of the purge, and will produce a user-friendly report to transaction all files selected and purged. The new process can be run in report only or report and update mode. This process will replace the existing Batch Cleanup job.

When to Run

This process is run on demand.

Major Input

N/A

Output

- Batch Output Purge report
-

Parameters

Job	Parameter	Description	Default Value
Batch Output Purge	NO_OF_PURGE_DAYS	Number of Days for Purge - enter as a numeric, positive, whole number. Will be subtracted from the current application date to derive the purge eligibility date.	
	REPORT_ONLY	Report Only – enter Y or N	

	PURGE_TYPE	Type of Purge "1" - Report files only, "2" - Non Report files only, "3" - both report and non report files (All).	
	ROOT_PURGE_DIR	Enter the root directory for the Non-Report files to be purged. Required when Type of Purge is 2 or 3. Not allowed when Type of Purge is 1.	
	INCLUDE_SUBDIR	Enter Y to include sub directories during a Non-Report purge, or N to exclude these directories. Required when Type of Purge is 2 or 3. Not allowed when Type of Purge is 1.	
	CLIENT_NM	Client Name for report	

Upon completion the batch should state if the job was successful, what the parameters were, run date and time.

Sort Sequence

None.

Selection Criteria

Selection is done based on the user input parameters, the purge can be of three types: Report Files, Non Report Files, All (Report and Non Report) Files.

- **Purge Type for Report Files**: The process will read the root directory for reports defined in ADV30Params.ini file and get the root location for reports folder, the process will fetch all the files whose modified date is less than or equal to purge validity date. If the process is for report mode details of the file will be written to the report and the process will select next file, if the report mode is Report and Update mode the process will write the details to report file, purge the file and then move on to next file. If a sub-folder becomes empty because of the purge it will be deleted also if the parent folder becomes empty because of the deletion of sub-folder it will also be purged except for the root report folder.
- **Purge Type for Non Report Files**: The process will evaluate the value in "Root Directory for Purge" parameter to define the root directory for the files to be purged; the system will then evaluate the "Include Sub directories" parameter. If the parameter value is 'Y' the process will select files to be purged from the sub-directories also, if the parameter value is 'N' the process will select files only from the root folder specified. If the process is for report mode details of the file will be written to the report and the process will select next file, if the report mode is Report and Update mode the process will write the details to report file, purge the file and then move on to next file. Non Report sub-directories are never deleted during a purge even if the sub-directory becomes empty as a result of the purge.
- **Purge Type for All Files:** The process will purge the report files as well as the non-report files in a single job process by following the procedure mentioned in the above two cases.
- **Excluding Selective Files:** If certain files in folder listed in 'Root Directory for Purge' parameter value should be excluded from the purge process then create a new file called BOP_ExcludeList.txt (case-sensitive) in the same folder and list down the names of files to be excluded, one on each line.

For example, if folder c:/Main is supplied as the Root Directory for Purge parameter value and this folder contains a file called abc.txt and a subfolder called sub that contains another file def.txt and both files need to be excluded from purge process then create a new file called BOP_ExcludeList.txt in c:/Main and list files to be excluded as abc.txt (on first line) and sub/def.txt (on second line).

Problem Resolution

- Look into the job log for errors.
- If the job fails for any data setup reasons in the Batch Output Purge process then correct the data setup and schedule a new job.

2.1.3 Begin Day

Description

Begin Day is a batch job that is available to update the system date parameter through the batch interface. The system date parameter is defined as “APPL_SYS_DT” in the Application Parameters Table, IN_APP_CTRL. This batch job will be the first job that is run in the day to set the “APPL_SYS_DT” parameter value to the current system date. The batch job can optionally be passed the parameter “APPL_SYS_DT” to update the system date to a date other than the current system date.

To reset the application system date, the user can create a Begin Day batch job. An optional System Date can be specified as discussed in the parameters below. If it is not specified, then the current system date is used. Once the job is submitted, the application system date is reset.

The Job return status is set to “Successful” if the application system date has been successfully reset and is set to “Failed” if an exception is encountered.

When to Run

Every day or on demand.

Major Input

A Date value specified as System Date (Optional)

Other Input

None

Output

The value of “APPL_SYS_DT” stored in Application Parameters Table (IN_APP_CTRL) is updated with the system date.

Parameters

Job	Parameter	Description	Default Value
Begin Day	APPL_SYS_DT (Optional)	System Date to begin processing (** **Refer to Note**: Pivot Date/Year Validation, while entering the date)	Current System Date

Sort Criteria

None

Selection Criteria

None

Troubleshooting

If job does not complete, then reschedule the job.

2.1.4 Batch Interface Pre-Processor

Job Name	Batch Interface Pre-Processor
Recommended Frequency	On Demand
Single Instance Required	No
Can be restarted?	N/A
Reports generated	None

Overview

The Batch Interface Pre-Processor performs a three way match among the Batch Interface Event (BIEVNT) record, details provided in the INF file, and transaction/reference table records in the XML file.

The job reads details provided in the INF file and uses the information to select the corresponding BIEVENT record. Matching is done based on the action and matching parameters provided at the job level:

a. The details provided in the INF file and the input XML file.

b. The details provided in the input XML file and BIEVNT record.

c. The BIEVNT record is updated with processing results.

Process Steps	Messages
1. Parameter Validation	• Parameters are valid or invalid depending on the Validation. If a parameter is invalid, a message is displayed in the log.
2. Reads and validates details in the INF file	• If a label is missing, the system displays the following error message: • Label <Field Label> is missing from the INF file • If a value is missing, the system displays the following error message: • <Field> is missing in INF file. • If a value is invalid, the system displays the following error message: • <Field> is invalid in Inf file. • The INF file details and error messages are written in the Error file.

Process Steps	Messages
3. Matching between the INF file and xml file	• If there is a mismatch between the details, the system displays the following error message: • <XML Field Label> does not match the <INF Field Label> on the INF file. • <XML Field Label> does not match <INF Field Label>. • The Transaction Info in the xml Interface file does not match the Batch details on the INF file. • The mismatch error details are written in the Error file. • The key file is created.
4. Matching between the BIEVNT record and xml file	• If there is a mismatch between the details, the system displays the following error message: • <Field Label> from input XML file does not match <Expected Field Label> from BIEVNT record. • <XML Field Label> does not match <BIEVNT Expected Field Label>. • If the BIEVNT record has already been processed, the system displays the following error message: • Duplicate file. The same batch ID has been processed by the pre-processor. • The system displays one of the following error messages, if pre approval is required, but not applied or a BIEVNT record is not found. • Pre Approval is Required. • Pre Approval is Required. Batch Interface Event Entry does not exist.
5. BIEVNT record update	• None

Major Input:

- Batch Interface Event (BIEVNT/BI_EVNT)
- INF File: Contains the information required for mapping with BIEVNT record.
- XML File: Contains reference table /transaction records in Advantage xml format used when table records or transactions are exported.

INF File format:

The INF file carries the information, which helps to link the records in the input XML file to the corresponding BIEVNT record.

- BatchID: Unique ID to identify the set of transactions/Reference table records.
- DeptCode: Department Code
- DocCreator: User ID in the transactions/reference table records.
- UnitCode: Unit Code
- ImportDate: Date on which the records will be imported.
- TransCode: Transaction Code of the transactions being imported applicable only for transactions.
- TableName: Table name of the reference table records being imported applicable for only reference table records.
- TransCount: Number of Transactions/ Reference table records.
- TransDollarAmount: Sum of amount on all the transactions, applicable only for transactions.

Below are examples of INF Files when processing an input containing transaction or reference table records.

Transaction:

```
BatchID=NFTEST4A;
DeptCode=010;
DocCreator=sa;
UnitCode=001;
ImportDate=2023-05-05;
TransCode=CR;
TableName=;
TransCount=200;
TransDollarAmount=2540;
```

Reference Table:

```
BatchID=NFTEST4A;
DeptCode=010;
DocCreator=sa;
UnitCode=001;
ImportDate=2023-05-05;
TransCode=;
TableName=R_FUND;
TransCount=200;
TransDollarAmount=;
```

Batch Parameters:

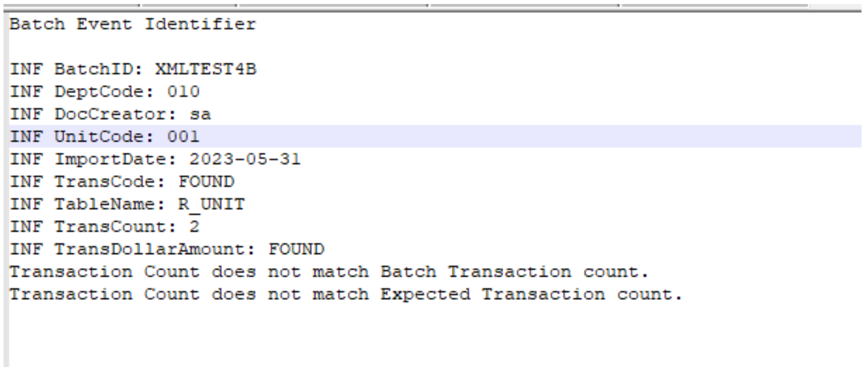
Parameter	Description	Default Value
ACTION	Required. Enter Amount Total Action (1 -Sum, 2 -Sum of Absolute Values, 3 -Sum of only Negative Line). This field controls how amount will be calculated when the input XML contains transactions.	1
AMSEXPORT	Required. Export Location at Batch Interface Pre-Processor Job	$$AMSROOT$$/ExportImport
AMSIMPORT	Required. Import Location at Batch Interface Pre-Processor Job	$$AMSROOT$$/ExportImport
AMSLOGS	Required. Logs Location at Batch Interface Pre-Processor Job	$$AMSROOT$$/Logs
AMT_FIELD	Optional. Amount Total field which will be used for calculating the amounts based on Action provided.	
APRV_REQ	Optional. Approval Requirement (1- Pre, 2- Post and 3-None). This Parameter controls whether admin approval on the BIEVENT record is required before the job is processed, is required after the job is processed, or is not required.	1
COMP_LVL	Optional. The Component containing the amount field used for the calculation of the amounts.	
ERR_FILENM	Required. Name of the File to which the error details will be written too.	
INF_FILENM	Required. Name of the INF File to be used as an input for the job.	
KEY_FILENM	Required. Name of the file containing output identifying input table records or transaction records that were processed.	
MATCH_DEPT	Optional. Match Department (1-Yes, 2- No). Controls whether the department code provided on INF file should match with the department code provided in the XML file.	1
MATCH_REC_COUNT	Optional. Match Record Count (1-Yes, 2- No)	

	Controls whether the value of the TransCount provided in the INF file should match with the number of reference table records in the XML file when the input contains table records. Either MATCH_REC_COUNT or MATCH_TRAN_COUNT should be provided, not both.	
MATCH_TOT	Optional. Match Total (1- Yes, 2- No) Controls whether the Value of TransDollarAmount in the INF file should be matched with the Amount calculated from the provided XML based on the Action selected.	1
MATCH_TRAN_COUNT	Optional. Match Transaction Count (1- Yes, 2- No) Controls whether the value of the TransCount provided in the INF file should match with the number of transactions in the XML file when the input contains transactions. Either MATCH_REC_COUNT or MATCH_TRAN_COUNT should be provided, not both.	1
XML_FILENM	Required. Name of the XML file that holds the reference table records / transactions.	

Major Output:

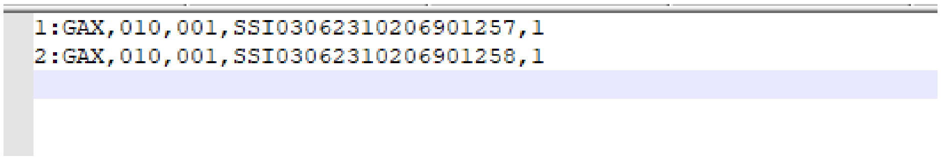
- Batch Interface Event (BIEVNT/BI_EVNT)
- Error File: Contains error related to matching.
- Key File: Contains the primary key details to identify a Transaction or reference table record.

Below is an example of a generated error file**:**



```
Batch Event Identifier

INF BatchID: XMLTEST4B
INF DeptCode: 010
INF DocCreator: sa
INF UnitCode: 001
INF ImportDate: 2023-05-31
INF TransCode: FOUND
INF TableName: R_UNIT
INF TransCount: 2
INF TransDollarAmount: FOUND
Transaction Count does not match Batch Transaction count.
Transaction Count does not match Expected Transaction count.
```

Below is an example of a generated Key file,



```
1:GAX,010,001,SSI03062310206901257,1
2:GAX,010,001,SSI03062310206901258,1
```

Batch Interface Event (BIEVNT) updates

- Approval Requirement: Pre, Post, or None
- Approval Status: Pending (Post) or Not Required if BIEVNT record has not been approved and Approval Requirement is Post or None.
- Actual Received Date/Time: Date and time of processing.
- Actual Transactions: Count of reference table/transaction records if matching was performed for count.
- Actual Total: Total amount based on the action selected in job parameters if matching was performed for total.
- Processing Status is updated to indicate the results of the matching. If there are no issues found with matching the details then:
    - Processing Status: Received/Processed
    - If there is any mismatch between the Expected Transactions, Expected Total on the BIEVNT record and the Actual Count, Actual Amount calculated from the input xml, then the Processing Status is updated to Received /Rejected. In this scenario, discepancy between the BIEVNT record and input xml needs to be addressed before processing the record.

Note: A BIEVNT record can be processed only once. If any discrepancy is found, a new BIEVNT record needs to be created along with new Batch ID, which also needs to be updated in the INF file.

Sort Criteria

None

Selection Criteria

Based on the Batch ID, Department Code, Unit Code, Import Date, and Transaction code or Table Name provided in the INF file, select the BIEVNT record for processing.

Problem Resolution

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

Return Code	Condition
Successful (1)	All the details provided in the BIEVNT record and input INF file match with the details obtained from the input xml file.
Warning (4)	No scenario.
Non-Fatal Error (8)	When there is a mismatch between the BIEVNT record/INF file with the input xml file.
Failed (12)	The job fails under the following conditions: • Parameters are invalid. • If any of the field labels missing from INF file. • If any of the required values is not available. • Run time exceptions for unexpected situations.
Terminated (16)	This Return Code is displayed when the job is terminated by the user.
System Failure (20)	This Return Code is displayed when the job is terminated because of database server or network issues.

2.1.5 BPRO To CACT Updates

Chain or Job Name	BPRO To CACT Updates
Recommended Frequency	On Demand
Single Instance Required	No
Can be restarted?	N/A
Reports generated	CACT Exception Report

Overview

This is an offline process that is automatically submitted by the system when a Billing Profile (BPRO) record has the Billing Type changed where the number of associated Customer Account Option (CACT) records is equal to or greater than the Application Parameter: CACT_COUNT_FOR_OFLN_UPD. The processing time of such a change that triggers several edits and subsequent updates can contribute to the system timing out if the changes have to be done in real-time.

The process of changing a Billing Type may be an iterative one. When the process cannot successfully update all the CACT records, it rolls back all that could have been updated as well as the BPRO update. A user must then access the exception report, address the errors on CACT individually, and then change the Billing Type on BPRO again to start another round of updates.

Process Steps	Messages
1. Selection of Records	• Selecting eligible records • If the selection returns 0 records, then the following message is issued: "No eligible record found". • Number of records (count) selected is displayed. • At the end, the following message is issued: "Selection of records completed."
2. Update CACT records with Billing Type from BPRO	• Records associated with a respective BPRO are identified with Department, Unit and BPRO. • Identified records are then updated with changed Billing Type, which is passed as a parameter to job.
3. Check for exceptions	• When updating CACT records, the system verifies if there are any exceptions encountered. • If any of the CACT records have exceptions, changes on CACT and BPRO are rolled back.
4. Rollback BPRO updates	• If any exceptions are encountered while updating CACT records, changes made to BPRO are rolled back with the help of the BPROToCACTUpdates.csv file generated before triggering the job, which would contain old values for the BPRO record.
5. Report	• CACT records with exceptions are written into the exception report.

Major Input

- Billing Profile (BPRO / R_BPRO)

Note: The default values listed are those delivered with the software. No need to change the defaults as they are populated by the system.

Parameter	Description	Default Value
Billing Profile (BPRO_CD)	A required selection parameter passed to the process by the system.	No Default
Department (DEPT_CD)	A required selection parameter passed to the process by the system.	No Default
Unit (UNIT_CD)	A required selection parameter passed to the process by the system.	No Default
BPRO File Name (BPRO_FILE_NAME)	System-generated file with old and new Billing Type values of the Billing Profile updated.	BPROToCACTUpdates.csv

Major Output

- Customer Account Options (CACT / R_CUST_ACCT_OPT)
- CACT Exception Report

Job Return Code

Return Code	Condition
Successful (1)	All of the selected CACT records are updated successfully with no errors or severe messages.
Warning (4)	This return code is not issued for this job.
Non-Fatal Error (8)	This return code is not issued for this job.
Failed (12)	This return code is issued when any of the CACT records updated fails.
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

None

Selection Criteria

CACT records associated with a BPRO record are selected based on the job parameters BPRO_CD, DEPT_CD and UNIT_CD.

Problem Resolution

If the job ends with a return code other than Successful after completion, a new job should be scheduled after the job log has been reviewed (the failed job should not be restarted). If the report generated has any exceptions related to severity level Error or Severe, those records should be rectified.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the CACT records are updated successfully	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non-Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Job failed due to exceptions greater than severity level Warning were caught while updating CACT records Failed because of runtime exceptions for an unexpected situation.	Errors reported with severity greater than warning need to be addressed on the CACT records and updated. Failure reason needs to be investigated before scheduling a new job.	After errors are addressed, make the BPRO change again. N/A
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.	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 either be restarted or schedule a new job.	N/A

2.1.6 Budget Archiving

Chain Name	Budget Archiving
Recommended Frequency	On Demand Recommend using the Pre-Archive Budget Report prior to running to verify all activity has completed to a satisfactory level for archiving.
Single Instance Required	Yes
Can be restarted?	N/A
Reports generated	Budget Archiving Statistics Report Budget Archiving Detail Report Budget Archiving Facilitator Report

Overview

With the progression of time, the data built up in the budgeting area of Advantage Financial becomes considerable. While such data is necessary for current budget lines as well as ones in the recent past, there are budget lines which are seldom, if ever, researched or reported on. The Budget Archiving process archives data from budget level and related tables as CSV or XML files using export functionality and purges the system of data no longer required.

> **Note 1**: Budget lines from structures with a BFY in their definition can only be archived if the FY equal to that BFY has been closed by the Annual Close process. There is a batch edit for this, but just because an FY is closed does not mean that the BFY is also closed to activity. The fact that a BFY is closed to activity should be confirmed through BFY Staging setup and assurances that no more adjustments are required. To archive off a budget line that is not complete will result in accounting transactions returning budget line not found errors.
>
> **Note 2**: Budget lines from structures without a BFY or have BFY 9999 need additional assurances that all activity for that budget line is completed.
>
> **Note 3**: Running the Pre-Archive Budget Report is a useful tool to find out if there are any pending transactions against a budget line. The report can also list transactions that update a bucket such as Encumbrances or Billed Earned Revenue where there is likely to be another downstream transaction.

Once parameters are validated and records are selected for a given set of parameters, facilitator records are submitted to the Job Manager, where the records from budgeting tables are exported using parallel processing. Once the process has successfully completed, the system will purge the archived records from the budget and related tables.

Selected records are archived and deleted from these tables:

1. Budget Levels (BUD_STRU_##_LVL_##)
2. Allotments (ALOT_STRU_##_LVL_##)
3. Activity Levels (BUD_STRU_##_LVL_## and ALOT_STRU_##_LVL_##) – deleted but optionally archived)
4. Budget Journal (JRNL_BUD)
5. Links (GN_LNK)
6. Link History (GN_LNK_HIST)
7. Budget Line Controls (GN_LN_CNST)

Selected records from these related tables are only deleted:

6. Systems Assurance 1 Shadow Table (SA_BUD)
7. COA Inference Pages (BFYINF, APPRINF, APBYINF) – Budget Fiscal Year, Appropriation, and BFY and Appropriation
8. COA Required Element Pages (RB44L3, RB45L3) – Budget 44/45 Level 3 Requirements
9. Any custom inference or required element pages where the deletion of a budget line deletes the inference or required element record too.

The following job steps comprise the Budget Archiving chain process:

- Archiving Preprocessor
- Archive Facilitator
- Post Archiving Process

The steps are all singleton jobs. This means that only one of their instances can be run at a time. The Archive Facilitator does have the ability to launch System Maintenance Utility jobs in parallel.

The output from the process includes three reports:

1. Budget Archiving Statistics Report showing a count of selected records per table.
2. Budget Archiving Detail Report (optional) listing all budget lines selected at level 1 of the structure archived.
3. Budget Archive Facilitator Report listing each archive file created and the count of records per file. When run in Report Only mode, this report is not created as the chain ends after the Archiving Preprocessor completes the Statistics report (and the Detail report, if indicated by parameter).

The acceptable job return codes (configured in the Configure Chain Job section of the Job Setup in CGI Advantage) for the jobs in the Budget Archive chain are delivered to be set to Successful. As with all CGI Advantage chain jobs, these acceptable return codes are configurable and may be changed to meet certain requirements.

This chain implements the Data Warehouse Archived Record Queue process, which is used to retain the table names and key values for records that are archived and deleted. The infoAdvantage reporting system uses this information to distinguish between records that have been archived (as valid historical data) before deletion and records that have simply been deleted from the system (as no longer needed). The process is enabled for most archiving processes when the value of Application Parameter Enable Data Warehouse Archived Record Queue (ENABLE_DW_ARCH_QUEUE) is *True*. Please see the *CGI Advantage System Administration Guide* for more information regarding this process and the Application Parameter.

Archiving the input source to the System Assurances 11 and 12 processes requires additional steps after archiving. Those steps are running of the Systems Assurance 11 and 12 in full mode with any BFY that has just been archived. This will remove data for that BFY from the shadow tables used by each assurance. There is no impact to System Assurance 1 as the archive program removes data from the shadow table used by that program.

Chain Job Return Code

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

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

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

Problem Resolution

Please refer to the individual job "Problem Resolution" section for more details.

Budget Archiving Chain: Archiving Preprocessor Job

Job Name	Archiving Preprocessor
Recommended Frequency	On Demand This job must be run as part of the Budget Archiving Chain.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Budget Archiving Statistics Report Budget Archiving Detail Report (optional)

Overview

This step first validates parameters. If parameters are valid, it then identifies the rows to-be-archived based on the selection criteria and divides the selected rows in each table to-be-archived into groups for each export file. For example, if SA_BUD has 20,000 rows for BFY 2000 and the Job Block Size parameter is 10,000, then two export files will be created.

The Budget Archiving Statistics Report lists the total number of records eligible for archival from each table. If the Budget Line Listing parameter is set to 'Y', the Chart of Accounts (COA) key fields for all selected level 1 records of the selected budget structure are written to the Budget Archiving Detail Report. If the run mode is set to 'Report Only', the process will end at this point.

In 'Full Run' run mode, a System Maintenance Utility parameter file is created for each export file to be created. The parameter file contains the table name, the identification of specific records, and other settings. A Facilitator table record referencing the SMU parameter file is created and saved. Facilitator records can be viewed in the Archive Facilitator Inquiry page (QARC) for the status. The file names for the archived records are written to the chain parameter file for use by the Post Archiving Process job of the chain.

Process Steps	**Messages**

10. Parameter Validation	• Run Started • Each parameter is listed • Validating Batch Parameters • If any is found to be invalid an error will also be issued • Batch Parameters are valid. or • Batch Parameters are not valid.
11. Pre-processing	The following messages will be issued when the job runs in Full Run mode. • Record selection started. Each table is listed with the number of records identified • Record selection completed. • Rendering Statistics report started. • Reports output folder mapped (followed by the HTML & PDF locations) • Rendering Statistics report completed. • Rendering Detail report started (*optional – by parameter*). • Reports output folder mapped (followed by the HTML & PDF locations) • Rendering Detail report completed (optional – by parameter). • Writing chain parameter file. • Chain parameter file (Chain Parameter File Name) written in directory (CGI Parameter Directory). • Number of records deleted from temporary table: (number of records) • Run Ended The following messages will be issued when the job runs in Report-Only mode. • Record selection started. • Record selection completed. • Rendering Statistics report started. • Reports output folder mapped (followed by the HTML & PDF locations) • Rendering Statistics report completed. • Rendering Detail report started (optional – by parameter). • Rendering Detail report completed (optional – by parameter). • Number of records deleted from temporary table: (number of records)

	• Run Ended

Major Input

- Budget Levels (BUD_STRU_##_LVL_##)
- Allotment Levels (ALOT_STRU_##_LVL_##)
- Activity Levels (BUD_STRU_##_LVL_##) – Budget & Allotment Lines
- Budget Journal (JRNL_BUD)
- Budget Links (GN_LNK)
- Budget Link History (GN_LNK_HIST)
- Budget Line Controls (GN_LN_CNST)
- Systems Assurance 1 Shadow Table (SA_BUD)
- COA Inference & Required Element tables (INF_BFY, INF_APPR, INF_APBY, INF_APPR2, INF_APBY2, INF_APPR_RSRC, CVIN_STRU44L3_REQ, CVIN_STRU45L3_REQ)

Batch Parameters

Parameter	Description	Default Value
ACTIVITY_ARCHIVE	Activity Level Archive Required. Enter 'Y' to archive and purge the Activity Level table. Enter 'N' to purge only.	No Default
AMSEXPORT	Export Location at Archiving Preprocessor Job Required. This must be a valid directory and will be used to determine where the export files will be written. (The size of archived files may be so great that a location other than the default Export/Import directory should be used.)	$$AMSROOT$$/ ExportImport
AMSPARM	Parameter Location at Archiving Preprocessor Job Required. This must be a valid directory containing the file designated by the Parameter File parameter.	$$AMSROOT$$/ Parms
BFY	Budget Fiscal Year Conditionally Required. If the budget structure is keyed by BFY, then this parameter is required.	No Default

	For Multi-Year budgets, this parameter is prohibited.	
BUD_STRU_ID	Budget Structure ID Required. Must be a valid ID in the Budget Structure (BUDST) table.	No Default
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) Required, not overrideable. This file is created during the run to convey parameters to the subsequent steps in the chain.	BudArchParams.txt
CLIENT_NM	Client Name for Reports Optional.	No Default
COMMIT_BLOCK_SIZE	Commit Block Size Required. This value will determine how often database transactions will be committed and can be used for performance tuning. The value must be a positive integer.	1000
DETAIL_BL	Budget Line Listing Required. Enter 'Y' to produce a listing of all level 1 budget lines selected. Enter 'N' to suppress the report.	N
EXP_FILE_TYP	Export File Type Required. Enter '1' for XML, '2' for CSV.	2
FILE_PREFIX	File Prefix Required. Prefix used for archive file names.	No Default
PARM_FILE	Input Selection File Required. Enter the name of the parameter file for budget archiving; this file must exist in the directory specified by the Parameter Location at Archiving Preprocessor Job parameter.	BudArchiveSelect.txt
JOB_BLOCK_SIZE	Job Block Size Required. The maximum number	10000

	of lines in an output file	
JOB_TYP	Job Type Required, not overrideable. Indicates to the system whether to run shared logic as the Pre-Archive Report job or as part of the Budget Archiving chain job. This must always be set to Archive for the Budget Archiving process.	Archive
RUN_MODE	Run Mode Required. Enter '1' for Report Only, '2' for Full Run.	1
SMU_FILE_PREFIX	SMU Facilitator Parameter File Prefix Required, not editable. This value will be used to determine a SysManUtil input parameter file name for each Facilitator job.	BudArch

Major Output

- Budget Archiving Statistics Report
- Budget Archiving Detail Report (optional – by parameter)
- Budget Archiving Temporary (BUD_ARCH_TMP) table
- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page.
- BudArch##_#.txt – File created to be passed to SMU to perform the archive. The ## is the chain job ID and the # is an incrementing number for each table archived.
- ARCH_DW_QUEUE_REC – The Data Warehouse Archive Record table keeps track of the values for all primary keys of each record being archived and deleted. The table name and attribute names are stored in ARCH_DW_QUEUE_TBL and linked to this record by unique ID.
- ARCH_DW_QUEUE_TBL – The Data Warehouse Archive table keeps track of the table name and primary key attribute names of each table being archived. If a record already exists for the table being archived, it will reuse the record. If it does not yet exist, it will be created.

Minor Output

- BudArchParams.txt – To pass information from the first job in the chain to later jobs.
- Folder in ExportImport/Budget Archive

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	This return code is issued under the following condition: - The Run Mode parameter is set to '1' (Report Only).
Non Fatal Error (8)	This return code is issued under the following condition: - No eligible records found in the level 1 table of the budget structure for the parameters entered.
Failed (12)	The job fails under the following conditions: One or more parameter(s) are invalid. - One or more required parameter(s) are not entered. - The Input Selection File is not found in the directory specified by the Parameter Location at Archiving Preprocessor Job parameter. - Run time exceptions for unexpected situations. When this job ends with a return of code Failed, subsequent jobs in the chain are set to Inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain are set to Inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return of code System Failure, subsequent jobs in the chain are set to Inactive.

Sort Criteria

None

Selection Criteria

The Input Selection File parameter identifies a text file that is required to be present in the parameter directory. It can be empty when the budget is keyed by BFY, but at least one set of selection criteria must be specified for no-year budgets (those without a BFY in their line definition). Budget lines that are defined to BFY 9999 will require the entry of 9999 in the BFY batch parameter.

Each set of selection criteria must start with the _PARAM_LINE_ separator. Selection criteria values in the input file may use commas to specify more than one value per COA element (but only one comma-separated list of values is allowed per PARAM LINE section). The "_" (underscore) may be used as a wildcard to match any one character. The "%" (percent sign) may be used as a wildcard to match 0 to *n* characters.

While the parameter file does allow for complex selection criteria, the easiest method to build a parameter file (and avoid parameter file edits for what is not allowed) is to run a series of database queries to get a list of UNID values from level 1 of the budget structure being reported on. Each budget line is uniquely defined by this UNID value and it is very straightforward to put into the file under one parameter line. Please exercise care not to exceed any limits for SQL IN conditions for the type of database. Such limits are often 1000, which is most likely higher than the manageable number of budget lines selected from level 1 of a budget structure using COA instead of BFY.

Input File Example:

```
_PARAM_LINE_
FUND_CD=010
DEPT_CD=010
APPR_CD=0028,0029,0030
_PARAM_LINE_
FUND_CD=150
DEPT_CD=020
APPR_CD=0001
_PARAM_LINE_
FUND_CD=20_
DEPT_CD=010,020
APPR_CD=1%
_PARAM_LINE_
UNID=22390
_PARAM_LINE_
UNID=22441
```

Within each _PARAM_LINE_ group, the selection criteria are grouped as AND statements. Then, the _PARAM_LINE_ groups are grouped as OR statements to form the full selection criteria. The above example would be interpreted as:

(Fund = ‘010’ **and** Department = ‘010’ **and** (Appr Unit is ‘0028’ **or** ‘0029’ **or** ‘0030’))

or (Fund = ‘150’ **and** Department = ‘020’ **and** Appr Unit = ‘0001’)

or (Fund matches ‘20_’ **and** (Department is ‘010’ **or** ‘020’) **and** Appr Unit matches ‘1%’)

or (UNID = 22390)

or (UNID = 22441)

It is acceptable to have a given record satisfy the criteria for multiple _PARAM_LINE_ groups. This may occur depending on how broad or specific the selection criteria are defined for each group. When this situation occurs, the Archiving Preprocessor will properly select the record just once. Please exercise care not to exceed any limits for SQL IN conditions for the type of database. Such limits are often 1000, which is most likely higher than the manageable number of budget lines selected from level 1 of a budget structure using COA instead of BFY.

Input File Rules:

1. The input file cannot contain BFY, because the process requires a BFY parameter to be specified where applicable, and is otherwise prohibited.
2. The input file must contain at least one PARAM LINE section with one key field for selection and at least one value for that key field if the budget structure is not keyed by BFY.
3. The input file can only include budget level 1 COA elements list or UNID.

4. If the selection criteria key is not UNID, then the full key to a COA element must be specified (except FY). For example, running with just Program will fail if Department is not also specified. Selecting on Unit will fail if Department is not also specified, but there does not have to be an FY, which is a key to each Unit.
5. The PARAM LINE section will be controlled several ways to prevent performance degradation but allow for some flexibility in selection when a COA list is specified:
   d. Wildcard and comma features cannot be combined into the same line, so the following is not allowed:

      _PARAM_LINE_

      DEPT_CD=010,02%

   e. Multiple lines may use the wildcard feature, so the following is allowed:

      _PARAM_LINE_

      DEPT_CD=01_

      MJR_PROG=100%

   f. Multiple wildcards are allowed in a single line, so the following is allowed:

      _PARAM_LINE_

      DEPT_CD=010

      MJR_PROG=%100%

   g. Only one line can use the comma feature, so the following is not allowed:

      _PARAM_LINE_

      DEPT_CD=010,02

      0MJR_PROG=10011, 10012

   Instead, the same can be achieved with multiple lines, for example:

      _PARAM_LINE_

      DEPT_CD=**010**

      MJR_PROG=10011,10012

      _PARAM_LINE_

      DEPT_CD=**020**

      MJR_PROG=10011,10012

6. Wildcard characters are prohibited when selecting by UNID.

Problem Resolution

If the job ends with a return code other than Successful or Warning after completing parameter validation, a new job should be scheduled after the job log has been reviewed (the failed job should not be restarted).

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)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. Sample Message: Parameter BFY is required for Structure 29.	Schedule a new job after entering a valid value for the parameter.	N/A
	Entered parameters are not valid. Sample Message: Parameter Job Block Size value 'ABC' is invalid, it must be a positive integer.	Schedule a new job after entering a valid value for the parameter.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Pre-processing

Possible Return Codes	Condition	Recommendation	Other Instructions

Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	The Run Mode parameter is set to '1' (Report Only). This is a normal condition that sets any subsequent jobs to inactive.	N/A	N/A
Non Fatal Error (8)	No tables need to be archived.	Confirm selection criteria before scheduling a new job.	The Pre-Archive Budget Report is a good tool for establishing correct selection criteria.
Failed (12)	Failed during attempt to write the chain parameter file.	Verify that the file system is not full, has the correct access, and so forth, before scheduling a new job.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Budget Archiving Chain: Archive Facilitator Job

Job Name	Archive Facilitator
Recommended Frequency	On Demand This job must be run as part of the Budget Archiving Chain.
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	None

Overview

The Facilitator job spawns multiple System Maintenance Utility (SMU) jobs to export (archive) selected Budget records to files. This step does **not** delete records from the Budget tables. The Archive Facilitator step retrieves the records from the Facilitator table for the chain's Job ID. It then reads the file names from these records and spawns multiple SMU jobs, using the parameter file names from the Facilitator table as input for the SMU jobs. The parameter files specify the "Table Export" command with other options set in the Archive Preprocess step. Each SMU job is responsible for exporting (archiving) table records from a single table. Multiple SMU jobs may be spawned to export data from a single table (depending on the Archiving Preprocessor Job Block Size parameter value), but one SMU job will never be responsible for exporting from more than one table.

Process Steps	Messages
1. Parameter Validation	• Run Started Each parameter is listed
2. Facilitator Job Processing	• The Run Number for this archive/restore process = xxx (xxx being the chain's Job ID. Restore is not a feature of Budget Archive.) • For each job in the Facilitator for the Run Number: • SMU Job - xxx - Spawned (*xxx being the Job ID)* • Each spawned job will have either of these results: • SMU Job - xxx – Processing completed successfully (*xxx being the Job ID)* • SMU Job - xxx - Failed *(xxx being the Job ID)* • The job slept a total of xxx times, for a Total Sleep Time of yyy seconds. *(xxx and yyy being the counts)* • Run Ended

Restartability Information

This job can be restarted if it fails due to any reason. After restart, the job will continue to process the records based on the status of each Facilitator record. Restarting this job step will restart all spawned jobs automatically.

Major Input

- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page.
- BS_AGENT – The job will consult the Job Manager for the status of jobs that have been spawned.

- Budget Level and related tables – Each SMU job spawned by the Facilitator job will export a subset of records, from a single table, depending on the starting and ending keys specified in the parameter file supplied to the Facilitator record.

Batch Parameters

Parameter	Description	Default Value
ARCHIVE_RESTORE_ID	Archive Restore ID Required, not overrideable. This value tells the Facilitator to select the archive table process for Budget tables.	9 (Budget Archive)
COMMIT_BLOCK_SIZE	Commit Block Size Required. This value will determine how often database transactions will be committed and can be used for performance tuning. This must be a positive integer.	1000
PROCESSOR_NO	Number of jobs for Facilitator to keep running Required. This value sets the number of jobs to launch and keep submitted/running. Exceeding the number of available VLS is acceptable. This must be a positive integer.	1
SLEEP_TIME	Number of seconds to wait between polling occurrences Required. This value sets the length of time between iterations, which checks the status of the jobs that are running, and the jobs that are launching for unprocessed Facilitator records as necessary.	5
UPDATE_STATUS	Update Status Required, not overrideable. This value tells the Facilitator to update the status of each Facilitator record based on the status of its corresponding job in the Job Manager.	Y

Major Output

- Exported CSV or XML files

- FACILITATOR – The Facilitator table is updated to reflect the Job Manager status.

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	This return code is issued under the following conditions: - No Facilitator records found for Run Number = xxx. Nothing to process. (xxx being the chain's Job ID) - SMU Job Parameter File NOT Found – (file name) Record skipped.
Non Fatal Error (8)	This step does not issue this return code.
Failed (12)	The job fails under the following conditions: One or more parameter(s) are invalid. - One or more required parameter(s) are not entered. - Parameters are invalid. - Run time exceptions for unexpected situations. When this job ends with a return of code Failed, subsequent jobs in the chain are set to Inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain are set to Inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return of code System Failure, subsequent jobs in the chain are set to Inactive.

Sort Criteria

None

Selection Criteria

FACILITATOR records contain the location of input parameter files for use by spawned System Maintenance Utility batch jobs.

Problem Resolution

No database restore is required. Both the Archive Facilitator chain step and each spawned System Maintenance Utility job can be restarted.

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)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	No Facilitator records found for Run Number (Nothing to process).	This is acceptable in the event that there are no records to process. A new chain can be started with only the Post Archiving Process enabled. Otherwise, the reason for the missing records must be investigated before a new chain job is run.	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. Sample Message: SLEEP_TIME is required to run the Facilitator	Schedule a new job after entering a valid value for the parameter.	N/A
	Entered parameters are not valid. Sample Message: The SLEEP_TIME must be an integer and > 0	Schedule a new job after entering a valid value for the parameter.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A

Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Facilitator Job Processing

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	SMU Job Parameter File not found.	If all files are missing, the reason must be investigated before a new chain job is run. If the files can be located and moved to the expected location, the job can be restarted.	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	This return code will be issued when any spawned SMU job fails. Sample Message: SMU Job – xxx – Failed	Determine the cause for the spawned SMU job to fail and resolve. The spawned job can be restarted independently if desired, and restarting the chain job will also restart any failed SMU jobs.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Budget Archiving Chain: Post Archiving Process Job

Job Name	Post Archiving Process
Recommended Frequency	On Demand This job must be run as part of the Budget Archiving Chain.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Budget Archive Facilitator Report

Overview

The parameter directory and the chain parameter file are the only two parameters for this step. All of the user parameters that this step needs are stored in the designated file by the Preprocessor. The first action of this step of the chain is to open the chain parameter file and read these parameters. The second action is to confirm that all facilitator records were processed successfully in step 2 of the chain job. If any facilitator job did not finish successfully, the process ends. The third action is to delete the selected records from the budget and related tables and to write the Budget Archive Facilitator Report.

Process Steps	Messages
1. Parameter Validation	• Run Started Each parameter is listed • Validating Batch Parameters If any is found to be invalid an error will also be issued • Batch Parameters are valid.
2. Post-processing	• Retrieving table information for budget structure (Budget Structure ID). • Retrieving table information completed. • Confirming Facilitator status for Run ID (Run ID of the current Chain Job). • Facilitator completed (number) of (number) job(s) for Run ID (Run ID of the current Chain Job). • If the number of jobs completed is not the same as the total number of jobs: • Post Archive deletion cannot proceed until all Facilitator jobs for Run ID (Run ID of the current Chain Job) have been completed. (Process ends here in this case)

	• Facilitator status check completed. • Deletion of selected records started. • At this point, a series of progression messages based on the commit block size will appear. Sample messages: • Deleted 1000 records from BUD_STRU_29_LVL_1. • Deleted 2000 records from BUD_STRU_29_LVL_1. • Deletion of selected records completed. • Rendering Facilitator report started. • Reports output folder mapped (followed by the HTML & PDF locations) • Rendering Facilitator report completed. • Number of records deleted from temporary table: (number of records) • Run Ended

Restartability Information

This job cannot be restarted. If the job failed due to any reason, a new job can be scheduled after correcting errors that caused the job to fail – either individually, or as part of a new chain job by disabling the first two jobs in the chain and specifying the same set of parameters.

Major Input

- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page. This job will check the table to see if any records remain that have not completed successfully.
- BS_AGENT – The job will consult the Job Manager for the status of jobs that have been spawned.
- Chain Parameter File – The Preprocessor step writes this file to communicate all of the settings that this step needs.

Batch Parameters

Parameter	Description	Default Value
AMSPARM	Parameter Location at Archiving Preprocessor Job Required. This must be a valid directory containing the file designated by the Parameter File	$$AMSROOT$$/ Parms

	parameter.	
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) Required, not overrideable. This file is created during the first job in the chain to convey parameters to this process.	BudArchParams.txt

Major Output

- Budget Levels (BUD_STRU_##_LVL_##) – Selected records are deleted from these tables.
- Allotments (ALOT_STRU_##_LVL_##) – Selected records are deleted from these tables.
- Activity Levels, Budget & Allotment Line (BUD_STRU_##_LVL_##) – Selected records are deleted from these tables.
- Budget Journal (JRNL_BUD) – Selected records are deleted from this table.
- Budget Links (GN_LNK) – Selected records are deleted from this table.
- Budget Link History (GN_LNK_HIST) – Selected records are deleted from this table.
- Budget Line Controls (GN_LN_CNST) – Selected records are deleted from this table.
- Systems Assurance 1 Shadow Table (SA_BUD) – Selected records are deleted from this table.
- COA Inference & Required Element tables (INF_BFY, INF_APPR, INF_APBY, INF_APPR2, INF_APBY2, INF_APPR_RSRC, CVIN_STRU44L3_REQ, CVIN_STRU45L3_REQ)
- Budget Archiving Temporary (BUD_ARCH_TMP) table – all rows inserted to this table by the pre-processor step are deleted in this step.

Job Return Code

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

Return Code	Condition
Successful (1)	The SMU jobs were confirmed to have completed, and all selected records were deleted.
Warning (4)	This step does not issue this return code.
Non Fatal Error (8)	This step does not issue this return code.
Failed (12)	The job fails under the following conditions: One or more parameter(s) are invalid.

	- One or more required parameter(s) are not entered. - Parameters are invalid. - Run time exceptions for unexpected situations. When this job ends with a return of code Failed, subsequent jobs in the chain are set to Inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain are set to Inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return of code System Failure, subsequent jobs in the chain are set to Inactive.

Sort Criteria

None

Selection Criteria

BUD_ARCH_TMP records identify the tables and their individual records to be deleted. The temporary records are inserted by the Preprocessor step during the record selection process and are keyed by the Chain Job ID from the Job Manager. This step also removes all temporary records after the archive deletions have been completed successfully.

Problem Resolution

Since the job cannot be restarted, if the job ends with any return code (Failed, Terminated or System Failure) a new chain job with the first two jobs in the chain disabled should be scheduled with the same parameters.

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)	The SMU jobs were confirmed to have completed, and all selected records were deleted.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A

Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. Sample Message: Parameter AMSPARM directory is required and must be a valid directory.	Enter the correct directory for Parameter Location and schedule a new chain job with only step 3 enabled.	N/A
	Entered parameters are not valid. Sample Message: Parameter Chain Parameter File is required to have file extension "txt".	Enter the correct file name for Chain Parameter File and schedule a new chain job with only step 3 enabled.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Post-processing

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

	code.		
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new chain job with only step 3 enabled.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

2.1.7 COA Usage Verification

Chain or Job Name	COA Usage Verification
Recommended Frequency	On Demand
Single Instance Required	No
Can be restarted?	No
Reports generated	None

Overview

COA Usage Verification is an optional batch process that is typically launched, automatically, by online COA reference pages, as a background process, when an attempt is made to delete a record. This batch job researches whether a COA element, sub-element, rollup, or other COA table record has been successfully used in order to advise the user before allowing deletion of the COA code. This batch process is only automatically launched if the “COA Deletion Prevention” APPCTRL parameter is set to a value that enables the COA Deletion Prevention features. When COA Deletion Prevention is enabled and an attempt is made to delete a COA record, the deletion is prevented until the batch job is launched and completes its research.

The COA Usage Verification batch process searches for the values of the COA's primary key(s) in, up to, five tables that are considered to authoritatively indicate that a COA has been “successfully used” if the COA is found. Each execution of the COA Usage Verification batch process can research the usage of exactly one COA record. The batch process stops searching for the COA’s usage once a first occurrence is found. The usage status is written to the batch log, and recorded in the COA Usage Verification temporary table (COA_USG_VER_TMP).

The COA Usage Verification temporary table contains one record per execution of the COA Usage Verification batch process. Each record stores information about the researched COA and the results, such as the table name of the COA (e.g. “R_FUND”) the full primary key of the researched COA, a field indicating whether the COA was determined to be successfully used at least once, and the timestamp at which the usage was verified.

This information is used, by the COA Deletion Prevention feature, the next time an attempt is made to delete the same COA. That next deletion attempt may be allowed or prevented again depending on the system settings associated with the COA Deletion Prevention feature (for example, APPCTRL parameters “COA Deletion Prevention” and “COA Deletion Verification Minutes”).

“Successful usage” of a COA is determined by looking at the results of budget transactions and accounting-based transactions. For budget transactions, a COA is determined to be successfully used if it appears in the Budget Transaction Line Component (BG_DOC_LN) table or the Budget Journal (JRNL_BUD) table. For accounting-based transactions, the COA is determined to be successfully used if the COA appears in the Posting Line Catalog (PSTNG_LN_CAT), Accounting Journal (JRNL_ACTG), or the Cost Accounting Journal (JRNL_CA) tables.

Although this batch process is designed to be launched automatically in response to an attempt to delete a COA record, there are technical limitations that prevent running this job manually. The COA Deletion Prevention features will work the same way regardless of whether the job is invoked manually or in response to a COA deletion attempt.

Process Steps	Messages

1. Parameter Validation	- Validating Batch Parameters - Parameters are valid or invalid depending on the Validation. - Batch Parameter validation completed
2. Budget Transaction/Accounting Transaction Search	If the COA record was not found in any of the five tables: - The record with <primary key values> is not currently in use. Otherwise, the process will stop and report the first table in which it was found: - The record with <primary key values> is currently in use in the <table> table.

Major Input

- Budget Transaction Line Component (BG_DOC_LN)
- Budget Journal (JRNL_BUD)
- Posting Line Catalog (PSTNG_LN_CAT)
- Accounting Journal (JRNL_ACTG)
- Cost Accounting Journal (JRNL_CA)

Batch Parameters

Parameter	Description	Default Value
TBL_NM	Table Name Required. This must be the name of a COA table in the Advantage Financial database.	No Default
COA_KEY_ATTS	COA Key Attributes Required. This is a list of the primary key column names for the COA table separated by the caret (^) character. For example, the primary key columns of R_UNIT would be listed as "FY^DEPT_CD^UNIT_CD". The column names can be listed in any order as long as it matches the order of the corresponding primary key values in the COA Key Values parameter.	No Default
COA_KEY_VALS	COA Key Values Required. This is a list of the primary key values to match the primary key column names in the COA_KEY_ATTS parameter value. For example, for	No Default

	R_UNIT, if the Unit record to be verified has primary keys values where FY is 2010, Department is 001, and Unit is 015, the value of COA_KEY_VALS should be "2010^001^015" The key values must be in the same order as the corresponding column names listed in the COA Key Attributes parameter.	

Major Output

- COA Usage Verification (COA_USG_VER_TMP)

Job Return code

The following table shows the potential job return codes for the COA Usage Verification batch job:

Return Code	Condition
Successful (1)	The job ends successfully.
Warning (4)	This job does not issue this return code.
Non Fatal Error (8)	This job does not issue this return code.
Failed (12)	The job fails under the following conditions: One or more parameter(s) are invalid. • One or more required parameter(s) are not entered. • 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 Criteria

N/A

Selection Criteria

The process accepts information identifying one record from a specific COA table as specified by the Table Name, COA Key Attributes, and COA Key Values parameters. The research performed by this process can easily be duplicated with direct SQL queries if desired by a user or administrator with direct database access.

First, build a SQL WHERE clause for the primary keys and primary key values of the desired COA table. For example, a WHERE clause for R_UNIT might be:

FY=2010 AND DEPT_CD='001' AND UNIT_CD='015'

Then add the WHERE clause to this query for each of the five tables (please note that FY changes to "FY_DC" in the accounting tables):

```
SELECT COUNT(1) FROM BG_DOC_LN WHERE FY=2010 AND DEPT_CD='001' AND UNIT_CD='015'

SELECT COUNT(1) FROM JRNL_BUD WHERE FY=2010 AND DEPT_CD='001' AND UNIT_CD='015'

SELECT COUNT(1) FROM PSTNG_LN_CAT WHERE FY_DC=2010 AND DEPT_CD='001' AND UNIT_CD='015'

SELECT COUNT(1) FROM JRNL_ACTG WHERE FY_DC=2010 AND DEPT_CD='001' AND UNIT_CD='015'

SELECT COUNT(1) FROM JRNL_CA WHERE FY_DC=2010 AND DEPT_CD='001' AND UNIT_CD='015'
```

If the query result is greater than zero, then the COA is considered to be in use.

It is acceptable if the SQL fails because any COA attribute does not exist in the table, not all COAs apply in all tables. For example, BSA_CD does not exist in JRNL_BUD and BG_DOC_LN as Balance Sheet Account Code is not a budgeting COA element. Any attempt to reference BSA_CD in a SQL statement against JRNL_BUD and BG_DOC_LN will result in a SQL error. The batch process does not encounter such SQL errors because it automatically skips any tables that do not contain one of the COA keys.

Also, please note that BSA and SBSA are special cases because Offset BSA or Offset SBSA, respectively, should also be included in the query. For example:

```
SELECT COUNT(1) FROM PSTNG_LN_CAT WHERE FY_DC=2010 AND (BSA_CD='001' OR OBSA_CD='001')
```

Problem Resolution

If the job ends with a return code other than Successful after completing parameter validation, a new job should be scheduled after the job log has been reviewed (the failed job should not be restarted).

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)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A

Failed (12)	Required parameters are not entered. Sample Message: Parameter COA Key Values is required.	Schedule a new job after entering a valid value for the parameter.	N/A
	Entered parameters are not valid. Sample Message: Parameter Table Name value 'R_COA' is not a valid table in the Advantage Financial database.	Schedule a new job after entering a valid value for the parameter.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Budget Transaction/Accounting Transaction Search

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	The process concluded successfully.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Failed because of runtime exceptions for an unexpected	Failure reason needs to be investigated before scheduling a new job.	N/A

	situation.		
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

2.1.8 Cost Accounting Archiving

Chain Name	Cost Accounting Archiving
Recommended Frequency	On Demand
Single Instance Required	Yes
Can be restarted?	N/A
Reports generated	Cost Accounting Archiving Statistics Report Expanded Selection Criteria Report(s) Cost Accounting Archiving Facilitator Report

Overview

With the progression of time, the data built up in the cost accounting area of Advantage Financial becomes considerable. While such data is necessary for reporting and online inquiry after a period of time, the data is no longer needed in the online application. The Cost Accounting Archiving process archives and deletes data based on Major Program, Program, and Task Order selection from Advantage Financial only. The data still remains in infoAdvantage.

The following job steps comprise the Cost Accounting Archiving chain process:

- Archiving Preprocessor
- Archive Facilitator
- Post Archiving Process

The steps are all singleton jobs. This means that only one of their instances can be run at a time. The Archive Facilitator does have the ability to launch System Maintenance Utility jobs in parallel.

Before any archive of data, it is assumed that research has been done to see that all activity has completed and that there will be no further need to record any more activity to a COA code. The Budget Pre-Archive Report is just one reporting tool that can be used in this review as it will locate transactions that have not completed the accounting chain (for example, encumbrances).

Along with the *completed* data assumption is the assumption that the data has *aged* according to data retention rules for the online application. To review proper aging of data before archiving, the Archiving Preprocessor step has an Activity Mode parameter that can be used in conjunction with the Research Cost Accounting Journal, Research Budget Journal, Research Posting Line Catalog, and Research Budget Line Catalog parameters to search for usage on these four tables, since the Last Activity Date parameter when running the Cost Accounting Archive in Report Only mode. If there’s a record found on one of these tables, an indicator is marked on the Expanded Selection Criteria report for that particular Major Program, Program, or Task Order.

As a final check, the Archiving Preprocessor job step will check that each Major Program, Program, and Task Order selected for archiving has an Effective End Date less than or equal to the current application date and the Active flag is unchecked. Such updates should be part of the normal business process of slowing down and eventually ceasing activity against a COA code.

The archive can run in one of three selection modes based on selection criteria specified in an input file:

- Task Orders Only
- Programs Only
- Major Programs along with the Programs and Task Orders associated

Running the Cost Accounting Archiving chain in *Report Only* mode is a useful tool to find out what Major Programs, Programs, or Task Orders are being identified by the selection criteria and how many records in the expanded selection will be archived and deleted by the run.

** Please note that if the archive is run in *Full* mode, the Application Parameter (APPCTRL) record for COA Deletion Prevention must be set to a value of *3* in order to delete records. **

This chain implements the Data Warehouse Archived Record Queue process, which is used to retain the table names and key values for records that are archived and deleted. The infoAdvantage reporting system uses this information to distinguish between records that have been archived (as valid historical data) before deletion and records that have simply been deleted from the system (as no longer needed). The process is enabled for most archiving processes when the value of Application Parameter Enable Data Warehouse Archived Record Queue (ENABLE_DW_ARCH_QUEUE) is *True*. Please see the *CGI Advantage System Administration Guide* for more information regarding this process and the Application Parameter.

Performance Recommendations for Research Activity Mode

In order to assist performance of Research Activity Mode, additional indexes can be added to the Budget Journal (JRNL_BUD), Cost Accounting Journal (JRNL_CA), Posting Line (PSTNG_LN_CAT), or Budget Transaction (BG_DOC_LN) Line tables depending on the type of research being performed for the Cost Accounting Archiving process. The indexes on these tables, particularly the Posting Line and Budget Journal, can impact performance times on high-volume daily processes that insert records into those tables. Since the Research Activity Mode of Cost Accounting Archiving may be used infrequently, the presence of these suggested indexes on these tables is only recommended temporarily while executing the archiving process and if performance execution time of the research is consistently high.

The three different indexes' SQL would look like the following for an Oracle DB:

For Major Programming research, consider:

```
CREATE INDEX N1BGDOCLN ON BG_DOC_LN (DOC_PHASE_CD, DEPT_CD,
MJR_PROG_CD)
CREATE INDEX N1JRNLBUD ON JRNL_BUD (DEPT_CD, MJR_PROG_CD)
CREATE INDEX N1JRNLCA ON JRNL_CA (DEPT_CD, MJR_PROG_CD)
CREATE INDEX N1PSTNGLNCAT ON PSTNG_LN_CAT (DOC_PHASE_CD, DEPT_CD,
MJR_PROG_CD)
```

For Task Order research, consider:

```
CREATE INDEX N2BGDOCLN ON BG_DOC_LN (DOC_PHASE_CD, DEPT_CD,
TASK_ORD_CD)
CREATE INDEX N2JRNLBUD ON JRNL_BUD (DEPT_CD, TASK_ORD_CD)
CREATE INDEX N2JRNLCA ON JRNL_CA (DEPT_CD, TASK_ORD_CD)
CREATE INDEX N2PSTNGLNCAT ON PSTNG_LN_CAT (DOC_PHASE_CD, DEPT_CD,
TASK_ORD_CD)
```

For Program research, consider:

```
CREATE INDEX N3BGDOCLN ON BG_DOC_LN (DOC_PHASE_CD, DEPT_CD,
PROG_CD)
CREATE INDEX N3JRNLBUD ON JRNL_BUD (DEPT_CD, PROG_CD)
CREATE INDEX N3JRNLCA ON JRNL_CA (DEPT_CD, PROG_CD)
CREATE INDEX N4PSTNGLNCAT ON PSTNG_LN_CAT (DOC_PHASE_CD, DEPT_CD,
PROG_CD)
```

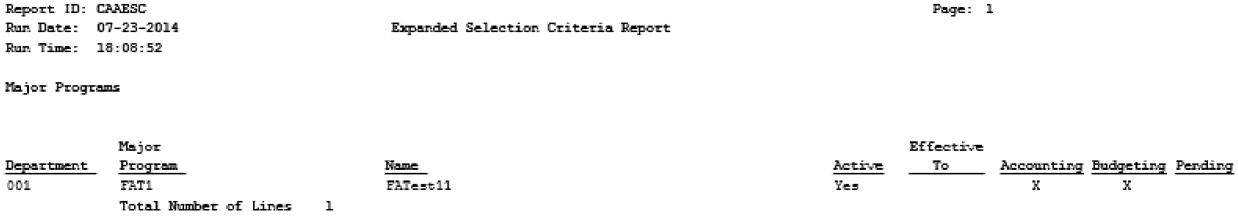
Details

One group of tables is archived to either CSV or XML files using export functionality and then purged. Another group of tables is just purged. The second group consists of tables that just aided in system processing (for example, did not control or result from reimbursement activity). Not all tables that could be considered as having Cost Accounting data are in this archive. Transactions (for example, CH, CAS, CAM, BGPDR, and BGPHE) still require Transaction Archiving. Budget lines on structures such as 11, 37, 38, 39, 40, and 47 still require Budget Archiving. The Cost Accounting Journal and various ledgers still require Journal/Ledger Archiving. Other tables will contain data that will have to be addressed manually if not addressed through the 'close out' process and procedures surrounding cost accounting entities. (A sample list of these is found at the end of the chain description in a Manual Intervention section.) The final type of cost accounting data is data that will not be archived because it is higher than Major Program such as Major Program Rollups and Drawdown Group records.

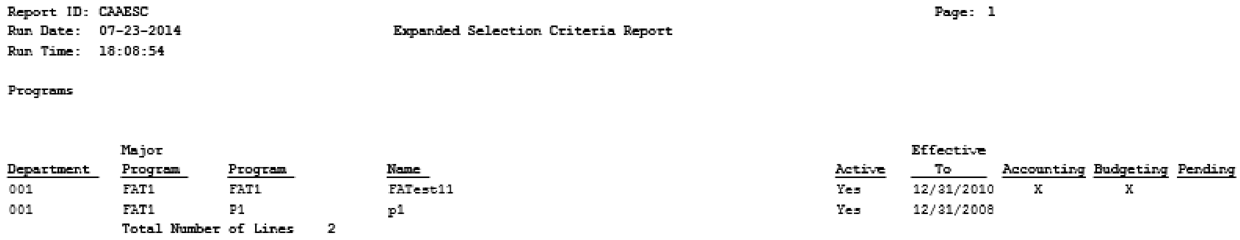
As with all jobs, parameters are validated first. Once that validation is successful, the first step in selection is to expand out any wildcards in the input file. The second step is to gather the list of Task Orders, the list of Programs, or the list of Major Programs. In the case of Major Programs, then the third step is to select Programs and Task Orders associated to one of the selected Major Programs. It would be very unlikely just to archive a Major Program that didn't have at least one Program associated with it. Associated Task Orders would only be found if the Streamlined Task Order Billing feature of Cost Accounting is used.

The selected Task Orders, Programs, and Major Programs are listed on a separate Expansion Listing report for each. Besides the full name being given for informational purposes, there are columns for Active and Effective To. Data in these latter two are populated for review given that a code cannot be archived if still active, has a blank Effective To date, or has an Effective To date after the current Application Date (see Note 1 earlier). In fact, the job will not simply skip such a record that is still 'active' and process the rest. The job will stop after selection if any code is still 'active'.

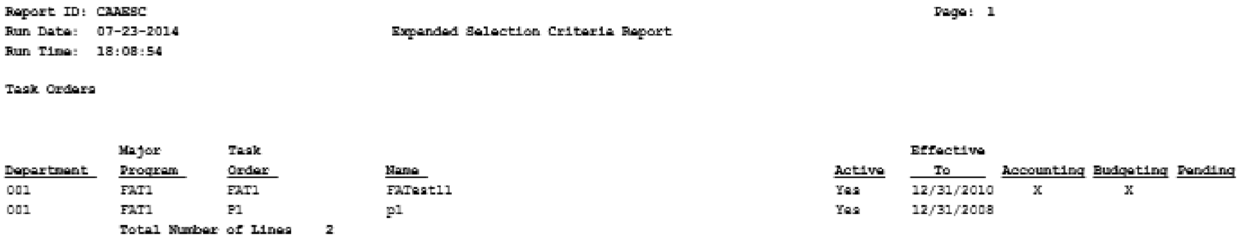
The Accounting column is marked for a specific Expanded Selection Criteria Report, if there is a matching record found on JRNL_CA that has a Run Time (RUN_TMDT) on or after the Last Activity Date parameter. If the Budgeting column is marked, there is a matching record found on the JRNL_BUD that has a Date/Time (DOC_LAST_DT) on or after the Last Activity Date parameter. If the Pending column is marked, there is a matching record found on either PSTNG_LN_CAT or BG_DOC_LN that has a Transaction Phase Code (DOC_PHASE_CD) of Pending and has a Run Time/Date (RUN_TMDT)/Transaction Last Date (DOC_LAST_DT), respectively, on or after the Last Activity Date parameter. Please note that the Accounting, Budgeting, and Pending columns are optional and will not be shown if their related research is not being performed.



```
Report ID: CAAESC                                                                              Page: 1
Run Date:  07-23-2014                    Expanded Selection Criteria Report
Run Time:  18:08:52

Major Programs


           Major                                                                   Effective
Department Program               Name                                   Active       To      Accounting Budgeting Pending
001        FAT1                  FATest11                               Yes                       X         X
           Total Number of Lines    1
```



```
Report ID: CAAESC                                                                     Page: 1
Run Date:  07-23-2014                      Expanded Selection Criteria Report
Run Time:  18:08:54

Programs


             Major                                                                      Effective
Department   Program    Program          Name                                   Active     To      Accounting Budgeting Pending
001          FAT1       FAT1             FATest11                               Yes     12/31/2010     X          X
001          FAT1       P1               p1                                     Yes     12/31/2008
             Total Number of Lines    2
```

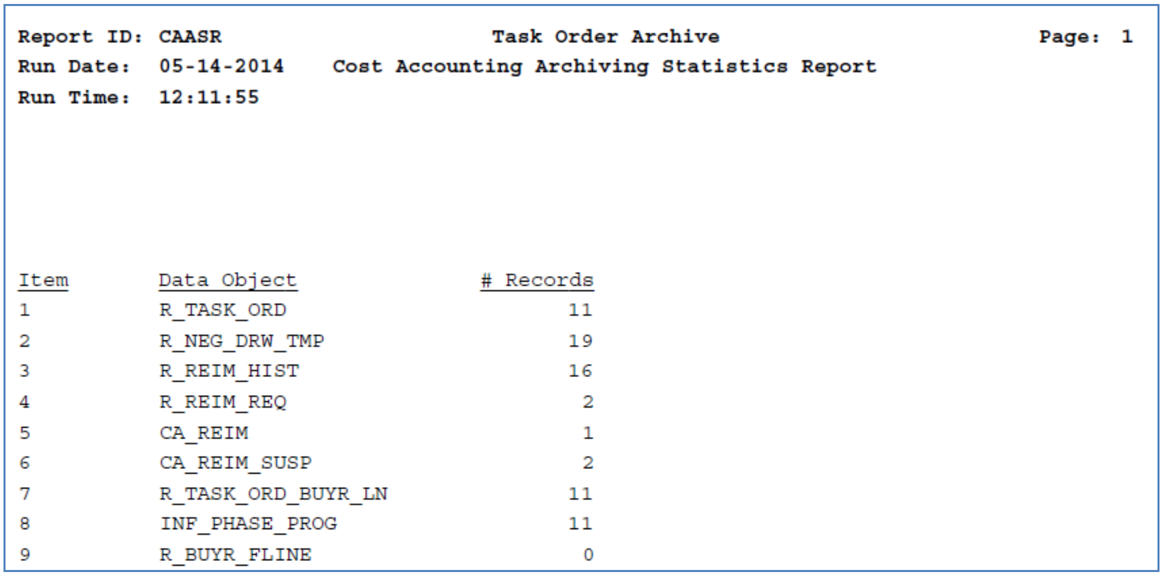


```
Report ID: CAAESC                                                                     Page: 1
Run Date:  07-23-2014                      Expanded Selection Criteria Report
Run Time:  18:08:54

Task Orders


             Major      Task                                                            Effective
Department   Program    Order            Name                                   Active     To      Accounting Budgeting Pending
001          FAT1       FAT1             FATest11                               Yes     12/31/2010     X          X
001          FAT1       P1               p1                                     Yes     12/31/2008
             Total Number of Lines    2
```

With one or more of the primary selection listings, records are identified on the tables defined in another parameter file. Once the final set of records is selected, facilitator records are submitted to the Job Manager, where the records from certain cost accounting tables are exported (archived) using parallel processing. Once the export process has successfully completed, the system will purge records for the archived and the non-archived tables.

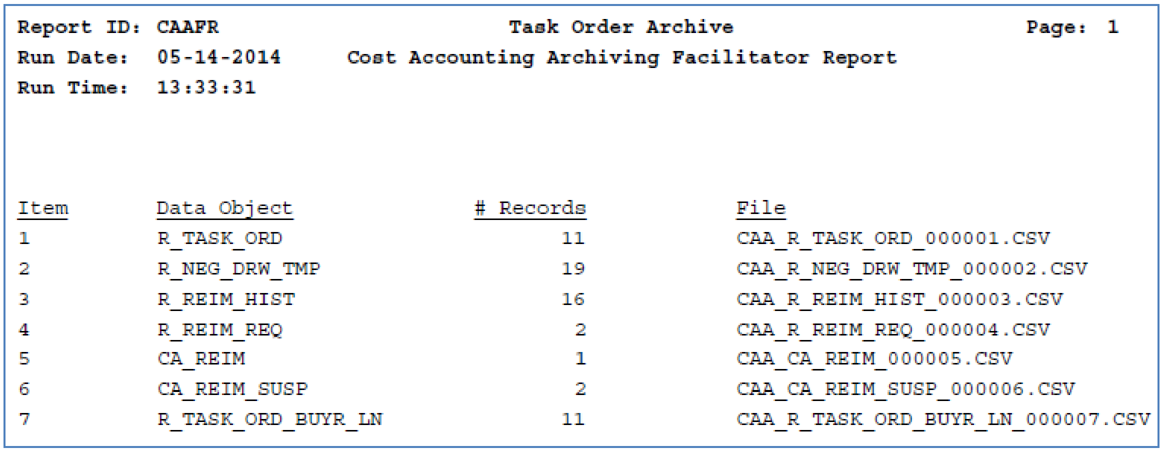
The output from the process includes three reports:

1. Cost Accounting Archiving Statistics Report showing a count of selected records per table. The sample below is abbreviated as there are many more items.



```
Report ID:  CAASR                   Task Order Archive                    Page:  1
Run Date:   05-14-2014    Cost Accounting Archiving Statistics Report
Run Time:   12:11:55






Item        Data Object                # Records
1           R_TASK_ORD                        11
2           R_NEG_DRW_TMP                     19
3           R_REIM_HIST                       16
4           R_REIM_REQ                         2
5           CA_REIM                            1
6           CA_REIM_SUSP                       2
7           R_TASK_ORD_BUYR_LN                11
8           INF_PHASE_PROG                    11
9           R_BUYR_FLINE                       0
```

2. Expanded Selection Criteria Reports (discussed earlier).
3. Cost Accounting Archive Facilitator Report listing each archive file created and the count of records per file. When run in *Report Only* mode, this report is not created as the chain ends after the Archiving Preprocessor completes the Statistics report and the Expanded Selection Criteria Report(s).



```
Report ID: CAAFR                          Task Order Archive                        Page:  1
Run Date:  05-14-2014      Cost Accounting Archiving Facilitator Report
Run Time:  13:33:31



Item        Data Object               # Records           File
1           R_TASK_ORD                       11           CAA_R_TASK_ORD_000001.CSV
2           R_NEG_DRW_TMP                    19           CAA_R_NEG_DRW_TMP_000002.CSV
3           R_REIM_HIST                      16           CAA_R_REIM_HIST_000003.CSV
4           R_REIM_REQ                        2           CAA_R_REIM_REQ_000004.CSV
5           CA_REIM                           1           CAA_CA_REIM_000005.CSV
6           CA_REIM_SUSP                      2           CAA_CA_REIM_SUSP_000006.CSV
7           R_TASK_ORD_BUYR_LN               11           CAA_R_TASK_ORD_BUYR_LN_000007.CSV
```

Chain Job Return Code

The acceptable job return codes (configured in the Configure Chain Job section of the Job Setup in CGI Advantage) for the jobs in the Cost Accounting Archive chain are delivered to be set to *Successful*. As with all CGI Advantage chain jobs, these acceptable return codes are configurable and may be changed to meet certain requirements.

The following table shows the potential return codes for the Cost Accounting Archiving chain. Note that the chain job will end with the highest return code across all of the jobs.

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

Problem Resolution

Please refer to the individual job “Problem Resolution” section for more details.

Advanced Selection Guidelines

The parameter file listing all the tables which are part of the archive can be modified within limits. Each table is listed with an export or a delete parameter line which can be changed, allowing you to retain some table data delivered as delete only or to simply delete data delivered to archive. Custom tables, such as a Funding Profile Inference, can be added in the same manner as those delivered in the file. The archive will not work with any table added that contains JRNL, LDGR, BUD_STRU, or DOC in the table name. The addition other tables to the file is possible, but something that should receive close attention and testing before attempting in a production environment.

Manual Intervention

The following list of pages should be addressed as a Cost Accounting entity is closing or is being officially closed. However, if not, data on these pages could contain Major Program, Program, or Task Order codes that would no longer be valid after an archive. This would result in New Year Table Initialization failing to roll forward a record, subsequent updates to a record to fail, and copying an existing record to create a new one would result in the new record failing. The following list contains only the baseline pages and you may have other custom ones also.

- ACTPL – Accounting Template
- REQBUD – Required Budget
- ACTINF – Activity Inference
- LPPA – Parameters for Lapse Process
- RLPA – Parameters for Roll Process
- CLMACT – Claim Account
- CMIAINT – CMIA Interest Parameters
- COAX – Chart of Account Crosswalk
- ADCP – Automatic Transaction Correction Parameters
- DISBMR – Disbursement Hold Exclusion
- FAIE – Fixed Asset Intention Exception
- FARO – Fixed Asset ReOrg Parameters
- FNCHG – Finance Charges Setup
- INTEA – Intercept External Allocation
- INTF – Intercept Fee Setup
- IVAD – Internal Vendor Accounting
- INTACT – Interest Accounting Line Crosswalk
- MATIN – Material Testing Class Inference
- REIMOTPT – Reimbursement Output Parameters
- PBDIST – Pool Base Distribution
- WHSE - Warehouse

Research

The following list of pages and system areas should be reviewed to see, if there is any ‘open’ cost accounting activity:

- Cost Accounting Budgets
- Reimbursement Budgets
- Fixed Assets still depreciating
- Fixed Asset Intent records not completed
- Fixed Asset Program Asset Generation records not completed
- Disbursement Requests
- Active Debt Instruments
- Pending Transactions
- Reimbursements In Progress that can complete (R_REIM_REQ, R_REIM_HOLD, CA_REIM)

Post Archive

Most find that moving the files created from the archive out of the Advantage file system gain the most in overall system file size. However, this is not a requirement but common place as there is not an automatic restore job for this data that would look to the initial location for the files.

Cost Accounting Archiving Chain: Archiving Preprocessor Job

Job Name	Archiving Preprocessor
Recommended Frequency	On Demand This job must be run as part of the Cost Accounting Archiving Chain.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	• Cost Accounting Archiving Statistics Report • Expanded Selection Criteria Report(s)

Overview

This step first validates parameters. If parameters are valid, it then identifies the rows to be archived based on the selection criteria and divides the selected rows in each table to be archived into groups for each export file. For example, if R_TASK_ORD has 200 rows and the Job Block Size parameter is 100, then two export files will be created.

Using this example, the 200 rows selected from R_TASK_ORD are expanded using the Instructional file, which contains table listings. For each of those table listings they will have between one and three selection keys. The three possible selection keys are Major Program Key Selection (MP_KEY_SELECTION), Program Key Selection (PG_KEY_SELECTION), and Task Order Key Selection (TO_KEY_SELECTION). In this case, it is a Task Order Key Selection, since it is selecting from the Task Order table, so all tables that have the TO_KEY_SELECTION will be included.

If selection is based on Major Program, then the initial selection would be using MP_KEY_SELECTION. After the Major Program expansion is complete, it will perform the same expansion for Program using the PG_KEY_SELECTION and lastly Task Order using TO_KEY_SELECTION.

The Cost Accounting Archiving Statistics Report lists the total number of records eligible for archival or deletion from each table. The Expanded Selection Criteria Report for Major Programs, Programs, and Task Orders show the specific records that will be archived for these tables based on the selection criteria and optionally shows if there has been an activity on the Costing Accounting Journal, Budget Journal, Posting Line Catalog, or Budget Line Catalog for those tables. If the run mode is set to *Report Only*, the process will end at this point.

In *Full* run mode, a System Maintenance Utility parameter file is created for each export file to be created. The parameter file contains the table name, the identification of specific records, and other settings. A Facilitator table record referencing the SMU parameter file is created and saved. Facilitator records can be viewed in the Archive Facilitator Inquiry page (QARC) for the status. The file names for the archived records are written to the chain parameter file for use by the Post Archiving Process job of the chain.

Process Steps	Messages
1. Parameter Validation	• Run Started *Each parameter is listed* • Validating Batch Parameters *If any is found to be invalid an error will also be issued* • Batch Parameters are valid. *or* • Batch Parameters are not valid.
2. Pre-processing	The following messages will be issued when the job runs in Full mode. • Record selection started. • Rendering Selection report started. • Reports output folder mapped (*followed by the HTML & PDF locations)* • Rendering Expanded Selection Criteria Report completed. • *Each table is listed with the number of records identified.* • Record selection completed. • Rendering Statistics report started. • Reports output folder mapped (*followed by the HTML & PDF locations)*. • Rendering Statistics report completed. • Writing chain parameter file. • Chain parameter file (Chain Parameter File Name) written in directory (CGI Parameter Directory). • Run Ended. The following messages will be issued when the job runs in Report-Only mode. • Record selection started. • Rendering Selection report started. • Reports output folder mapped (*followed by the HTML & PDF locations)* • Rendering Expanded Selection Criteria Report completed. • *Each table is listed with the number of records identified.* • Record selection completed. • Rendering Statistics report started. • Reports output folder mapped (*followed by the HTML & PDF locations)*. • Rendering Statistics report completed. • Number of records deleted from temporary table: (number of records) • Run Ended.

Major Input

- Major Program (R_MJR_PROG)
- Program (R_PROG)
- Task Order (R_TASK_ORD)
- Instruction File Table List (Tables are based on contents in the file)
- JRNL_CA, if research is enabled for the Cost Accounting Journal
- JRNL_BUD, if research is enabled for the Budget Journal
- PSTNG_LN_CAT, if research is enabled for the Posting Line Catalog
- BG_DOC_LN, if research is enabled for the Budget Line Catalog

Batch Parameters

Parameter	Description	Default Value
ACT_MODE	Activity Mode (1 - No Research, 2 - Research)	1
AMSEXPORT	Export Location at Archiving Preprocessor Job. Optional. If the default value is changed, it must be a valid directory and will be used to determine where the export files will be written. (The size of archived files may be so great that a location other than the default Export/Import directory should be used.)	$$AMSROOT$$/ ExportImport
AMSPARM	Parameter Location at Archiving Preprocessor Job. Required. This must be a valid directory containing the file designated by the Parameter File parameter.	$$AMSROOT$$/ Parms
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) Required, not overrideable. This file is created during the run to convey parameters to the subsequent steps in the chain.	CAArchParams.txt
CLIENT_NM	Client Name for reports. Optional.	No Default
COMMIT_BLOCK_SIZE	Commit Block Size. Required. This value will determine how often database transactions will be committed and can be used for performance tuning. The value must	1000

	be a positive integer.	
EXP_FILE_TYP	Export File Type. Required. Enter '1' for XML, '2' for CSV.	2
FILE_PREFIX	File Prefix. Required. Prefix used for archive file names. Job will use CAArch if left blank.	CAArch
JOB_BLOCK_SIZE	Job Block Size. Required. The maximum number of lines in an output file. If left blank the process will assume 10000.	10000
LAST_ACT_DT	Last Activity Date. Date from which usage will be evaluated. Enter in mm/dd/yyyy format.	No Default
PARM_FILE	Input Selection File. Required. Enter the name of the parameter file for cost accounting archiving; this file must exist in the directory specified by the Parameter Location at Archiving Preprocessor Job parameter.	CAArchiveSelect.txt
RSRCH_CA_JRNL	Research Cost Accounting Journal (Y - Research, N - Do not research)	Y
RSRCH_JBUD	Research Budget Journal (Y - Research, N - Do not research)	Y
RSRCH_PSTNG_CAT	Research Posting Line Catalog (Y - Research, N - Do not research)	Y
RSRCH_BUD_CAT	Research Budget Line Catalog (Y - Research, N - Do not research)	Y
RUN_MODE	Run Mode. Required. Enter '1' for Report Only, '2' for Full. If left blank then 1 will default.	1
SMU_FILE_PREFIX	SMU Facilitator Parameter File Prefix. Required, not editable. This value will be used to determine a SysManUtil input parameter file name for each Facilitator job.	CAArch
TBL_PARM_FILE	Instructional file with table listing (txt). Required, not editable. This file will be used to identify tables for selection expansion.	CAArchListParams.txt

Major Output

The following are output in both the *Report Only* and *Full* modes:

- Cost Accounting Archiving Statistics Report
- Expanded Selection Criteria Report(s)
- Cost Accounting Archiving Temporary (CA_ARCH_TBL) table which contains a record for each table and a record for each selected record in that table
- Research results from the Cost Accounting Journal, Budget Journal, Posting Line Catalog, and Budget Line Catalog, if conducted.

The following are only output in *Full* mode:

- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page.
- CAArch##_#.txt – File created to be passed to SMU to perform the archive. The ## is the chain job ID and the # is an incrementing number for each table archived.
- ARCH_DW_QUEUE_REC – The Data Warehouse Archive Record table keeps track of the values for all primary keys of each record being archived and deleted. The table name and attribute names are stored in ARCH_DW_QUEUE_TBL and linked to this record by unique ID.
- ARCH_DW_QUEUE_TBL – The Data Warehouse Archive table keeps track of the table name and primary key attribute names of each table being archived. If a record already exists for the table being archived, it will reuse the record. If it does not yet exist, it will be created.

Minor Output

These are only output in *Full* mode:

- CAArchParams.txt – To pass information from the first job in the chain to later jobs.
- Folder in ExportImport / CostActgArchive

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	This return code is issued under the following condition: - The Run Mode parameter is set to '1' (*Report Only*).
Non Fatal Error (8)	This return code is issued under the following condition: - No eligible records found for the selection file. - Run mode is *Report Only* and 1 or more codes failed validations for archive.
Failed (12)	The job fails under the following conditions:

Return Code	Condition
	One or more parameter(s) are invalid. - One or more required parameter(s) are not entered. - The Input Selection File is not found in the directory specified by the Parameter Location at Archiving Preprocessor Job parameter. - The Instructional File is not found in the directory specified by the Parameter Location at Archiving Preprocessor Job parameter. - Run time exceptions for unexpected situations. - Run mode is *Full* and 1 or more codes failed validations for archive. When this job ends with a return of code Failed, subsequent jobs in the chain are set to Inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain are set to Inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return of code System Failure, subsequent jobs in the chain are set to Inactive.

Sort Criteria

None

Selection Criteria

The Input Selection File parameter identifies a text file that is required to be present in the parameter directory (PARMS). At least one set of selection criteria must be specified.

Each set of selection criteria must start with the _PARAM_LINE_ separator. Selection criteria values in the input file may use commas to specify more than one value per COA element (but only one comma-separated list of values is allowed per PARAM LINE section). The "_" (underscore) may be used as a wildcard to match any one character. The "%" (percent sign) may be used as a wildcard to match 0 to *n* characters.

While the parameter file does allow for complex selection criteria, the easiest method to build a parameter file (and avoid parameter file edits for what is not allowed) is to run a series of database queries to get a list of key values for the Major Program, Program, or Task Order desired to be archived. Each Cost Accounting line is uniquely defined by their key value and it is straightforward to put into the file under one parameter line.

The layout of this file is identical to that used by the Budget Archive, so the selection of COA codes to archive can be re-used to create both files.

Input File Example:

```
_PARAM_LINE_
DEPT_CD=010
MJR_PROG_CD=MJR1,MJR2,MJR3
_PARAM_LINE_
DEPT_CD=020
MJR_PROG_CD=MP1
_PARAM_LINE_
DEPT_CD=001
MJR_PROG_CD=PGM%
```

Within each _PARAM_LINE_ group, the selection criteria are grouped as AND statements. Then, the _PARAM_LINE_ groups are grouped as OR statements to form the full selection criteria. The above example would be interpreted as:

> (Department = ‘010’ **and** (Major Program is ‘MJR1’ **or** ‘MJR2’ **or** ‘MJR3’))
>
> **or** (Department = ‘020’ **and** Major Program = ‘MP1’)
>
> **or** (Department is ‘001’ **and** Major Program matches ‘PGM%’)

It is acceptable to have a given record satisfy the criteria for multiple _PARAM_LINE_ groups. This may occur depending on how broad or specific the selection criteria are defined for each group. When this situation occurs, the Archiving Preprocessor will properly select the record just once.

Input File Rules:

1. The input file must contain at least one PARAM LINE section with two key fields for selection and at least one value. DEPT_CD will always be one of the key fields.
2. The input file can only include a set of Major Programs, Programs, or Task Orders. More than one set type is not allowed.
3. The PARAM LINE section will be controlled several ways to prevent performance degradation but allow for some flexibility in selection when a COA list is specified:
    a. Wildcard and comma features cannot be combined into the same line, so the following is not allowed:

    ```
    _PARAM_LINE_
    DEPT_CD=010,02%

    MJR_PROG=MP1
    ```

    b. Multiple lines may use the wildcard feature, so the following is allowed:

    ```
    _PARAM_LINE_
    DEPT_CD=01_
    MJR_PROG=100%
    ```

    c. Multiple wildcards are allowed in a single line, so the following is allowed:

    ```
    _PARAM_LINE_
    DEPT_CD=010
    MJR_PROG=%100%
    ```

    d. Only one line can use the comma feature, so the following is not allowed:

    ```
    _PARAM_LINE_
    DEPT_CD=010,020
    MJR_PROG=10011, 10012
    ```

    Instead, the same can be achieved with multiple lines, for example:

```
_PARAM_LINE_
DEPT_CD=010
MJR_PROG=10011,10012
_PARAM_LINE_
DEPT_CD=020
MJR_PROG=10011,10012
```

The Instructional file parameter identifies a text file that is required to be present in the parameter directory.

Each set of selection criteria must start with the _PARAM_LINE_ separator. Selection Key criteria values in the input file may use commas to specify more than one column.

Input File Example:

```
_PARAM_LINE_
TABLE_NAME= R_REIM_HIST
MP_KEY_SELECTION=DEPT_CD, MJR_PROG_CD
PG_KEY_SELECTION=DEPT_CD, PROG_CD
TO_KEY_SELECTION=DEPT_CD, TASK_ORD_CD
**ACTN_CD=TBLEXPORT
```

Within each _PARAM_LINE_ group, depending on the type of archiving taking place, that particular key is used. For example, if a Major Program archive is taking place, the MP_KEY_SELECTION will be used during the first round of selection expansion. If the Major Program record had DEPT_CD ‘010’ and MJR_PROG_CD ‘MP1’, it would then perform a search on R_REIM_HIST for records matching those values on that table. Any records found will execute the action indicated, which in the example above is TBLEXPORT.

As the Major Program run continues, the 2nd expansion would be for Program records for that Major Program and that would use the PG_KEY_SELECTION criteria. Any records captured during the 1st expansion would not be included again. During the 3rd expansion, which would be for Task Orders for that Major Program, the process would use the TO_KEY_SELECTION criteria. Any records captured during the 1st and 2nd expansions would not be included again.

Input File Rules:

1. The column values provided must exist on both the table specified and the selection key table it represents.
2. At least on key selection value must be provided for any table indicated.
3. The table name must be provided for any key selections indicated.
4. The **ACTN_CD must be either TBLEXPORT or TBLDELETE and must be present.
5. The tables for Major Program, Program, and Task Order do not have to be present in the file as the archive is already aware of these.

Problem Resolution

If the job ends with a return code other than Successful or Warning after completing parameter validation, a new job should be scheduled after the job log has been reviewed (the failed job should not be restarted).

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)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. **Sample Message:** Parameter Run Mode value must be ‘1’ or ‘2’.	Schedule a new job after entering a valid value for the parameter.	N/A
	Entered parameters are not valid. **Sample Message:** Parameter Job Block Size value 'ABC' is invalid. It must be a positive integer.	Schedule a new job after entering a valid value for the parameter.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Pre-processing

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	The Run Mode parameter is set to '1' (*Report Only*). This is a normal condition that	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
	sets any subsequent jobs to inactive.		
Non Fatal Error (8)	No tables need to be archived because no records match selection criteria.	Confirm selection criteria before scheduling a new job.	N/A
	Run mode is Report Only and 1 or more codes failed validations for archive.	Take Expansion Report and update Major Program, Program, or Task Order codes as necessary.	Refine selection criteria to not select codes that are not ready to archive.
Failed (12)	Failed during attempt to write the chain parameter file.	Verify that the file system is not full, has the correct access, and so forth, before scheduling a new job.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Cost Accounting Archiving Chain: Archive Facilitator Job

Job Name	Archive Facilitator
Recommended Frequency	On Demand This job must be run as part of the Cost Accounting Archiving Chain.
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	None

Overview

The Facilitator job spawns multiple System Maintenance Utility (SMU) jobs to export (archive) selected Cost Accounting records to files. This step does **not** delete records from the Cost Accounting tables. The Archive Facilitator step retrieves the records from the Facilitator table for the chain's Job ID. It then reads the file names from these records and spawns multiple SMU jobs, using the parameter file names from the Facilitator table as input for the SMU jobs. The parameter files specify the "Table Export" command with other options set in the Archive Preprocess step. Each SMU job is responsible for exporting (archiving) table records from a single table. Multiple SMU jobs may be spawned to export data from a single table (depending on the Archiving Preprocessor Job Block Size parameter value), but one SMU job will never be responsible for exporting from more than one table.

Process Steps	Messages
1. Parameter Validation	• Run Started *Each parameter is listed*
2. Facilitator Job Processing	• The Run Number for this archive/restore process = xxx (*xxx being the chain's Job ID. Restore is not a feature of Cost Accounting Archive.*) • For each job in the Facilitator for the Run Number: • SMU Job - xxx - Spawned (*xxx being the Job ID)* • Each spawned job will have either of these results: • SMU Job - xxx – Processing completed successfully (*xxx being the Job ID)* • SMU Job - xxx - Failed *(xxx being the Job ID)* • The job slept a total of xxx times, for a Total Sleep Time of yyy seconds. *(xxx and yyy being the counts)* • Run Ended

Restartability Information

This job can be restarted if it fails due to any reason. After restart, the job will continue to process the records based on the status of each Facilitator record. Restarting this job step will restart all spawned jobs automatically.

Major Input

- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page.
- BS_AGENT – The job will consult the Job Manager for the status of jobs that have been spawned.
- Cost Accounting related tables – Each SMU job spawned by the Facilitator job will export a subset of records, from a single table, depending on number of starting and ending keys specified in the parameter file supplied to the Facilitator record.

Batch Parameters

Parameter	Description	Default Value
ARCHIVE_RESTORE_ID	Archive Restore ID Required, not overrideable. This value tells the Facilitator to select the archive table process for Cost Accounting tables.	12 (Cost Accounting Archive)
COMMIT_BLOCK_SIZE	Commit Block Size Required. This value will determine how often database transactions will be committed and can be used for performance tuning. This must be a positive integer.	1000
PROCESSOR_NO	Number of jobs for Facilitator to keep running Required. This value sets the number of jobs to launch and keep submitted/running. Exceeding the number of available VLS is acceptable. This must be a positive integer.	1
SLEEP_TIME	Number of seconds to wait between polling occurrences Required. This value sets the length of time between iterations, which checks the status of the jobs that are running, and the jobs that are launching for unprocessed Facilitator records as necessary.	5
SMU_CTLG_ID	SMU Catalog ID Required, not overrideable. This value tells the Facilitator to create a specific SMU job.	3
UPDATE_STATUS	Update Status Required, not overrideable. This value tells the Facilitator to update the status of each Facilitator record based on the status of its corresponding job in the Job Manager.	Y

Major Output

- Exported CSV or XML files
- FACILITATOR – The Facilitator table is updated to reflect the Job Manager status.

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	This return code is issued under the following conditions: No Facilitator records found for Run Number = xxx. Nothing to process. (xxx being the chain's Job ID) SMU Job Parameter File NOT Found – (file name) Record skipped.
Non Fatal Error (8)	This step does not issue this return code.
Failed (12)	The job fails under the following conditions: One or more parameter(s) are invalid. One or more required parameter(s) are not entered. Parameters are invalid. Run time exceptions for unexpected situations. When this job ends with a return of code Failed, subsequent jobs in the chain are set to Inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain are set to Inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return of code System Failure, subsequent jobs in the chain are set to Inactive.

Sort Criteria

None

Selection Criteria

FACILITATOR records contain the location of input parameter files for use by spawned System Maintenance Utility batch jobs.

Problem Resolution

No database restore is required. Both the Archive Facilitator chain step and each spawned System Maintenance Utility job can be restarted.

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)	All of the parameters are	N/A	N/A

Possible Return Codes	**Condition**	**Recommendation**	**Other Instructions**
	validated successfully.		
Warning (4)	No Facilitator records found for Run Number (Nothing to process).	This is acceptable in the event that there are no records to process. A new chain can be started with only the Post Archiving Process enabled. Otherwise, the reason for the missing records must be investigated before a new chain job is run.	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. **Sample Message:** SLEEP_TIME is required to run the Facilitator	Schedule a new job after entering a valid value for the parameter.	N/A
	Entered parameters are not valid. **Sample Message:** The SLEEP_TIME must be an integer and > 0	Schedule a new job after entering a valid value for the parameter.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Facilitator Job Processing

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	SMU Job Parameter File not found.	If all files are missing, the reason must be investigated before a new chain job is run. If the files can be located and moved to the expected location, the job can be restarted.	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	This return code will be issued when any spawned SMU job fails. **Sample Message:** SMU Job – xxx – Failed	Determine the cause for the spawned SMU job to fail and resolve. The spawned job can be restarted independently if desired, and restarting the chain job will also restart any failed SMU jobs.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

<u>Cost Accounting Archiving Chain: Post Archiving Process Job</u>

Job Name	Post Archiving Process
Recommended Frequency	On Demand This job must be run as part of the Cost Accounting Archiving Chain.
Single Instance Required	Yes
Can be restarted?	No

Reports generated	Cost Accounting Archive Facilitator Report

Overview

The parameter directory and the chain parameter file are the only two parameters for this step. All of the user parameters that this step needs are stored in the designated file by the Preprocessor. The first action of this step of the chain is to open the chain parameter file and read these parameters. The second action is to confirm that all facilitator records were processed successfully in step 2 of the chain job. If any facilitator job did not finish successfully, the process ends. The third action is to delete the selected records from the Cost Accounting related tables and to write the Cost Accounting Archive Facilitator Report.

Cost Accounting Archive Facilitator Report Sample

Process Steps	Messages
1. Parameter Validation	• Run Started *Each parameter is listed* • Validating Batch Parameters *If any is found to be invalid an error will also be issued* • Batch Parameters are valid.
2. Post-processing	• Confirming Facilitator status for Run ID (Run ID of the current Chain Job). • Facilitator completed (number) of (number) job(s) for Run ID (Run ID of the current Chain Job). • If the number of jobs completed is not the same as the total number of jobs: • Post Archive deletion cannot proceed until all Facilitator jobs for Run ID (Run ID of the current Chain Job) have been completed. (Process ends here in this case) • Facilitator status check completed. • Retrieving table information for archived and deleted tables. • Retrieving table information completed. • Deletion of selected records started. • At this point, a series of progression messages will appear. Sample messages: o Deleted 2 records from R_PROG. • Deletion of selected records completed. • Rendering Facilitator report started. • Reports output folder mapped (*followed by the HTML & PDF locations)* • Rendering Facilitator report completed. • Number of records deleted from temporary table: (number of records) • Run Ended

Restartability Information

This job cannot be restarted. If the job failed due to any reason, a new job can be scheduled after correcting errors that caused the job to fail – either individually, or as part of a new chain job by disabling the first two jobs in the chain and specifying the same set of parameters.

Major Input

- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page. This job will check the table to see if any records remain that have not completed successfully.

- BS_AGENT – The job will consult the Job Manager for the status of jobs that have been spawned.
- Chain Parameter File – The Preprocessor step writes this file to communicate all of the settings that this step needs.

Batch Parameters

Parameter	Description	Default Value
AMSPARM	Parameter Location at Archiving Preprocessor Job Required. This must be a valid directory containing the file designated by the Parameter File parameter.	$$AMSROOT$$/ Parms
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) Required, not overrideable. This file is created during the first job in the chain to convey parameters to this process.	CAArchParams.txt

Major Output

- Major Program (R_MJR_PROG) - Selected records are deleted from this table.
- Program (R_PROG) - Selected records are deleted from this table, depending on the run being performed. For example, a Task Order run would not delete from this table.
- Task Order (R_TASK_ORD) - Selected records are deleted from this table, depending on the run being performed. For example, a Program run would not delete from this table.
- Instruction File Table List - Selected records are deleted from these tables.
- Cost Accounting Archiving Temporary (CA_ARCH_TBL) table – all rows inserted to this table by the pre-processor step are deleted in this step.

Job Return Code

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

Return Code	Condition
Successful (1)	The SMU jobs were confirmed to have completed, and all selected records were deleted.
Warning (4)	This step does not issue this return code.
Non Fatal Error (8)	This step does not issue this return code.
Failed (12)	The job fails under the following conditions: One or more parameter(s) are invalid. - One or more required parameter(s) are not entered. - Parameters are invalid. - Run time exceptions for unexpected situations.

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

Sort Criteria

None

Selection Criteria

CA_ARCH_TBL records identify the tables and their individual records to be deleted. The temporary records are inserted by the Preprocessor step during the record selection process and are keyed by the Chain Job ID from the Job Manager. This step also removes all temporary records associated with that Chain Job ID after the archive deletions have been completed successfully.

Problem Resolution

Since the job cannot be restarted, if the job ends with any return code (Failed, Terminated or System Failure) a new chain job with the first two jobs in the chain disabled should be scheduled with the same parameters.

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)	The SMU jobs were confirmed to have completed, and all selected records were deleted.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. **Sample Message:**	Enter the correct directory for Parameter Location and	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
	Parameter AMSPARM directory is required and must be a valid directory.	schedule a new chain job with only step 3 enabled.	
	Entered parameters are not valid. **Sample Message:** Parameter Chain Parameter File is required to have file extension "txt".	Enter the correct file name for Chain Parameter File and schedule a new chain job with only step 3 enabled.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Post-processing

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new chain job with only step 3 enabled.	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

2.1.9 Data Conversion for Prior Bypass Functionalities

Chain or Job Name	Data Conversion for Prior Bypass Functionalities
Recommended Frequency	Single
Single Instance Required	This is one time conversion job for the Bypass Lower Level Approval functionality so there should not be any need for parallel instances.
Can be restarted?	No
Reports generated	N/A

Overview

As a part of the Bypass Lower Level Approval functionality, two new columns have been added in WF_APRV_SH so the purpose of this job is to insert the values in these two new columns for existing records.

The process which is being followed has been mentioned step by step.

1. First a database connection is created.
2. If the connection is successful then all records are fetched from WF_APRV_SH table where WF_APRV_SH.SEQ_NO = 1.
3. Records are fetched from WF_APRV_SH for each iteration of step 2. By applying some constraint or filter like WF_APRV_SH, DOC_CD, WF_APRV_SH .DOC_DEPT_CD, WF_APRV_SH .DOC_ID, WF_APRV_SH .DOC_VERS_NO.
4. FINAL_ASSIGNEE_ID and FINAL_ASSIGNEE_LEVEL are calculated and added in the update statement.
5. Update Batch is executed for every 500 records until all records are not updated for all rows from step 2.
6. Database Connection is closed.

Process Steps	Messages
1. Database Connection.	• Connecting to Database • Connected to Database
2. Selection of Records.	• Retrieving the Records from WF_APRV_SH Table. • Records has been retrieved
3. Update the records	• Updating the records • WF_APRV_SH.FINAL_ASSIGNEE_ID and WF_APRV_SH.FINAL_ASSIGNEE_LEVEL has been updated.

Major Input

- This job performs the operation on WF_APRV_SH table only.
- No need of batch parameter.

Major Output

- If the job completes successfully then it updates the WF_APRV_SH .FINAL_ASSIGNEE_LEVEL and WF_APRV_SH.FINAL_ASSIGNEE_ID column values for existing records that are displayed on the Worklist Pipeline page.
- Progress can be monitored throw the below tables.
  1. Log table : BS_AGENT_LOG
  2. Records Table :WF_APRV_SH

Job return codes have been mentioned in below table.

Return Code	Condition
Successful (1)	WF_APRV_SH .FINAL_ASSIGNEE_LEVEL and WF_APRV_SH.FINAL_ASSIGNEE_ID is updated Successfully.
Warning (4)	No eligible records found. This could be because of the following reasons: • No record exists in WF_APRV_SH table.
Failed (12)	The job will fail under the following conditions: • If there is any database connection issue. • Unable to find the record or table WF_APRV_SH. • Run time exceptions for unexpected situations. • Column FINAL_ASSIGNEE_LEVEL and FINAL_ASSIGNEE_ID does not exist in WF_APRV_SH.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a Return Code of Terminated subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a Return Code of System Failure, subsequent jobs in the chain are set to inactive.

Sort Criteria

N/A

Selection Criteria

All existing records in WF_APRV_SH table will be populated with <2> field values

Problem Resolution

This section discusses the problems and the resolutions at the job step level.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of 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 two conditions. • Encounters any runtime exceptions. • Failed during restart. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	If another instance of the job has already been scheduled and run successfully, then this job should not be restarted – only a new job should be scheduled.
	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.	
	If job failed due to Column FINAL_ASSIGNEE_LEVEL and FINAL_ASSIGNEE_ID	Make sure that Bypass Lower Level Approval functionalities are deployed properly.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and run successfully, then this job should not be restarted – only a new job should be

			scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and run successfully, then this job should not be restarted – only a new job should be scheduled.

2.1.10 Future Transaction Trigger

The Future Transaction Trigger (FDT) Process creates all transactions set to be future triggered based on their original transaction and the options set for future triggering.

When to Run

This process can be run on demand or included as a part of the nightly cycle. It is recommended that this job be run as part of the nightly cycle. After the Begin Day job increments the Application Date is the most likely location for this job.

Depending on what types of transactions are triggered, a site may want to run a System Maintenance Utility job to submit all or some of the triggered transactions. Transactions loaded from the Future Transaction Trigger job only create transactions in with a transaction **Phase** of *Draft*. If users do not need to intervene to set amounts and other data on the triggered transactions, then the System Maintenance Utility job will submit the transactions.

Description

The process involves the scheduling of a chain job, which can be accessed by clicking on **Utilities : Chain Jobs : Future Transaction Trigger** on the Batch Catalog.

The chain process consists of the following batch jobs:

- Job 1 – FDT Preprocessor
- Job 2 – FDT Facilitator
- Job 3 – FDT Exception Report

Job No	Job	Description	Output
1	FDT Preprocessor	Prepares transactions for future triggering and creates blocks of work.	• Parameters are validated, SMU parameter files are created and the Chain job parameter file is created.
2	FDT Facilitator	Spawns SMU jobs for each block of work.	Triggered transactions are created.
3	FDT Exception Report	Cleans FDT table of old triggers and generates exception report.	• The FDT table is cleaned of completed triggers; Exception Report is generated.

Parameters:

For the entire chain process a user has to enter parameters for the ‘Future Transaction Trigger’ job only. The process propagates required parameters down the chain jobs. Some of them are non-overrideable parameters, which are provided with the Job Setup and do not have to be specified by the user. These are meant for system use.

Job No	Job	Parameter	Description	Default Values	For System Use (non-overridable by user)
1	FDTPreprocess	TRIG_DATE	Trigger all transactions up to and including this date. If no date is specified, then the current application date is used.		No
1		CHAIN_PARM_FILE	Common Chain Parameter File. Must end in .txt.	FDTParams.txt	Yes
1		DOC_DEPT_CD	Optional comma-separated list of transaction department codes. If no value is specified, the program will run triggers for all transaction department codes.		No
1		JOB_BLK_SIZE	Number of records for each spawned job to process. Must be a positive integer.	50	No
1		CLIENT_NAME	Client Name for Report		No
1		SMU_FILE_PREFIX	SMU Parameter File Prefix	FDT	Yes
1		PARALLEL_JOB_COUNT	Number of jobs to keep running	*1*	*No*
1		POLLING_TIME	Number of seconds to wait between polling.	30	No
1		COMMIT_SIZE	Commit Block Size for Transaction Creation (required). This figure determines the number of transactions to be committed at a time for all jobs in the chain that performs transaction submit and can be used for performance enhancement depending upon the server memory configuration.	1	No
1		AMSPARM	The Parameter file location.	$$AMSROOT$	Yes

				$/Parms	
1		SMU_CATALOG_ID	Batch Job Catalog ID for the System Maintenance Utility Job.	3	Yes
2	FDT Process	CHAIN_PARM_FILE	Common Chain Parameter File. Must end in .txt.	FDTParams.txt	Yes
		AMSPARM	The Parameter file location.	$$AMSROOT$$/Parms	Yes
		AMSLOGS	Log Location	$$AMSROOT$$/Logs	Yes
		ROUND_NO	Round Number used by Facilitator.	1	Yes
		RUN_TYPE	Run Type used by Facilitator.	2	Yes
		SUBMIT_FILE_NM	Report file for Round 1 (.txt).	FDTRoundOne.txt	Yes
3	FDT Exception Report	AMSLOGS	Log Location.	$$AMSROOT$$/Logs	Yes
		AMSPARM	Parameter File Location.	$$AMSROOT$$/Parms	Yes
		CHAIN_PARM_FILE	Chain Parameter File.	FDTParams.txt	No
		SUBMIT_FILE_NM	Report file for Exception (.txt).	FDTExcpt.txt	No

Please Note - Every time a parameter fails validation the system logs the error and sets the **Job Return Code** to *Fatal Error* and stops the job.

Major Input

FDT table	FDT_WKLD

Output

- Exception report for transactions that failed to be created.
- If no records meet the selection criteria then the job will stop with a return code of Warning.

Selection Criteria

- Selection criteria for obtaining transactions from the FDT_WKLD table is that the doc_dept_cd is in the parameter (or all DOC_DEPT_CD values if none is specified) and the date is less than or equal to the specified trigger date. If no trigger date is specified then the current application date is used. To be selected, a trigger must also not be expired, that is the trigger's expiration date must be greater than the current application date.

  Order by:
- FDT_WKLD.FDT_SEQ_NO.

Problem Resolution

- If any transaction(s) fail to be created, a report is generated and those transactions will be tried again during the next run of the FDT Chain Job. The user should review this report and assure that the transactions which failed to be created are indeed active. The user should also ensure that the transactions on which the trigger is based contain valid data being carried over to the trigger.
- The conversion script should be run on existing installations prior to running the Future Transaction Trigger Process to ensure that existing triggers are converted to the new process. The conversion script only needs to be run once, and should be run while the system is offline and not processing any existing triggers.
- If the process fails to complete at any stage, the following actions can be taken to restart the chain:
- If the job fails during the FDT Preprocessor, then the entire chain must be restarted.
- If the job fails during the FDT Facilitator, each spawned SMU job must be restarted. Then the FDT Facilitator can be restarted and the chain will complete.
- If the job fails during the Report process, the report can be restarted to complete the chain.
- If any transactions that fail to be created and are in the exception report, they will be attempted again during each subsequent run of the chain until their expiration date. The user must use the exception report and/or log to find out why the transaction(s) failed to create, and correct the problem.
- Possible causes for failure to create a transaction:
- New or Copy action not allowed for that transaction.
- Version number specified in the trigger does not exist or is not in the final state.
- Department code is invalid or inactive.
- Source Transaction has been cancelled.

2.1.11 Inbound Interface Reconciliation Report

Chain or Job Name	Inbound Interface Reconciliation Report
Recommended Frequency	Daily as part of the nightly cycle, and on demand
Single Instance Required	Yes
Can be restarted?	No
Reports generated	The following report is generated: • Inbound Interface Reconciliation Report

Overview

The Inbound Interface Reconciliation Report process is used in conjunction with the Interface Reconciliation table of Advantage Financial. The report selects records from the Interface Reconciliation table, reads the batch log tables, and then summarizes interfaced transaction information for the Transaction Load and Submit Status of each Interface File Name found on the selected Interface Reconciliation record. The process uses the summarized information to generate the Inbound Interface Reconciliation report.

Note: Interface Reconciliation table information is partly populated by a non-Advantage process which loads and submits interfaced transactions into Advantage Financial. As part of the non-Advantage process, the following data regarding the interfaced transactions is loaded into the Interface Reconciliation table and is used as part of the selection criteria logic for the Inbound Interface Reconciliation process.

- Application Date
- Transaction Code
- Transaction Department
- Source File Name

The report process summarizes Load Success Count and Load Fail Count along with the Submit Success Record Count and Submit Reject Record Count from the corresponding batch logs. The selected Interface Reconciliation records are updated with the summarized counts. Additionally, the Load Status and Submit Status are updated by comparing the Load Success/Fail and Submit Success/Reject counts, respectively.

Steps for running the process:

1. Parameter Validation
2. Select Interface Reconciliation Records
3. Calculate and Update Load Status and Submit Status
    a. Calculates Load Success Count by getting the sum of all the values following ‘Rows Saved’ in the batch job log message (MMO field of BS_AGENT_LOG). The process looks up the Batch Job Log (BS_AGENT_LOG) table based on the Job ID (AGENT_ID) of the Multi Process Import job that loads and submits the interface file and all System Maintenance Utility jobs that are spawned by the Multi Process Import job. The File Name from the Interface Reconciliation table is used to find the Multi Process Import Job on the Batch Job (BS_AGENT) table.
    b. Similarly, calculates Load Fail Count by getting the sum of all the values following ‘Rows in Error’ in the batch job log message (MMO field of BS_AGENT_LOG).

c. Calculates Submit Success Record Count by getting the sum of all the values following 'Succeeded' in the batch job log message (MMO field of BS_AGENT_LOG). The process looks up the Batch Job Log (BS_AGENT_LOG) table based on the Job ID (AGENT_ID) of the Multi Process Import job that loads and submits the interface file and all System Maintenance Utility jobs that are spawned by the Multi Process Import job. The File Name from the Interface Reconciliation table is used to find the Multi Process Import Job on the Batch Job (BS_AGENT) table.

d. Similarly, calculates the sum of all the values following 'Failed' in the batch job log message (MMO field of BS_AGENT_LOG) to get the count of transactions that failed to submit.

e. Calculates Submit Reject Record Count by getting the difference of the Submit Success Record Count and 'Failed' count captured as above.

f. Load Status is set to 1 if Load Success Count matches to the count set during PDI load, that is, Data Junction Header Count on the Interface Reconciliation table. Otherwise, Load Status is set as 0.

g. Also, Submit Status is set to 1 if Submit Success Count matches to the Load Success Count on the Interface Reconciliation table. Otherwise, Submit Status is set as 0.

4. Generate Inbound Interface Reconciliation Report

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters • Parameters are valid or invalid depending on the Validation. • Batch Parameter validation completed
2. Selection of Records	• If the selection returns 0 records, then the following messages will be issued: • No records were selected for processing based on report parameters. • No data found for the report.
3. Calculate and Update Load Status and Submit Status	• If the filter returns 0 records for update, then the following messages will be issued: • There is no data to update on the Interface Reconciliation Table. • No data found for the report. • If runtime exception while extracting or updating Interface Reconciliation record(s), then the following messages will be issued: • Exception in extracting Interface Reconciliation Table • Exception in processing the extracted Interface Reconciliation records • Exception while updating Interface Reconciliation records • Exception encountered while executing Interface Reconciliation process
4. Generate Inbound Interface Reconciliation	• If runtime exception while generating the report, then the following messages will be issued: • Exception encountered during report data fetch.

Report	• Exception encountered during report creation.

This process cannot be restarted on failure and requires a new job to be scheduled. No check point information is stored or maintained as the process cannot be restarted.

Major Input

- Interface Reconciliation (R_INT_RECON)
- Batch Job Manager (BS_AGENT)
- Batch Job Logs (BS_AGENT_LOG)

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
Report Date (CONTROL_DATE)	The optional date (format of MM-DD-YYYY) that is used to select Interface Reconciliation records. If left blank the Application System Date defaults.	(blank)
Batch Catalog ID (CTLG_ID)	The required ID of the Multi Process Import job for selection of job logs.	1267

Major Output

- Interface Reconciliation (R_INT_RECON)
- Inbound Interface Reconciliation Report

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected interface reconciliation records have been processed successfully, a report has been generated, and records are updated successfully.
Warning (4)	No eligible records found. This could be because of the following reasons: • No records found on the Interface Reconciliation table for the CONTROL_DATE parameter.
Non-Fatal Error (8)	This return code is not issued for this job
Failed (12)	The job will fail under the following conditions: • 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 Criteria

Records written to the report are sorted by:

- Transaction Department Code (DEPT_CD)
- Transaction Code (DOC_CD)
- Source File Name (FILE_NAME)

Selection Criteria

Interface Reconciliation record with Application Date (APP_DATE) that matches the Report Date (CONTROL_DATE) parameter or if left blank then the Application System Date.

The File Name from the Interface Reconciliation table is used to find the Multi Process Import Job on the Batch Job (BS_AGENT) table.

Problem Resolution

No database restore is required. Correct the problem and rerun the job executing the program. No restoration of data sets or files from backups is required for this program.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully. All of the selected records are processed and reported successfully.	N/A	N/A
Warning (4)	Job ended because there is no record on the Interface Reconciliation table or no matching record for the provided Report Date parameter. **Sample Message:** There is no data to update on the Interface Reconciliation Table	Ensure there are records on the Interface Reconciliation table for the provided parameter value and schedule a new job.	Alternatively, the job can be rescheduled with different parameters.

	No data found for the report		
Non-Fatal Error (8)	This Return Code is not issued for this job.	N/A	N/A
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following condition: • Encounters any runtime exceptions If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and then the new job can be scheduled.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The new job can be scheduled.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The new job can be scheduled.	N/A

2.1.12 Interface Batch Demand Report

Job Name	Interface Batch Demand Report
Recommended Frequency	On Demand
Single Instance Required	No
Can be restarted?	No
Reports generated	Yes

Overview

The Interface Batch Demand Report generates a report based on the data captured in the Processed Batch Interface Event (BIEVNT) page. The job selects BIEVNT records based on the date provided in the DATE batch parameter and based on the processing status of the BIEVNT record.

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value is displayed in the log. • Batch Parameter validation completed
2. Selection of Records	• Processing of BIEVNT record(s) for report is started. • If the selection returns 0 records, then the following message is displayed: “No records available of this date”. • Number of records (count) selected is displayed: “Number of records processed: (count)” • At the end, the following message is displayed: “Processing of BIEVNT record(s) for report completed”.
3. Report generation	• Rendering report started • Rendering report completed

Major Input

Tables

- Processed Batch Interface Event (BIEVNT/BI_EVNT) records

Batch Parameters

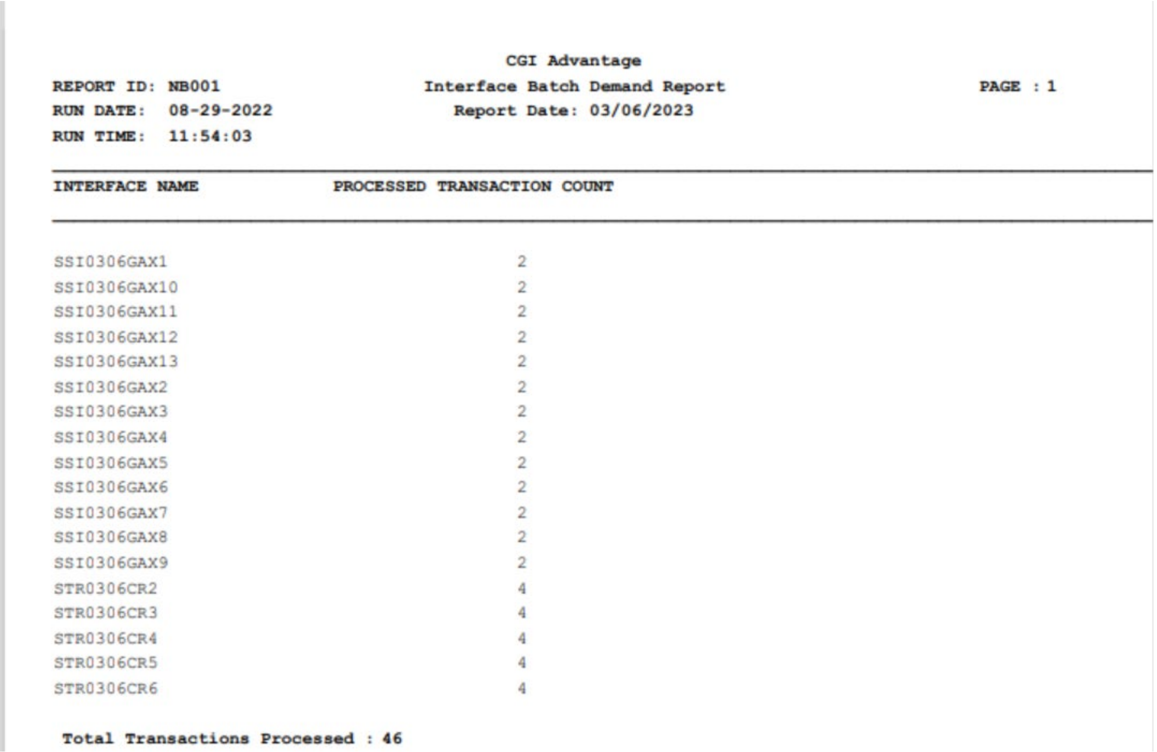
Parameter	Description	Default Value
CLIENT_NM	Client name for report Used to print the client’s name on the report.	

Parameter	Description	Default Value
DATE	Current Date (MM/DD/YYYY) used to pick the BIEVNT records.	
PROG_COUNT	Progression Counter used for the Progression message Counter.	100
REPORT_ID	Report ID Used to print the Report ID in report.	
SELECT_BLOCK	Select Block Size Controls the number of records selected from BIEVNT.	100

Major Output

The Interface Batch Demand Report is produced by this job. This report lists the processed batched interface records from the Batch Interface Event table and the number of processed interfaced transactions for the selected BIEVNT record.

Sample report:



```
                                     CGI Advantage
REPORT ID: NB001              Interface Batch Demand Report                   PAGE : 1
RUN DATE:  08-29-2022            Report Date: 03/06/2023
RUN TIME:  11:54:03
_______________________________________________________________________________________
INTERFACE NAME            PROCESSED TRANSACTION COUNT
_______________________________________________________________________________________

SSI0306GAX1                              2
SSI0306GAX10                             2
SSI0306GAX11                             2
SSI0306GAX12                             2
SSI0306GAX13                             2
SSI0306GAX2                              2
SSI0306GAX3                              2
SSI0306GAX4                              2
SSI0306GAX5                              2
SSI0306GAX6                              2
SSI0306GAX7                              2
SSI0306GAX8                              2
SSI0306GAX9                              2
STR0306CR2                               4
STR0306CR3                               4
STR0306CR4                               4
STR0306CR5                               4
STR0306CR6                               4

 Total Transactions Processed : 46
```

Sort Criteria

Records written to report are sorted by,

- Interface Name (BATCH_ID).

Selection Criteria

The Interface Batch Demand report selects records from the BIEVNT table based on the following selection criteria:

- The Import Date on the BIEVNT equals the date provided in the Date parameter.
- The Processing Status on the BIEVNT record is equal to Received/Processed.

Problem Resolution

The following table shows the potential job return codes for this job. No database restore is required. Correct the problem and rerun the job executing the program. No restoration of datasets or files from backups is required for this program.

Return Code	Condition
Successful (1)	The job ends as successful when the Report is generated with processed BIEVNT records.
Warning (4)	A warning results when no BIEVNT record matches selection criteria. This occurs if there are no records with a matching Import Date and/or the Processing Status of the
Non-Fatal Error (8)	No scenario.
Failed (12)	The job fails under the following conditions: • Parameters are invalid. • Run time exceptions for unexpected situations.
Terminated (16)	The job is terminated by the user.
System Failure (20)	A System Failure is displayed when the job is terminated because of database server or network issues.

2.1.13 Interface Transaction Exception Report

Chain or Job Name	Interface Transaction Exception Report
Recommended Frequency	Daily as part of the nightly cycle, and on demand
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Interface Transaction Exception Report

Overview

The Interface Transaction Exception Report process reports failed and rejected interfaced transactions. The report selects records from the Interface Reconciliation table and reads the batch log tables for the transactions that failed to load or submit.

Note: The Interface Reconciliation table information is partly populated by a non-Advantage® process which loads and submits interfaced transactions into Advantage® Financial. As part of the non-Advantage® process, the following data regarding the interfaced transactions is loaded into the Interface Reconciliation table.

- Application Date
- Transaction Code
- Transaction Department
- Source File Name

The report process lists the transactions that failed to load as error and those failed to submit as reject. Along with this indicator, report details the transaction information and error messages issued for failure.

Steps for running the process:

1. Parameter Validation
2. Select Interface Transaction records failed to load
3. Select Interface Transaction records failed to submit
4. Generate Interface Transaction Exception Report

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters • Parameters are valid or invalid depending on the Validation. • Batch Parameter validation completed
2. Selection of transaction records that failed to load	• Found Error Transaction: <<Transaction Identifier – Trans. Code, Trans. Dept, ID>> • If runtime exception while extracting Interface transaction record(s) failed to load, then the following message is issued: • Error encountered while getting transactions that failed to load. Please contact your

	administrator
3. Selection of transaction records that failed to submit	• If Exception Log File is not found, then following message is issued: • Exception file not found: <<Log File Name>> • Found Reject Transaction: <<Transaction Identifier – Trans. Code, Trans. Dept, ID>> • If runtime exception while extracting Interface transaction record(s) failed to submit, then the following message is issued: • Error encountered while getting transactions that rejected. Please contact your administrator
4. Generate Interface Transaction Exception Report	• If no records to report on, then the following message is issued: • No Records Found • If exception log file(s) are not available to generate report, then the following message is issued: • Unable to create a full Interface Transaction Exception Report. One or more exception log files were not found. • Number of records inserted in report: <<count>> • If runtime exception while generating report, then the following message is issued: • Error encountered while creating the report. Please contact your administrator

This process cannot be restarted on failure and requires to schedule a new job. No check point information is stored or maintained as process cannot be restarted.

Major Input

- Interface Reconciliation (R_INT_RECON)
- Batch Job Manager (BS_AGENT)
- Batch Job Logs (BS_AGENT_LOG)

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
AMSLOGS	The required location of the exception log files.	$$AMSLOGS$$
CLIENT_NAME	An optional name to appear at the top of the report.	(blank)

Control Date (CONTROL_DATE)	An optional selection parameter used to select the interface transaction records that failed to load or submit. If left blank, the Application System Date defaults. Enter as MM-DD-YYYY	(blank)
Department Code (DEPT_CD)	An optional selection parameter of a single department.	(blank)
Transaction Code (DOC_CD)	A required selection parameter for a single transaction code for the error report.	(blank)
Error Display (ERR_DISPLAY)	The required indication for the level of error display in the report: • 0 - Errors, warnings and informational • 1 - Errors and warnings • 2 - Only errors	(blank)

Major Output

- Interface Transaction Exception Report

Job Return Code

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

Return Code	Condition
Successful (1)	All of the interface exceptions to load and submit transactions are captured and reported successfully.
Warning (4)	No eligible records found to report. This could be because of the following reasons: • No interface transactions resulted in error to load or submit.
Non-Fatal Error (8)	One or more exception log files are not found or available to report.
Failed (12)	The job may fail under the following conditions: • 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 Criteria

Records selected to process are sorted by,

- Job ID (AGNT_ID)
- Log Counter (LOG_CTR)

Selection Criteria

For transactions that failed to load, the records are selected from the BS_AGENT and BS_AGENT_LOG tables with the following conditions:

- Run Start Time (RUN_DT_TM) for the batch job (BS_AGENT) matches the date specified in the Date parameter and the job ID is of type System Maintenance Utility or Multi Process Import job.
- Job Name (JOB_NM_UP) for the batch job (BS_AGENT) contains the file name of the original input interface file.
- Job Log (MMO from BS_AGENT_LOG) message containing one or more issues as,
    - Job Log (MMO from BS_AGENT_LOG) contains primary keys of transactions that failed to load.
    - Job Log (MMO from BS_AGENT_LOG) contains error indicating that the transaction had a bad field and could not load.
    - Job Log (MMO from BS_AGENT_LOG) contains error indicating that the transaction had some bad data and could not load.
    - Job Log (MMO from BS_AGENT_LOG) contains error indicating that the line number limit was exceeded for transaction component(s) and could not load.

For transactions that are rejected during validation, the records are selected from Exception Log File with the following conditions:

- Transaction Code (DOC_CD) matches the Transaction Code parameter.
- Transaction Department Code (DOC_DEPT_CD), if entered, matches the Department parameter.

Problem Resolution

No database restore is required. Correct the problem and rerun the job executing the program. No restoration of datasets or files from backups is required for this program.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully. All of the selected records are processed and reported successfully.	N/A	N/A

Warning (4)	Job ended because there is no interface transaction record failed to load or submit or no matching record for the provided parameters. **Sample Message:** No Records Found	Ensure there are records that are failed to load or submit for provided parameter value and schedule a new job.	Alternatively, the job can be rescheduled with different parameters.
Non-Fatal Error (8)	Job ended because one or more exception log files are not found. **Sample Message:** Exception file not found Unable to create a full Interface Transaction Exception Report. Exception log files were not found	Ensure expected files are available and schedule a new job.	Alternatively, the job can be rescheduled with different parameters.
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following condition: • Encounters any runtime exceptions If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and then the new job can be scheduled.	N/A.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The new job can be scheduled.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the system failure needs to be investigated. The new job can be scheduled.	N/A

2.1.14 Job Manager Output Archive

Description

The Job Manager Table Archive process is a chain job. The main purpose of this chain job is to archive records from the BS Agent, BS Agent Log, and BS Agent Parm tables as per user specified selection criteria. The selected records are archived using the System Maintenance Utility "Table Archive" command.

Unlike several other archiving jobs, this chain does not implement the Data Warehouse Archived Record Queue process, which is used to retain the table names and key values for records that are archived and deleted. The infoAdvantage reporting system uses this information to distinguish between records that have been archived (as valid historical data) before deletion and records that are simply deleted from the system (as no longer needed). For Job Manager Output Archive, the data being archived is not critical for reporting needs.

This chain job consists of two individual processes:

Job No	Job	Description	Output
1	Job Manager Archive	The main purpose of this job is to select records for archiving from the BS Agent table per predefined selection criteria of date. Using key information from the selected BS Agent records the process will then select related child records from the BS Agent Log and BS Agent Parameter tables. As records are selected the process will build System Maintenance Utility parameter files. Archive Facilitator table records will be created by the selection process, one record for each generated parameter file. Finally, the selection process will generate a user-friendly report documenting all records selected for archive.	• Generates System Maintenance Utility parameter files • Inserts new Archive Facilitator record for each parameter file generated. • Archive Report detailing all selected records
2	Archive Facilitator	The Archive Facilitator process will read the Archive Facilitator records inserted by the first job for the specified Run Number, and for each record found whose Facilitator Status = "Table	• Generates System Maintenance Utility XML files containing the archived records from the BS Agent, BS Agent Log, and BS Agent Parm tables. • Generates System Maintenance Utility report if the

		ready for archive" the Facilitator will initiate a System Maintenance Utility process. The System Maintenance Utility will be invoked with an action of "Table Archive" and will archive all records identified in the related parameter file with each individual Archive Facilitator record. As the System Maintenance Utility is processing the file the Facilitator updates the "Facilitator Status" on the Archive Facilitator record being processed. The status is set to "Table Archiving" while the archive is occurring, and once completes the status changes to "Table Archive Complete".	Suppress Reports parameter = N.

When to Run

Because the Job Manager tables are updated each time a batch job is executed in the system, it is recommended that this archiving job be set up to run either on a nightly basis or at some regular interval. When run for the very first time, it is recommended that the parameters be set up so as to archive the data in smaller segments.

Additionally, it is recommended that the archived data be moved off of the application file server at regular intervals in order to maintain adequate disk space for application processing.

Major Input

- BS Agent (BS_AGENT)
- BS Agent Parm (BS_AGENT_PARM)
- BS Agent Log (BS_AGENT_LOG)

Output

- Job Manager Output Selection Report
- SMU Report(s)
- Parameter file(s) with Job Manager Tables' record keys
- Add records to the Facilitator table with the parameter file name(s)

The SMU Report(s) will only be generated if the Suppress Reports parameter is set to Y. In this case, a standard SMU report will be generated for each SMU parameter file created during the archiving process. In order to access the SMU report, find the SMU Job ID spawned by the Facilitator step. Then, find the SMU Job ID in the View All Jobs summary list. Click the View Reports link to see the SMU report.

Sort Sequence

The report is ordered by AGENT ID of BS Agent table.

Parameters

Batch Parameters

Job	Parameter	Description	Default Value
Job Manager Archive	Client Name (CLIENT_NM)	Optional field. This parameter defines the client name for Report	No Default
	Number of Days for Archive (ARCH_ DAYS)	Required Field. This parameter should be numeric, positive, and a whole number. It is subtracted from the current application date to derive an eligible for archive date which is date on which BS Agent records become eligible for archive	No Default
	File Location (FILE_LOCATION)	Required Field. This parameter defines the location to which the system generated parameter and XML files will be written.	No Default
	Report Only (REPORT_ONLY)	Required Field. This parameter defines the "Report_Only" parm to define if records are to be purged in addition to reporting on the selected records.	No Default

	Suppress SMU Reports Y or N (SUPPRESS_RPTS)	Required Field. This parameter defines the "SUPPRESS_REPORTS" parm in the parameter file. When this parameter is defined as “Y” a related SMU report will not be generated for each processed parameter file. If “N” one SMU report is generated for each parameter file.	N
	Tolerance (TOLERANCE)	This parameter defines a benchmark for the approximate number of records to be written to each XML archive file.	1000
Archive Facilitator	Archive Restore ID(1-Table Archive) (ARCHIVE_RESTORE_ID)	Required non editable field. Entry of a value in this field specifies the System Maintenance Utility command to be executed during the Archive. Must be “1” for Archive.	1
	Commit Block Size (COMMIT_BLOCK_SIZE)	Required field. Controls how many records are committed by the application at one time. The size should be compatible with technical capabilities and performance guidelines	1000
	Number of Processors (PROCESSOR_NO)	Required Field. Defines the number of SMU processes to be executed simultaneously	2

	Run Number RUN_NO	This field need not be entered by user during SMU action "Table Archive"	
	Sleep Time (Frequency to pool for free processing slots) SLEEP_TIME	Defines the frequency to pool for free processing slots when initiating SMU processes.	5
	Update Facilitator Status UPDATE_STATUS	Update Facilitator Status. If Report Only is Y Update Status must be N.	Y

Please Note - Every time a parameter fails validation the system logs the error and sets the **Job Return Code** to *Failed*, which in turn deactivates further jobs down in the chain, and prevents them from executing.

Also this Job spawns multiple System Maintenance Utility (SMU) jobs in **Report Only** Mode as well.

- Sort Sequence

The report is ordered by AGENT ID of BS Agent table.

Selection Criteria

The BS Agent records whose RUN_EXPR_TM date is less than or equal to the system derived Archive Eligibility Date are selected for archive. The Archive Eligibility Date is derived by subtracting the "Number of Days for Archive" parameter from the current application date as defined on the "APPL_SYS_DT" entry on the application parameters table. The BS Agent Log and BS Agent Parm records whose AGNT_ID equals the AGNT_ID of each selected BS Agent record is selected for archive.

Problem Resolution

If the chain process was discontinued for any reasons then each of the jobs in the chain has the ability to be restarted from the point it left off. The Restart ability of the jobs can be used anytime except in the case of program errors or exceptions.

2.1.15 Job Manager Output Restore

Description

The main purpose of this job is to restore previously archived records of the BS Agent, BS Agent Log, and BS Agent Parm tables as per a user specified archive Run number. The archived records are restored using the System Maintenance Utility "Table Unarchive" command.

When to Run

On-demand

Major Input

- System Maintenance Utility XML files containing the archived records from the BS Agent, BS Agent Log, and BS Agent Parm tables.

Output

- Restores BS Agent record
- Restores BS Agent Log records
- Restores BS Agent Parm records

Parameters

Batch Parameters

Parameter	Description	Default Value
Archive Restore ID(2- Table Unarchive) (ARCHIVE_RESTORE_ID)	Required non editable field. Entry of a value in this field specifies the System Maintenance Utility command to be executed during the Archive. Must be “2” for Restore.	2
Commit Block Size (COMMIT_BLOCK_SIZE)	Required field. Controls how many records are committed by the application at one time. The size should be compatible with technical capabilities and performance guidelines	1000

Number of Processors (PROCESSOR_NO)	Required Field. Defines the number of SMU processes to be executed simultaneously	2
Run Number *RUN_NO*	Required Field. Defines the run number associated with the XML files to be restored.	
Sleep Time (Frequency to pool for free processing slots) SLEEP_TIME	Defines the frequency to pool for free processing slots when initiating SMU processes.	5
Update Facilitator Status UPDATE_STATUS	Defines if the Facilitator Status field on the associated Archive Facilitator record should be updated	Y

Please Note - Every time a parameter fails validation the system logs the error and sets the **Job Return Code** to *Failed.*

- Sort Sequence

The records are restored in the order of parent/child (BS Agent, BS Agent Log, BS Agent Parm) record order as well to maintain referential integrity across the three BS Agent related tables.

Selection Criteria

The System Maintenance Utility will select and will restore all records found within the XML file associated with each individual Archive Facilitator record having the user specified Run number.

Problem Resolution

If the process was discontinued for any reasons then the job has the ability to be restarted from the point it left off. The Restart ability of the job can be used anytime except in the case of program errors or exceptions.

2.1.16 Journal Ledger Archiving

Chain Name	Journal/Ledger Archiving
Recommended Frequency	On Demand
Single Instance Required	Yes
Can be Restarted?	N/A
Reports Generated	Yes, the process has run modes Full Run and Report Only.

Overview

Historical trails and summarizations of transaction data is the most important output of any financial application. CGI Advantage Financial is delivered with a large set of journals to record transaction details and ledgers to record summarized details. A single table, the Journal/Ledger Control Detail (JLCTRL) table supports the use and options of historical data sources.

As historical stores of information, journals and ledgers will grow very large over time. An archive process is available to remove records for a specified year from most journals and ledgers. The following journals can be archived: Accounting, Cash, Cost Accounting, BFY Not Equal FY, and the 1099 journal. Any ledger can be archived as long as it contains Fiscal Year or Budget Fiscal Year in the summarization criteria.

How many years of data are retained in the online application is a function of policy as well as one of technical limitations. There are only a couple of rules for archiving journal or ledger data in the application that must be followed:

1. A year cannot be archived from a journal or ledger if the Closing Process Run value for that year on the Fiscal Year (FY) table is not set to *true.* Even though 1099 Journals are archived by Calendar Year, this validation is still performed by comparing the indicated Calendar Year to the corresponding Fiscal Year record to see if that Fiscal Year is closed.
2. A ledger marked as System Assurance 01 or System Assurance 02 (detailed later) cannot be archived for a year if that year has not already been pre-archived by that systems assurance process.
3. A journal marked as System Assurance 09 (detailed later) cannot be archived for a year if that year has not already been pre-archived by that systems assurance process.
4. A ledger cannot be archived for a year if the Source Journal for that ledger has not also been archived for that year. This means that the application can retain summary data for past years while allowing the archival of the detail data (journals) for those past years.

The Journal/Ledger Archiving process optionally archives data from journals and ledgers as CSV or XML files using export functionality and purges the system of data no longer required. Once parameters are validated and records are selected for a given BFY, FY, or Calendar Year (for 1099 journals), facilitator records are submitted to the Job Manager where the records from journal/ledger tables are exported using parallel processing. Once the process has successfully completed, the system will purge the archived records from the journal, ledger, and cross reference table. The Journal/Ledger Control and Fiscal Year tables are then updated to reflect the new archive status.

This run sheet is written with the assumption that deleted journal or ledger records will be written out to CSV or XML files. While it is common to write out as such, but then move the generated files off to external storage to make the biggest improvement from the archive, one may choose to just not create this output initially because there is no desire for the possibility to restore the data. In order to make the process not create CSV or XML output (reports will still be produced), the configuration required to not create the output is the middle job step, Archive Facilitator, is disabled in the chain to allow only job steps one and three. Additionally, the Verify Facilitator Counts parameter in the Post-Archive Process job step must be set to *No.*

The following jobs comprise the Journal/Ledger Archiving process:

- Archiving Preprocessor
- Archive Facilitator
- Post-Archive Process

The steps are all singleton jobs. This means that only one of their instances can be run at a time. The Archive Facilitator does have the ability to launch the System Maintenance Utility jobs in parallel.

The output from the process is a report that details the number of records that will be archived from the journal, ledgers and Journal Ledger Cross Reference table and a report that lists the name of each file that will be exported. When run in Report Only mode, the chain ends after Archiving Preprocessor completes with only number of records information.

The acceptable job return codes (configured in the Configure Chain Job section of the Job Setup in CGI Advantage) for the jobs in the Journal/Ledger Archive chain are delivered to be set to Successful. As with all CGI Advantage chain jobs, these acceptable return codes are configurable and may be changed to meet certain requirements. For simplicity, this run sheet assumes the chain is configured as delivered, including parameter settings.

Unlike several other archiving jobs, this chain does not implement the Data Warehouse Archived Record Queue process, which is used to retain the table names and key values for records that are archived and deleted. The infoAdvantage reporting system uses this information to distinguish between records that have been archived (as valid historical data) before deletion and records that are simply deleted from the system (as no longer needed). For Journal/Ledger Archiving, infoAdvantage determines this information by consulting the Last Archive Year value on the Journal/Ledger Control Detail (JLCTRL) table and an explicit list of archived Journal/Ledger records is not required.

Major Input

- Journal Ledger Control (R_JRNL_LDGR_CTRL) – Journals and Ledgers must have Available for Archive checked in the Journal/Ledger Control table. If System Assurance 01 is checked, then the SA01 Pre-Archive Year must have been set by the System Assurance 01 chain job to match the Archive FY parameter. If System Assurance 02 is checked, then the SA02 Pre-Archive Year must have been set by the System Assurance 02 chain job to match the Archive FY parameter. If System Assurance 09 is checked, then the SA09 Pre-Archive Year must have been set by the System Assurance 09 batch job to match the Archive FY parameter.
- Fiscal Year (R_FY) – The FY table must have Closing Process Run checked for the Archive FY parameter.
- SORT_PARM_FILE – The sort file used defines all of the files used for the archive. If the file has a sort value defined, only one value should be defined. When a sort value is

defined, the system settings such as Polling Time, Parallel Job count and Block Size, should be optimized. This will enable more efficient processing of the job.

- Journals and Ledgers – The Archive Tables parameter determines which tables are archived. It can be set with specific tables. If left blank, all tables with the Available for Archive flag checked in the Journal/Ledger Control table will be archived.
- Journal/Ledger Cross Reference (JRNL_LDGR_XREF) – All records in the Journal Ledger Cross Reference table that reference exported and deleted journal records will be deleted. Because a year has to be archived from the source journal before it can be archived from a ledger, the archiving of a ledger will not report any JRNL_LDGR_XREF activity.

Major Output

- Exported CSV/XML files
- Journal/Ledger Archive Report
- Journal/Ledger Facilitator Report
- Fiscal Year (R_FY)
- Journal/Ledger Control (R_JRNL_LDGR_CTRL)

Chain Job Return Code

The following table shows the potential return codes for the Journal/Ledger Archiving chain. Note that the chain job will end with the highest return code across all of the jobs.

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

Problem Resolution

Please refer to the individual job “Problem Resolution” section for more details.

Journal/Ledger Archiving Chain: Archiving Preprocessor Job

Job Name	Archiving Preprocessor
Recommended Frequency	On Demand This job must be run as part of the Journal/Ledger Archiving Chain.
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	Journal/Ledger Archiving report lists the number of journal/ledger records to be archived in the Archive Facilitator step. In addition, the number of JLXREF records to be deleted is listed.

Overview

This step first validates parameters and determines whether each table is ready to be archived. If parameters are valid, it then divides the rows in each table to be archived into groups for each export file. For example, if LDGR_SA_BUD has 20,000 rows for BFY/FY 2000 and the Job Block Size parameter is 10,000, then two export files are created. The number of records eligible for archival is written to the Journal/Ledger Archiving Report. If the run mode is set to 'Report Only', the process ends at this point.

- In Full mode, a System Maintenance Utility parameter file is created for each export file containing the table name, the beginning and ending record number, and other settings. A Facilitator table record referencing the SMU parameter file is created and saved. Facilitator records can be viewed in the Archive Facilitator Inquiry page (QARC) for the status. The file names for the archived records are written to the Journal/Ledger Facilitator Report.

Process Steps	Messages
1. Parameter Validation	• Validating batch parameters. • If the parameter is invalid, the invalid value is displayed in the log along with the error message. • Reading sort parameter file. • (Sort Parameter File Name) contains valid parameters. • Batch parameters are valid.
2. Pre-processing	The following messages are issued when the job runs in Full Mode. • Table division started. • Table division completed. • Rendering report started. • Rendering report completed. • Writing chain parameter file.

	- Chain parameter file (Chain Parameter File Name) written in directory (AMS Parameter Directory). The following messages are issued when the job runs in Report-Only Mode. - Table division started. - Table division completed. - Rendering report started. - Rendering report completed.

Restartability Information

This job cannot be restarted. If the job has failed for any reason, schedule a new job after correcting the errors that caused the initial job to fail.

Major Input

- Journal/Ledger Control (R_JRNL_LDGR_CTRL) – Journal and Ledger tables must have Available for Archive checked in the Journal/Ledger Control table. If System Assurance 01 is checked, then the SA01 Pre-Archive Year must have been set by the System Assurance 01 chain job to match the Archive FY parameter. If System Assurance 02 is checked, then the SA02 Pre-Archive Year must have been set by the System Assurance 02 chain job to match the Archive FY parameter. If System Assurance 09 is checked, then the SA09 Pre-Archive Year must have been set by the System Assurance 09 batch job to match the Archive FY parameter.
- Fiscal Year (R_FY) – The FY table must have Closing Process Run checked for the Archive FY entered as a parameter.
- Journals and Ledgers – The Archive Tables parameter determines which tables are archived. It can be set with specific tables or left blank. If left blank, all tables with Available for Archive checked in the Journal/Ledger Control table is archived.
- Journal/Ledger Cross Reference (JRNL_LDGR_XREF) – All records in the Journal Ledger Cross Reference table that reference exported and deleted journal records are deleted. Because a year has to be archived from the source journal before it can be archived from a ledger, the archiving of a ledger does not report any JRNL_LDGR_XREF activity.

Major Output

- Journal/Ledger Archiving Report

Batch Parameters

Parameter Name	Description	Default Value
Export Location at Archiving Preprocessor Job	The required directory location where the export files is written. Note that the size of archived	$$AMSROOT$$/ExportImport

	files may be so great that a location other than the default Export/Import directory should be used.	
Parameter Location at Archiving Preprocessor Job	The required directory location where the file designated by the Sort Parameter File parameter is located.	$$AMSROOT$$/Parms
Archive Tables	An optional listing of a subset of journal and/or ledger tables to be exported where the Journal Ledger Control (JLCTRL) record for the table has the Available for Archive set to *true.* If left blank, all tables available for archiving are archived.	(blank)
Archive Year	A required year used for journal and ledger record selection (FY or BFY, depending on the table and its record in Journal/Ledger Control). For 1099 journals, record selection occurs based on the Calendar Year (CY) of the 1099 Journal.	(blank)
Common Chain Parameters File (.txt)	A required text file name that is created with parameters for subsequent steps in the chain.	JLArchParams.txt
Client Name for Report	An optional name to appear as the first line in the header of the reports created. If left blank, the system wide default for Client Name (if established) is be used.	
Commit Block Size	A required performance parameter to determine how often database transactions are committed.	1000
Export File Type (1 - XML, 2 - CSV)	A required parameter specifying the export file type, 1 for XML (Extensible Markup Language) or 2 for CSV (Comma Separated Values). Note that CSV is strongly recommended because it is more compact.	2
Number of Lines per Archive File	A required maximum number of records to be written to each export file.	10000
Run Mode (1 - Report Only, 2 -	The required run mode: 1	1

Report and Archive)	(Report Only) or 2 (Report and Archive).	
SMU Facilitator Parameter File Prefix	A required prefixed used to determine a SysManUtil parameter file name for each Facilitator job/export file. Do not change the value delivered.	JLArch
Sort Parameter File	A required parameter of a text file that lists by journal and ledger an optional field that is used to sort the export files. If a sort field is provided for a given table, records are divided into separate files for each value in the sort field. Note that sorting can only be done on a field that is populated on every record to be archived.	JLArchSort.txt

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	No eligible records found.
Non Fatal Error (8)	N/A
Failed (12)	The job fails under the following conditions: • Required Parameters are not entered. • Entered parameters are invalid. • Run time exceptions for unexpected situations. When this job ends with a return code of Failed, subsequent jobs in the chain are set to Inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain are set to Inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure subsequent jobs in the chain are set to Inactive.

Sort Criteria

None

Selection Criteria

N/A

Problem Resolution

If the job ends with a return code other than Successful or Warning after completing parameter validation, a new job should be scheduled after the job log has been reviewed (the failed job should not be restarted).

- Performance
- To improve the performance of this job, it is recommended that an index be created on the table JRNL_LDGR_XREF for the fields: JRNL_REC_NO, JRNL_ID, and LDGR_ID. The fields must be used in that order for the job to accurately use this temporary index.
- An example of such an index is:

```
CREATE INDEX jla_temp
  ON jrnl_ldgr_xref (
    jrnl_rec_no,
    jrnl_id,
    ldgr_id
```

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

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation
Successful (1)	All parameters are valid.	N/A
Warning (4)	N/A	This step does not issue this return code.
Non Fatal Error (8)	N/A	This step does not issue this return code.
Failed (12)	Required parameters are not entered. **Sample Message:** Parameter Run Mode is required.	Schedule a new job after entering a valid value for the parameter.

	Attempt to archive already-archived year. **Sample Message:** Parameter Archive Tables value JRNL_1099 is not eligible for archive, Last Archive Year in the Journal Ledger Control table is 2010 and must be less than parameter Archive Year.	Schedule a new job after entering a valid value for the Archive Tables or Archive Year parameter. This message is typically issued when an attempt is made to archive a FY/BFY that is earlier than, or equal to, the year being tracked as the last year previously archived for the journal or ledger, in Journal/Ledger Control's (JLCTRL) Last Archive Year field. Please note that as of CGI Advantage 3.11, the JLCTRL Last Archive Year field is interpreted as the last-archived **Calendar Year** for 1099 journals instead of FY or BFY. If at least one archival of the 1099 Journal was completed **prior** to upgrading to the 3.11 release, the Last Archive Year value for the 1099 Journal on the Journal/Ledger Control (JLCTRL) table may need to be adjusted to allow all 1099 journal records from that Last Archive Year to be archived. For example, suppose the 1099 Journal was archived for FY 2010 prior to upgrading to release 3.11. The 1099 Journal is still likely contain records from *calendar year* 2010 but the Journal/Ledger Control (JLCTR) reflects a Last Archive Year of 2010 implying that all of 2010's records had been archived. If Journal/Ledger Archiving was to be run again (in 3.11) with a Last Archive Year of 2010, the system would interpret that as having already archived all of calendar year 2010 from the 1099 Journal effectively orphaning all of the residual calendar year 2010 1099 Journal records not archived prior to the 3.11 upgrade. The 3.11 release included a "data fix" (to be applied during the 3.11 upgrade) to adjust (subtract one year from) the Last Archive Year value (if any) on Journal/Ledger Control records for journals that are 1099 journals. This allows the next execution of Journal/Ledger Archiving (for the 1099 Journal) to archive those residual calendar year records from the "last archived" year.
	Entered parameters are not valid. **Sample Message:** Archive Year value 1899 is not a valid year in the Fiscal Year table.	Schedule a new job after entering a valid value for the parameter.

	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated before scheduling a new job.

Step 2: Pre-processing. This step is performed only if the parameter validation is successful.

Possible Return Codes	Condition	Recommendation
Successful (1)	All of the selected records processed successfully.	N/A
Warning (4)	N/A	This step does not issue this return code.
Non Fatal Error (8)	No tables need to be archived.	Confirm that the Archive Tables parameter contains table names that have not yet been archived before scheduling a new job.
Failed (12)	Failed during attempt to write Chain Parameter File.	Verify that the file system is not full, has the correct access, etc. before scheduling a new job.
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated before scheduling a new job.

Journal/Ledger Archiving Chain: Archive Facilitator Job

Job Name	Archive Facilitator
Recommended Frequency	On Demand This job must be run as part of the Journal/Ledger Archive Chain.

Single Instance Required	Yes
Can be Restarted?	Yes
Reports Generated	No

Overview

The Facilitator job spawns multiple System Maintenance Utility (SMU) jobs. The process gets the records from the Facilitator table for the chain's Job ID. It then reads the file names from these records and spawns multiple SMU jobs, using the parameter file names from the Facilitator table as input for the SMU jobs. The parameter files specify the "Table Export" command with other options set in the Archive Preprocess step.

Process Steps	Messages
1. Parameter Validation	• If any parameter is invalid, the invalid value is displayed in the log along with the error message.
2. Facilitator Job Processing	• The Run Number for this archive/restore process = xxx (xxx being the chain's Job ID) • For each job in the Facilitator for the Run Number: • SMU Job - xxx - Spawned (xxx being the Job ID) • Each spawned job has either of these results: • SMU Job - xxx - Processing completed successfully (xxx being the Job ID) • SMU Job - xxx - Failed (xxx being the Job ID) • The job slept a total of xxx times, for a Total Sleep Time of yyy seconds. (xxx and yyy being the counts)

Restartability Information

This job can be restarted if it fails due to any reason. After restart, the job continues to process the records based on the status of each Facilitator record. Restarting this job step restarts all spawned jobs automatically.

Major Input

- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page.
- BS_AGENT – The job consults the Job Manager for status of jobs that have been spawned.
- Journals and Ledgers – Each Facilitator job exports a subset of records that depends on the FY/BFY being archived, the sort field if present, and starting and ending Record Number.

Major Output

Exported files

Batch Parameters

Parameter Name	Description	Default Value
Archive Restore ID (1 – Table Archive)	A required ID instructing the Facilitator to select the archive table process. Note: Do not change the value delivered.	1
Commit Block Size	A required performance parameter to determine how often database transactions are committed.	1000
Number of jobs for Facilitator to keep running	A required number of jobs to launch and keep submitted/running. Exceeding the number of available VLS is acceptable.	1
Number of seconds to wait between polling occurrences	A required length of time between each iteration, which checks the status of the jobs that are running, and the jobs that are launching for unprocessed Facilitator records.	5
Update Status	A required indication to instruct the Facilitator to update the status of each Facilitator record based on the status of its corresponding job in the Job Manager. Note: Do not change this value from that delivered.	Y

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected payment records are processed successfully.
Warning (4)	The job issues a warning under the following conditions: • No Facilitator records found for Run Number = xxx. Nothing to process. (xxx being the chain's Job ID) • SMU Job Parameter File NOT Found – (file name) Record skipped.
Non Fatal Error (8)	N/A
Failed (12)	The job fails under the following conditions:

	• Required parameters are not entered. • Parameters are invalid. • Run time exceptions for unexpected situations. When this job ends with a return code of Failed subsequent jobs in the chain are set to inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return code of Terminated, subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure subsequent jobs in the chain are set to inactive.

Sort Criteria

None.

Selection Criteria

See parameters.

Problem Resolution

- No database restore is required. Both the Archive Facilitator chain step and each spawned System Maintenance Utility job can be restarted.

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

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation
Successful (1)	All parameters are valid	N/A
Warning (4)	No Facilitator records found for Run Number (Nothing to process).	This is acceptable in the event that there are no records to process. A new chain can be started with only the Post Archiving Process enabled, which updates the Fiscal Year (FY) and Journal Ledger Control (JLCTRL) to reflect that the archive has been completed. Otherwise, the reason for the missing records must be investigated before a new chain job is run.

Non Fatal Error (8)	N/A	This step does not issue this return code.
Failed (12)	Required parameters are not entered. **Sample Message:** SLEEP_TIME is required to run the Facilitator	Enter a valid parameter value and restart the job.
	Entered parameters are not valid **Sample Message:** The SLEEP_TIME must be an integer and > 0	Enter the correct parameter value and restart the job.
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new chain job.
Terminated (16)	Job is terminated manually by the user.	Investigate the reason for the termination, resolve it and restart the job. If the restart is not possible, then schedule a new chain 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 system failure, resolve it and restart the job. If the restart is not possible, then schedule a new chain job with the same set of parameters.

Step 2: Facilitator Job Processing. This step is performed only if the parameter validation step is successful.

Possible Return Codes	**Condition**	**Recommendation**
Successful (1)	All parameters are valid.	N/A
Warning (4)	SMU Job Parameter File not found.	If all files are missing, the reason must be investigated before a new chain job is run. If the files can be located and moved to the expected location, the job can be restarted.
Non Fatal Error (8)	N/A	This step does not issue this return code.
Failed (12)	This return code is issued when any spawned SMU job fails. **Sample Message:** SMU Job – xxx – Failed	Determine the cause for the spawned SMU job to fail and resolve. The spawned job can be restarted independently if desired, and restart the chain job. Also restart any failed SMU jobs.
Terminated (16)	Job is terminated manually by the user.	Investigate the reason for the termination, resolve it and restart the job. If the job

		cannot be restarted immediately, then schedule a new job.
System Failure (20)	When the job is terminated because of database server or network issues.	Investigate the reason for the system failure, resolve it and restart the job. If the job cannot be restarted immediately, then schedule a new job.

Journal/Ledger Archiving Chain: Post Archiving Process Job

Job Name	Post Archiving Process
Recommended Frequency	On Demand This job must be run as part of the Journal/Ledger Archiving Chain.
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	The Journal/Ledger Facilitator Report lists the exported files created by the Archive Facilitator step.

Overview

After validating parameters, the first action of this step of the chain is to confirm that all facilitator records were processed successfully in step 2 of the chain job.

Processing Steps

Process Steps	Messages
1. Parameter Validation	• Validating batch parameters. • If the parameter is invalid, the invalid value is displayed in the log along with the error message. • Batch parameters are/are not valid.
2. Database Updates	• Facilitator completed xxx of yyy jobs for Run ID zzz. (xxx and yyy being counts, zzz being the Chain job's Run ID – The job stops at this point if completed jobs are not the same as the total jobs) • Depending on the number of tables that were archived, for example: • Deleted xxx records from JRNL_LDGR_XREF joined to JRNL_ACTG. (xxx being the count) • Deleted xxx records from JRNL_ACTG. (xxx being the count) • Deleted xxx records from LDGR_SA_BUD. (xxx being the count) • Updated xxx records in R_JRNL_LDGR_CTRL. (xxx

	being the count). • Depending on whether all tables marked Available for Archive in the Journal/Ledger Control table were processed in this run: • Archive complete for xxxx. (xxxx being the Archive Year parameter – this is when Archived in the FY record is updated) • xxx tables remaining to archive for yyyy. (xxx being the count, yyyy being the Archive Year parameter)

Restartability Information

This job cannot be restarted. If the job failed due to any reason a new job can be scheduled – either individually or as part of a new chain job by disabling the first two jobs in the chain, with the same set of parameters after correcting errors that caused the job to fail.

Major Input

- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page. This job checks the table to see if any records remain that have not completed successfully.
- Journals and Ledgers – All rows for the given FY/BFY exported from each table are now deleted.

Major Output

- Journal/Ledger Control (R_JRNL_LDGR_CTRL) – Last Archive Year in the Journal/Ledger Control table is updated with the FY/BFY for each journal and ledger exported.
- Fiscal Year (R_FY) – If all Journals and Ledgers marked Available for Archive in the Journal/Ledger Control table have been archived for the given Archive Year, then Archived is checked in the FY table for the record matching the Archive Year parameter.

Parameters

Batch Parameters

Description (Caption)	Parameter Name	Default Value
Parameter Location at Post Archiving Process Job	The required directory location specified in the first job in the chain where the parameter file is located.	$$AMSROOT$$/Parms
Common Chain Parameters File (.txt)	The required name of the parameter file produced in the first job in the chain.	JLArchParams.txt
Delete Block Size	A required performance parameter that determines the number of records to be deleted in one block. The value should	100000

	not be kept low to degrade performance neither it should be kept very high to result in error while deleting due to volume of records.	
Verify Facilitator Counts	A required indication to trigger the execution of the Archive Facilitator job. A value of *Yes* results in the job creating the XML or CSV output files. A value of *No* results in the job not executing and thus no output files being created.	Yes

Job Return Code

The following table shows the potential job return codes for the System Assurance 01 Report job.

Return Code	Condition
Successful (1)	All the database updates are successful.
Warning (4)	N/A
Non Fatal Error (8)	N/A
Failed (12)	The job fails under the following conditions: • Required parameters are not entered • 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. When this job ends with a return code of Terminated subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure subsequent jobs in the chain are set to inactive.

Sort Criteria

None.

Selection Criteria

The Archive Year and Archive Tables parameters are written to the Common Chain Parameters File in the Archive Preprocessor step.

Problem Resolution

Since the job cannot be restarted, if the job ends with any return code (Failed, Terminated or System Failure) a new chain job with the first two jobs in the chain disabled should be scheduled with the same parameters.

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 Location is required and must be a valid directory.	Enter the correct directory for Parameter Location and schedule a new chain job with only step 3 enabled.	
	Entered Parameters are not valid. **Sample Message:** Parameter Chain Parameter File is required to have file extension "txt".	Enter the correct file name for Chain Parameter File and schedule a new chain job with only step 3 enabled.	
	Failed because of runtime exceptions for an unexpected situation.	The Failure reason needs to be investigated before scheduling a new chain job with only step 3 enabled.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling a new chain job with only step 3 enabled.	
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 chain job with only step 3 enabled.	

Step 2: Database Updates. This step is performed only when the parameters are valid.

Possible Return Codes	Condition	Recommendation
Successful (1)	Database updates completed.	N/A

Warning (4)	N/A	This step does not issue this return code.
Non Fatal Error (8)	N/A	This step does not issue this return code.
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The Failure reason needs to be investigated before scheduling a new chain job with only step 3 enabled.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling a new chain job with only step 3 enabled.
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 chain job with only step 3 enabled.

2.1.17 Matching Tables Archive

Description

The Matching Tables Archive Chain archives records from the six Matching Status tables:

- Matching Status Award
- Matching Status Award Line
- Matching Status Invoice Header
- Matching Status Invoice
- Matching Status Receiver
- Matching Status Payment

This chain involves two jobs. The first job performs the record selection, and the second job is the Facilitator job, which submits multiple instances of the System Maintenance Utility (SMU) to complete the archiving of the records.

The record selection process selects the Matching Status tables' records that are eligible to be archived and writes their record keys to a parameter file, which will then be used by the Facilitator/SMU process to archive the selected records. Records are considered eligible for archive if the following is true:

- Their associated transaction(s) have already been archived off the transaction catalog.
- Their related records are also eligible for archive. Related records are defined as transactions that are referenced by or referenced to other transactions that also have records on the Matching Status tables.

If a particular Award Transaction or record being researched has too many related records, this transaction/record, and all of its related records, will be considered ineligible for archive. This happens when the number of related records is greater than the number of "Transaction Selection Limit", specified in the job parameter.

Once the record selection is completed, one record per generated parameter file is added to the Facilitator (FACILITATOR) table.

This Selection Process also generates two reports:

- Selection Process Report. Lists all Transaction ID's associated to the Matching Table records that are eligible for archive, and if no records are eligible, then this report is generated with a "No records eligible for archive selection" message. (Note: The report shows two totals: the Total Number of Transactions selected for archive, and the Grand Total Number of Records Archived. The report lists only the transaction IDs related to the records that were archived since if a single record for a given transaction was archived, then all records for the transaction were archived. The Grand Total figure is the total number of records (related to the transaction IDs listed) that were archived off of the Matching tables. This batch process does not archive the actual transactions. Archiving transactions can be accomplished either online or through the Transaction Archive batch job.)
- Too Many to Archive Report. If any records fall under the "Too Many to Archive" criteria, then this report is generated, listing all Award Transaction ID's that have too many related records to be archived.

The Facilitator job reads the Facilitator table to get all records for the specific Job ID, and spawns multiple SMU jobs using the parameter file names from the Facilitator table as input. These SMU jobs are submitted with the "Table Archive" command.

This chain implements the Data Warehouse Archived Record Queue process, which is used to retain the table names and key values for records that are archived and deleted. The infoAdvantage reporting system uses this information to distinguish between records that have been archived (as valid historical data) before deletion and records that have simply been deleted from the system (as no longer needed). The process is enabled for most archiving processes when the value of Application Parameter Enable Data Warehouse Archived Record Queue (ENABLE_DW_ARCH_QUEUE) is *True*. Please see the *CGI Advantage System Administration Guide* for more information regarding this process and the Application Parameter.

When to Run

This process is run on demand.

Major Input

- Matching Status Award (MATCH_STA_AWD)
- Matching Status Award Line (MATCH_STA_AWD_LN)
- Matching Status Invoice Header (MATCH_STA_INV_HDR)
- Matching Status Invoice (MATCH_STA_INV)
- Matching Status Receiver (MATCH_STA_RECV)
- Matching Status Payment (MATCH_STA_PYMT)

Output

- Matching Status Tables Archive Selection Report
- Matching Status Tables Too Many to Archive Report
- SMU Report(s)
- Parameter file(s) with Matching Status Tables records keys.
- Add records to the Facilitator table with the parameter file name(s).
  - ARCH_DW_QUEUE_REC – The Data Warehouse Archive Record table keeps track of the values for all primary keys of each record being archived and deleted. The table name and attribute names are stored in ARCH_DW_QUEUE_TBL and linked to this record by unique ID.
  - ARCH_DW_QUEUE_TBL – The Data Warehouse Archive table keeps track of the table name and primary key attribute names of each table being archived. If a record already exists for the table being archived, it will reuse the record. If it does not yet exist, it will be created.
- The SMU Report(s) will only be generated if the Suppress Reports parameter is set to Y. In this case, a standard SMU report will be generated for each SMU parameter file created during the archiving process. In order to access the SMU report, find the SMU Job ID spawned by the Facilitator step. Then, find the SMU Job ID in the View All Jobs summary list. Click the View Reports link to see the SMU report.

Parameters

All parameters listed below should be setup both at the chain level and at the specific job level. The default values should be set at the chain level, and it should be blank at the job level, which will trigger an inference from the values in the chain level. If the parameters from the chain level need to be overridden, then this can be done at the job level.

Job	Parameter	Description	Default Value
Matching Tables Archive	CLIENT_NAME	Client Name	
	ARCHV_TYP	Archive Type (1 - Archive all eligible records, 2 - Archive by Award Transaction ID)	1
	AWD_DOC_ID	Award Transaction ID	
	AWD_DOC_DEPT_CD	Award Transaction Department Code	
	AWD_DOC_CD	Award Transaction Code	
	DOC_SEL_LIMIT	Transaction Selection Limit	
	FILE_LOCATION	File Location. Required Field. Directory for generated parameter and xml files.	
	TOLERANCE	Tolerance (Approximate number of records to be written to each xml archive file)	
Matching Tables Archive Facilitator	PROCESSOR_NO	Number of Processors	2
	COMMIT_BLOCK_SIZE	Commit Block Size	1000
	REPORT_ONLY	Report Only. Required Field. Enter a value of “Y” or “N”. If “Y” SMU will not purge the selected records.	
	RUN_NO	Run Number	
	SLEEP_TIME	Sleep Time	5

	SUPPRESS_RPTS	Suppress SMU Reports Y or N. If changed to "Y" SMU will not generate a report.	N
	UPDATE_STATUS	Update Status	Y
	ARCHIVE_RESTORE_ID	Archive Restore ID	1 ("Table Archive")

Upon completion the batch should state if the job was successful, what the parameters were, and the run date and time.

Selection Criteria

The actual record selection process is complex because of the number of related records each Matching Status Table record and transaction could have. The "Transaction Selection Limit" parameter is used to set the limit on the number of related records an Award transaction can have in order for it to be eligible for archive.

Records can be selected in one of two ways, depending upon the Archive Type. If the Archive Type = 1, then the system will select all eligible records (archived transactions with a PO transaction type and found in the Matching Status Award table). In this case, the user must also enter a Transaction Selection Limit parameter value. If the Archive Type = 2, then the system will search only for the specific transaction entered in the Award Transaction Code, Award Transaction Department, and Award Transaction ID parameters.

For each eligible award transaction, the process should perform the following logic:

5. Select all records from the Matching Status Invoice, Matching Status Receiver and Matching Status Payment tables whose Award Transaction Code, Department and ID match the Award Transaction ID that the process is currently working with. If any of these records are associated with a transaction that has not yet been archived off of the Transaction Catalog, then all of these related records should not be archived.
6. If all associated transactions have already been archived, the process will now need to start researching these transaction IDs to determine whether they reference any other award transaction and if these award transactions are associated to any other Matching Status table records.
    a. For each of the selected Invoice Transaction IDs, the system should select all distinct award transaction ID's from the Matching Status Invoice table where the Invoice Transaction Code, Department and ID matches the Transaction ID of the Invoice Transaction it is working with. If award transactions are found, these transactions will also need to be researched.
    b. For each of the selected Receiver Transaction IDs, the system should select all distinct award transaction ID's from the Matching Status Receiver table where the Receiver Transaction Code, Department and ID matches the Transaction ID of the Receiver Transaction it is working with. If award transactions are found, these transactions will also need to be researched.
    c. For each of the selected Payment Transaction IDs, the system should select all distinct award transaction ID's from the Matching Status Payment table where the Payment Transaction Code, Department and ID matches the Transaction ID of the Payment Transaction it is working with. If award transactions are found, these transactions will also need to be researched.

7. If no records were selected in Step 2, then this award transaction has been fully researched and all related records are eligible and will be written to the parameter file and to the report.
8. If Step 2 did select new Award transactions, then the same research needs to be performed for each of these new award transaction IDs.

Problem Resolution

- Look into the job log for errors. Correct the problem and restart the job.
- If the job fails for any data setup reasons then correct the data setup and schedule a new job.

2.1.18 Matching Tables Restore

Description

The purpose of this job is to restore records that were previously archived from the six Matching Status tables:

- Matching Status Award
- Matching Status Award Line
- Matching Status Invoice Header
- Matching Status Invoice
- Matching Status Receiver
- Matching Status Payment

The restore process consists of the Facilitator job, which spawns multiple System Maintenance Utility (SMU) jobs.

The Facilitator process gets the records from the Facilitator table for a specific Run ID entered in the job parameter. It then reads the file names from these records and spawns multiple SMU jobs, using the parameter file names from the Facilitator table as input for the SMU jobs. These SMU jobs are submitted with the “Table Unarchive” command.

When to Run

This process is run on demand.

Major Input

- Facilitator (FACILITATOR) table
- Parameter files created by the Matching Table Archive Chain process.

Output

- The Matching Status Tables are updated as the records are restored to these tables.

Parameters

Job	Parameter	Description	Default Value
Matching Tables Archive Facilitator	PROCESSOR_NO	Number of Processors	2
	COMMIT_BLOCK_SIZE	Commit Block Size	1000
	RUN_NO	Run Number	
	SLEEP_TIME	Sleep Time	5
	UPDATE_STATUS	Update Status	Y

	ARCHIVE_RESTORE_ID	Archive Restore ID	2 (“Table Unarchive”)

Upon completion the batch should state if the job was successful, what the parameters were, and the run date and time.

Selection Criteria

All records specified in the parameter file(s) will be restored to the Matching Status tables.

Problem Resolution

- Look into the job log for errors. Correct the problem and restart the job.
- If the job fails for any data setup reasons then correct the data setup and schedule a new job.

Note

If a user needs to access *and modify* a transaction related to a matching record, both the matching record and the transaction should be restored.

2.1.19 MRT Generation

Chain or Job Name	MRT Generation
Recommended Frequency	On Demand
Single Instance Required	Yes
Can be Restarted?	No
Can be Rerun?	Yes
Reports generated	None

Overview

To improve performance, static data (data changed infrequently) accessed frequently by the Advantage applications can be stored in an offline file allowing for quicker and more efficient access to the related data. Within the current Advantage baseline this offline file, referred to as the MRT file, holds data that is contained within the security tables listed in the “Major Input” section below; no other tables are eligible for MRT storage within baseline Advantage. Since this data is read from the MRT file rather than from the individual tables themselves newly saved security data on the related tables is not available until the MRT file is updated to include this new data. Updating the file can be accomplished online through the MRT generation process, or through the MRT Generation jobstream.

The MRT Generation jobstream allows for the ability to update the MRT file offline, for example as part of nightly cycle processing. After generation, the MRT file includes related security table data for use across all applications including Administration, Financial, HRM (including ESS), and Procurement Vendor (VSS). The Advantage applications that share the same Administration application will always share a single MRT File, with the exception of VSS which always has its own MRT File.

To be consistent with the online MRT generation process this job can only be run in the Administration Application. The “MRT Generation” will also be available in Procurement Vendor (VSS) due to the fact that Procurement Vendor does not share the same Administration application as other Advantage applications.

Summarized Description of Generate MRT File Location and Generate MRT File Name

MRT Generation allows the user to define the Generate MRT File Location and Generate MRT File Name. If this parameter is left blank, the value will default from the ADV30Params.ini file during execution. However, this is an editable parameter and the user can specify a value different from the default if desired, and any user defined value will always take precedence over the default. If the user-supplied value is different than the default then the ADV30Params.ini file will need to be updated before the related MRT file can be used.

Summarized Descriptions of Each MRT Generation Mode

- The user is able to select either production or debug mode for MRT Generation through the Generate MRT Mode parameter. This is an editable parameter with two possible valid values:
- “1” Production: This parameter will run exactly how Production mode runs when one selects the “Generate MRT (Production)” link from the Generate MRT (GMRT) page in the Administration Application. If left blank this parameter will default to a value of “1” from BATSETUP.

- “2” Debug: This parameter will run exactly how Debug mode runs when one selects the “Generate MRT (Debug)” link from the Generate MRT (GMRT) page in the Administration Application.
- The data file generated by the debug mode is unusable by the application but formatted in an easy to view layout.
-

MRT Generation Process

Upon submission of the “MRT Generation” jobstream all parameters will be validated. This means that if one parameter is invalid all remaining parameters will be validated prior to ending the job with a “Fatal Error” return code and error messages in the log.

If parameters are invalid then the following error messages will be returned in the job log:

> Generate MRT File Location: “Generate MRT File Location is invalid”
> Generate MRT File Name: “Generate MRT File Name is invalid”
> Generate MRT Mode:
>> <Null> Value: “Mode is required”
>> Any character other than “1” or “2”: “Mode is invalid”

Once all parameters are successfully validated the job will be driven by the designated mode and create the MRT file as done online through the Generate MRT (GMRT) page. All security tables designated for the MRT generation process will be processed each time MRT Generation is performed. During execution the following progress messages will be displayed within the job log:

> MRT Generation Started
>
> MRT Mode parameter value: *specified mode value or default from BATSETUP*
>
> File name: *specified file name or default from ADV30Params.ini*
>
> File location: *specified file location or default from ADV30Params.ini*
>
> ***For each table written to the file a message will appear in the job log saying this table has been started.
>
> ***For each table written to the file a message will appear in the job log saying this table has been written.
>
> MRT Generation Completed
>
> Job ended

Process Steps	Messages
1. Parameter Validation	• Run Started • 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. If MRT Generation Mode verification is successful then the log will contain a message stating: • parameter: MRT_DEBUG_MODE; value: <X>

	If MRT Generation Mode verification is unsuccessful then the log will contain a message stating: • MRT Generation Mode value other than 1 or 2 the log will contain a message stating: parameter: MRT_DEBUG_MODE; value: <X> • Invalid Mode specified. MRT Generation Mode value is <Null> the log will contain a message stating: • Mode is required If MRT File Name and Location are unspecified in parameter setup the Log will contain the following messages: • Unspecified MRT File Name, defaulting to: <X> • Unspecified MRT File Location, defaulting to: <X> If MRT File Name and Location are specified in parameter setup the Log will contain the following messages: • parameter: MRT_FILE_NM; value: <X> • parameter: MRT_FILE_LOC; value: <X> If parameters are invalid the log will contain a message stating: • Batch MRT Generation : Invalid Batch Parameters If parameters are valid the log will contain a message stating: • MRT Generation has started
2. Selection of Records	Once all parameters are verified each table related to MRT Generation will have messages written to the log for each table being written: • Now starting R_SC_APPL_RSRC_QRY • R_SC_APPL_RSRC_QRY has been written to the MRT File • Now starting R_SC_RSRC_ACCS • R_SC_RSRC_ACCS has been written to the MRT File • Now starting q_CombOrgCode • q_CombOrgCode has been written to the MRT File • Now starting R_SC_RSRC_FLD • R_SC_RSRC_FLD has been written to the MRT File • Now starting R_SC_FLD_ACCS • R_SC_FLD_ACCS has been written to the MRT File • Now starting R_SC_FLD_CMPR • R_SC_FLD_CMPR has been written to the MRT File

	- Now starting R_SC_FGN_ORG - R_SC_FGN_ORG has been written to the MRT File - Now starting R_SC_UI_FLD_SEC - R_SC_UI_FLD_SEC has been written to the MRT File - MRT Generation Complete - Run Ended

- MRT Generation is not re-startable however this job can be is re-run through the process of submitting a new job on demand or part of a cycle.
- The “MRT Generation” job can be configured to be run as often as needed (i.e. multiple times per nightly cycle). The job will regenerate the MRT file and the server will need to be restarted before the related security changes are recognized by subsequent jobstreams.

Major Input Tables

Data Structure	Database Name
Application Resource QueryObject	R_SC_APPL_RSRC_QRY
Access Control (SCRACS) table	R_SC_RSRC_ACCS
Security QueryObject used internally	q_CombOrgCode
Secured Field Resources (SCFUSE) tables	R_SC_RSRC_FLD
Field Access (SCFACS) table	R_SC_FLD_ACCS
Field Comparison	R_SC_FLD_CMPR
Foreign Organization (SCFORG) table	R_SC_FGN_ORG
UI Field Security table	R_SC_UI_FLD_SEC

-

Batch Input

Parameter	Description	Default Value
Generate MRT File Location	This parameter will default to the value from the ADV30Params.ini file during execution if left blank. However, this is an editable parameter and the user can specify a value different from the default if desired, and any user defined value will always take precedence over the default. If a user-supplied value is different than the default then the ADV30Params.ini file will need to be updated before the related MRT file can be used.	Defaults from the “MRTFileLocation=” parameter name in the ADV30Params.ini file
Generate MRT File Name	This parameter will default to the value from the ADV30Params.ini file during	Defaults from the “MRTFileName=”

	execution if left blank. However, this is an editable parameter and the user can specify a value different from the default if desired, and any user defined value will always take precedence over the default. If the user-supplied value is different than the default then the ADV30Params.ini file will need to be updated before the related MRT file can be used.	parameter name in the ADV30Params.ini file
Generate MRT Mode	This is an editable parameter with two possible valid values: • "1" Production: This parameter will run exactly how Production mode runs when one selects the "Generate MRT (Production)" link from the Generate MRT (GMRT) page in the Administration Application. If left blank this parameter will default to a value of "1" from BATSETUP. • "2" Debug: This parameter will run exactly how Debug mode runs when one selects the "Generate MRT (Debug)" link from the Generate MRT (GMRT) page in the Administration Application. • The data file generated by the debug mode is unusable by the application but formatted in an easy to view layout.	Defaults from BATSETUP

Major Output

MRT Generation Production Mode generates the MRT file and the memory image is updated to contain a new snapshot of data generated from the database. The MRT file is then reloaded by restarting the application server.

MRT Generation Debug Mode generates a MRT Flat file that has additional log information, for example the number of tables generated.

Job Return Code

The following table shows the potential job Return Codes for the MRT Generation process.

Return Code	Condition
Successful (1)	This return code is issued if all parameters are verified successfully and all tables are written to the MRT file without error.
Warning (4)	N/A

Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: • Invalid parameters • I/O exceptions • System exception errors When this job ends with a return code of failed the job must be rescheduled as MRT Generation cannot be restarted.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return code of Terminated, subsequent jobs in the chain will be set to Inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain are set to Inactive.

Sort Criteria

N/A

Selection Criteria

All MRT related tables will be processed during the MRT Generation Job regardless of the mode selected.

Problem Resolution

If the job ends with a return code of "Fatal Error" after completing parameter validation, a new job should be scheduled (the failed job cannot be restarted). Because failed jobs parameters are stored in the Journal Log table, the rescheduled job must have valid values for the parameters showing error messages.

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

Step 1: Parameter Validation

Return Code	Condition	Recommendation	Other Instructions
Successful (1)	This return code is issued if all parameters are verified successfully and all tables are written to the MRT file without error.	N/A	N/A
Warning (4)	N/A	This step does not issue this return	N/A

		code.	
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Require Generate MRT Mode is invalid. **Sample message:** Invalid Mode specified.	Schedule a new job after entering a valid mode value: “1” or “2”	N/A
	Required Generate MRT Mode is null. **Sample message:** Mode is required	Schedule a new job after entering a valid mode value: “1” or “2”	N/A
	Required Generate MRT File Location is invalid. **Sample message:** Invalid Batch Parameters	Schedule a new job after entering a valid Generate MRT File Location.	If the default value is used and the job fails one should verify the “MRTFileLocation=” parameter name in the ADV30Params.ini
	Required Generate MRT File Name is invalid. **Sample message:** Invalid Batch Parameters	Schedule a new job after entering a valid Generate MRT File Name.	If the default value is used and the job fails one should verify the “MRTFileName=” parameter name in the ADV30Params.ini
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated before scheduling a new job.	N/A

Step 2: Selection of Records

This step is performed only if the parameter validation is successful.

Return Code	Condition	Recommendation	Other Instructions
Successful (1)	This return code is issued if all parameters are verified successfully and all tables are written to the MRT file without error.	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.	Failure reason needs to be investigated before scheduling a new job.	N/A
	Job failed while writing tables.	Analyze the reason for the failure and restart the job.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated before scheduling a new job.	N/A
	When the job is terminated because of I/O exceptions.	Reason for the System Failure needs to be investigated before scheduling a new job.	N/A
	When the job is terminated because of System exception errors.	Reason for the System Failure needs to be investigated before scheduling a new job.	N/A

2.1.20 Multiple Transaction Submit

Description

The Multiple Transaction Submit is a chain job, which is used to submit all the scheduled transactions during the day. This chain job is a series of batch jobs run as a group, and each batch job calls the System Maintenance Utility to submit the transaction. Different types of jobs can be set up for each type of transaction to be submitted. For each job, the action code and transaction type has to be specified to run the job.

The user can create a new Multiple Transaction Submit chain job. For each batch job in the chain, the action code (defaulted to 162 which is the action code for submit action) and the transaction type has to be specified. The chain job can be submitted to run the multiple transaction submit processes.

The Job return status is set to “Successful” if all the scheduled transactions have been successfully submitted and is set to “Failed” if an exception is encountered.

When to Run

Daily as part of the nightly cycle, and on demand.

Major Input

- Action Code (ACTN_CD)
- Transaction Type (DOC_TYP)

Other Input

None

Output

The scheduled transactions for the types specified in the chain job are submitted

Parameters

Batch Parameters

Job	Parameter	Description	Default Values
Multiple Transaction Submit	ACTN_CD	The code for the action to be performed on the transactions.	162 (Submit)
	DOC_TYP	The type of transactions to be submitted. Each transaction type is specified with value for its type.	

Sort Criteria

None

Selection Criteria

- Select transactions where:

  Transaction Type = DOC_TYP specified as batch parameter and
- For each transaction, perform the action = ACTN_CD specified as another batch parameter, which is submit the transaction.

Troubleshooting

No database restore is required. Correct the problem and rerun the job executing the program. No restoration of datasets or files from backups is required for this program.

2.1.21 Multi Process Approve

Chain or Job Name	Multi Process Approve
Recommended Frequency	On Demand
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	Exception Report

Overview

The Multi Process Approve batch job processes the pending transactions in a Worklist by spawning a number of System Maintenance Utility (SMU) jobs that are configured for performing an action of Approve. The Multi Process Approve job creates a specified number of param files that will have the info of the transactions to be processed. While spawning child SMUs, these param files will be given to the SMUs that will process these records for action of Approve Transaction.

The Multi Process Approve batch job performs the following steps for processing Worklist records:

1. Parameter Validation - The job will validate all the parameters entered by the user. Should any fail validation, the job will end before processing begins.
2. Record Selection - The job will select worklist records on the basis of following job parameters:
   - Approver ID
   - Approval Role
   - Approval Level
   - Transaction Code
   - Transaction Department Code
   - Transaction Last Modified Date

   Note: The primary record selection depends on the Approver ID and Approval Role. Approver ID is used for record selection as well as the User ID to be used for approving the transactions. If the Approval Role is left blank, records are selected based only on **Approver ID**. If the Approval Role is populated, records are selected based only on **Approval Role**.
3. Record Processing & Parameter File Creation - After selecting the records matching selection criteria, the Multi Process Approve batch writes the transaction information in a number of param files specified by the Child Job Count parameter. The records are written to the param files in a Round-Robin fashion, where the param file will be switched after writing a number of records specified by Param Block Size in the current parameter file.
4. Spawn SMU Job(s) - After creating the parameter files, the Multi Process Approve batch spawns a number of SMU jobs specified by the Child Job Count. Each of the SMU jobs are set to perform an action of Approve Transaction. Each child SMU can also be set to create an Exception Report that can be specified by Exception Report Indicator.
5. Await Child Completion - Once all the child jobs are spawned, the Multi Process Approve batch waits for them to complete.

6. Concatenate Exception Reports - If the Multi Process Approve batch is configured to create an Exception Report by specifying the type of report in Exception Report Indicator, a summarized report named by job parameter Exception Report File Name is created. This file will contain the info written by each child SMU in their own Exception Reports while processing the records.

Process Steps	Messages
1. Parameter Validation	• Validating job parameters. • If the parameter is invalid, the invalid value is displayed in the log. • If validation was successful: o Parameter validation successful. Otherwise o Parameter validation failed, stating parameter error.
2. Record Selection	• Selecting eligible records. • Selection completed.
3. Record Processing / Parameter File Creation	• Started processing Worklist records. • If records were processed, o Total records processed <Record Count>. o Finished processing records. o Param files for child jobs created. • If no records were processed, o No eligible records found • If number of child jobs created is less than the Child Job Count specified, o Number of child jobs will be less than number of child jobs specified.
4. Spawn SMU Job(s)	• Setting up child jobs. • Scheduling job with ID = <Job ID> • Enabled job with ID = <Job ID>
5. Await Child Completion	• Waiting for child jobs to complete. • Child job pending count:<active job count> • Child jobs completed.
6. Concatenate Exception Reports	• Summarizing exception reports. • Summarization of exception reports completed.

The Multi Process Approve batch job is restartable. The job can be restarted for any step from Record Processing/Parameter File Creation onward. The job will keep saving the checkpoint information during various steps in its run:

- Record Processing/Parameter File Creation:

  During this step, the Multi Process Approve batch selects Worklist records and creates a number of param files. The following information is saved in the checkpoint during this step:

  Checkpoint Step: PRCS_RECS

  Param File Offset Array

  Last processed Transaction Information

  Next index number in the parameter file array

  Number of records processed till now

  On restart, the job will re-start writing records to the param file it left off with.

- Spawn SMU Job(s):

  During this step, the Multi Process Approve batch spawns the number of SMU jobs configured through the param file created in the previous step for action Approve Transaction. The following information is saved in the checkpoint during this step:

  Checkpoint Step: SETUP_CHILD

  Job ID Array

  Next Index in the SMU job ID array

  Total number of records processed by Multi Process Approve batch

  On restart, the job creates the leftover number of child jobs.

- Await Child Completion:

  During this step, the Multi Process Approve batch waits for all the child SMU jobs to complete the processing. The job will thus keep on checking the completion status of the child jobs after an interval specified by Stagger Time. The following information is saved in the checkpoint during this step:

  Checkpoint Step: AWAIT_CHILD

  Job ID Array

  Number of child jobs

  Total number of records processed by Multi Process Approve

  On restart, the job checks the status of the child jobs and restarts the child SMUs if required.

- Concatenate Exception Reports:

  This is the last step of the job processing. If the Multi Process Approve job has been configured to create an Exception Report, the batch concatenates all the Exception Reports created by the child SMUs. The following information is saved in the checkpoint during this step:

  Checkpoint Step: CONCAT_REP

  Job ID Array

  Total number of records processed by Multi Process Approve

On restart, the job concatenates the reports from the beginning.

- Completion:

  Once the job has finished all the specified steps, it will commit the checkpoint information with Checkpoint Step: CMPL.

Major Input

- Worklist (WF_APRV_WRK_LST)
- Transaction Header (DOC_HDR)
- User Info (R_SC_USER_INFO)
- Approver Role (R_WK_USER_ROLE)

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
AMSPARM	Required. Param File Directory	$$AMSPARM$$
AMSEXPORT	Required. Export/Import Directory	$$AMSEXPORT$$
DOC_CD	Optional. Transaction Code (comma separated list)	No Default Considers * and empty input as wildcard
DOC_DEPT_CD	Optional. Transaction Department Code (comma separated list)	No Default Considers * and empty input as wildcard
APRV_ID	Required. Approver ID	No Default
APRV_ROLE	Optional. Approval Role (comma separated list)	No Default
APRV_LVL	Required. Approval Level	No Default
FRM_DT	Optional. From Date (mm/dd/yyyy)	No Default
TO_DT	Optional. To Date (mm/dd/yyyy)	No Default
THREAD_CT	Required. Child Thread Count. If left blank a value of 1 will be assumed.	4
EXCEP_REP_FILE_NM	Conditionally Required if Exception Report Indicator is 1 - 5. Exception Report File Name (without file extension)	AprvExcpRep
EXCEP_REP_IND	Optional. Exception Report Indicator (1-Detailed, 2-Failed, 3- Processed, 4-	1

	Failed Transaction Lines, 5-Transaction Status). If left blank, no exception report will be produced.	
PARM_BLK_SIZE	Optional. Param Block Size. If left blank a value of 100 will be assumed.	100
SEL_BLK_SIZE	Optional. Select Block Size. If left blank a value of 100 will be assumed.	100
APPLY_OVERRIDES	Optional. Apply Overrides (true/false). If left blank a value of 'false' is assumed.	True
STAGGER_TIME	Optional. Stagger Time (seconds). If left blank a value of 30 is assumed.	30
SMU_CTLG_ID	Required. SMU Catalog ID for current application.	3

Major Output

- Worklist & Transaction Header - Transactions approved at the specified level through various spawned SMU threads.
- Exception report for the processed transactions.

Job Return Code

Return Code	Condition
Successful (1)	All the selected records are processed successfully and all child SMU jobs were Successful.
Warning (4)	No eligible records found. This could be because of the following reasons: • No pending records found on the Worklist eligible to be selected by the selection criteria.
Non Fatal Error (8)	Any of the child SMU jobs complete with return code of Non Fatal Error.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid. • Run time exceptions for unexpected situations. • A child SMU job failed 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 job selects the records sorted based on following attributes:

- Transaction Code
- Transaction Department Code
- Transaction ID
- Transaction Version Number

Selection Criteria

The job selects the records based on following:

- Approver ID
- Approval Role (if specified)
- Approval Level
- Transaction Code
- Transaction Department Code
- Transaction Last Modified Date (if specified)

Problem Resolution

1. Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the parameters are validated successfully.	N/A	N/A
Warning (4)	Job step does not return with this code	N/A	N/A
Non Fatal Error (8)	Job step does not return with this code	N/A	
Failed (12)	Job failed due to Fatal conditions or invalid parameters.	In this step, the job can fail under the following two conditions. 1) Encounters any runtime exceptions and 2) Invalid batch Parameters If the job fails because of the runtime exceptions, investigate the exception reported by the process	N/A

		before starting a new job.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. The job cannot be restarted at this step. A new job should be scheduled.	N/A
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the System Failure needs to be investigated. The job cannot be restarted at this step. A new job should be scheduled.	N/A

2. Record Selection

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the records are successfully selected.	N/A	N/A
Warning (4)	Job step does not return with this code	N/A	N/A
Non Fatal Error (8)	Job step does not return with this code	N/A	N/A
Failed (12)	Job failed due to Fatal conditions	In this step, the job can fail under the following two conditions. 1) Encounters any runtime exceptions and 2) Failed during restart. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only new job should be scheduled.
	Failed while restarting the job since another instance of the job has already been run successfully.	**Recommendation:** Schedule a new job.	N/A

Terminated (16)	Job is terminated manually by the user.	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 new job should be scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only new job should be scheduled.

3. Record Processing & Parameter File Creations

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the records are successfully written to the param file.	N/A	N/A
Warning (4)	None of the selected records were eligible for writing to param file.	Run the job if there are eligible records available in worklist to be selected.	Alternatively the job can be rescheduled with a different set of parameters.
Non Fatal Error (8)	Job step does not return with this code	N/A	N/A
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following two conditions. 1) Encounters any runtime exceptions and 2) Failed during restart. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only new job should be scheduled.
	Failed while restarting the job since another instance of the job	**Recommendation:** Schedule a new job.	

	has already been run successfully.		
Terminated (16)	Job is terminated manually by the user.	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 new job should be scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only new job should be scheduled.

4. Spawn SMU Job(s)

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the child SMU jobs are successfully spawned.	N/A	N/A
Warning (4)	Job step does not return with this code	N/A	N/A
Non Fatal Error (8)	Job step does not return with this code	N/A	N/A
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following two conditions. 1) Encounters any runtime exceptions and 2) Failed during restart. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	If any of the child jobs had already started processing records, the parent job needs to be restarted so that the final report contains the records processed by that SMU.

	Failed while restarting the job since another instance of the job has already been run successfully.	**Recommendation:** Schedule a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	If any of the child jobs had already started processing records, the parent job needs to be restarted so that the final report contains the records processed by that SMU.
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	If any of the child jobs had already started processing records, the parent job needs to be restarted so that the final report contains the records processed by that SMU.

5. Await Child Completion

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the child jobs successfully completed.	N/A	N/A
Warning (4)	Job step does not return with this code	N/A	N/A
Non Fatal Error (8)	This return code will be issued when the Max Line Limit validation fails in this step.	The job would have skipped those records that have failed in the Max Line Limit validation. In order to process payments for those records, before scheduling the new job the settings on the Transaction Component Requirements Table for the Transaction Type AD needs to be changed appropriately.	N/A

Failed (12)	Job failed due to Fatal conditions or if any of the child jobs failed.	In this step, the job can fail under the following two conditions. 1) Encounters any runtime exceptions and 2) Failed during restart. 3) Any of the child jobs fail. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	If any of the child jobs failed, investigate the reason for the failure. The child job will be restarted automatically if the parent job is restarted.
	Failed while restarting the job.	**Recommendation:** Try restarting the job again as the report generated by the child jobs needs to be used to create a final report.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and run successfully, then restarting this job will overwrite the report created by the other job. The previous report needs to be backed up before restarting this job.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and run successfully, then restarting this job will overwrite the report created by the other job. The previous report needs to be backed up before restarting this job.

6. Concatenate Exception Reports

Possible Return Codes	Condition	Recommendation	Other Instructions

Successful (1)	Exception report is generated successfully by concatenating exception reports from all the child jobs.	N/A	N/A
Warning (4)	Job step does not return with this code	N/A	N/A
Non Fatal Error (8)	Job step does not return with this code	N/A	N/A
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following two conditions. 1) Encounters any runtime exceptions and 2) Failed during restart. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	If another instance of the job has already been scheduled and run successfully, then restarting this job will overwrite the report created by the other job. The previous report needs to be backed up before restarting this job.
	Failed while restarting the job since another instance of the job has already been run successfully.	**Recommendation:** Try restarting the job again as the report generated by the child jobs needs to be used to create a final report.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and run successfully, then restarting this job will overwrite the report created by the other job. The previous report thus needs to be backed up before restarting this job.
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and run successfully, then restarting this job will overwrite the report created by the other

			job. The previous report thus needs to be backed up before restarting this job.

2.1.22 Multi Process Submit

Job Name	Multi Process Submit
Recommended Frequency	On Demand
Single Instance Required	No
Can be restarted?	Yes
Reports generated	None

Overview

The Multi Process Submit job selects draft transactions based on the selection criteria and spawns the System Maintenance Utility (SMU) jobs to submit them. The Multi Process Submit job creates a specified number of parameter files that distribute and identify the transactions to be processed. Each spawned child SMU job is submitted with a corresponding parameter file to process the assigned transactions using the DOCSUBMIT action.

Process Steps	Messages
1. Parameter Validation	• Batch parameter validation started. • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value is displayed in the log. • Batch parameter validation completed.
2. Selection of Records	• Selection Criteria: <selection criteria> • Sort Order: <sort order> • If the selection returns 0 records, then the following message is displayed: “No records found to process”. • Number of records (count) selected is displayed.
3. Record Processing/Parameter File Creation	• Message indicating progress of processing is displayed: “Number of records written to parameter file = <count>” • Created parameter files successfully.
4. Spawn SMU Job(s)	• “SMU Job - <job ID> - Spawned” is issued for each spawned child SMU Submit job. • “Restarted SMU Job - <job ID>” is issued for each restarted spawned child SMU Submit job.
5. Await Child Completion	• N/A
6. Concatenate Exception Files	• Exception reports created successfully.
7. Record stats	• “Record Stats for job < job ID > - Records Submitted: Processed<Count>, Failed <Count>, Succeeded <Count>” is issued for each spawned child SMU Submit job. • “Total number of records processed for load <Count>,

	Failed <Count>, Succeeded <Count>” • “Record Stats for Resubmit job < job ID > - Records Submitted: Processed<Count>, Failed <Count>, Succeeded <Count>” is issued for each restarted spawned child SMU Submit job. • “Total number of records processed for Resubmit <Count>, Failed <Count>, Succeeded <Count>” is issued for restarted job.

Steps in Running this Process

1. Parameter Validation:

   The job validates the job parameters. If parameter validation fails, the job ends.

2. Selection of Records:

   The job selects draft transactions based on criteria specified in the job parameters.

3. Record Processing/Parameter File Creation:

   During this step, the Multi Process Submit job selects transaction records and creates number of parameter files.

4. Spawn SMU Job(s):

   During this step, the Multi Process Submit job spawns the number of SMU jobs configured through the parameter files created in the previous step for the action Submit Transaction.

   Once this step is completed, checkpoint is created which stores the comma separated job IDs as a check point which will be used for restarting the spawned SMU Submit Transactions jobs if any of them did not complete with Return Code of Successful.

5. Await Child Completion:

   During this step, the Multi Process Submit job waits for all the child SMU jobs to complete the processing. The job thus keeps on checking the completion status of the child jobs.

6. Concatenate Exception Files:

   This is the last step of the job processing. If the Multi Process Submit job is configured to create an Exception Report file, the batch concatenates all the Exception Files created by the child SMUs.

Major Input

- Transaction Header Catalog (DOC_HDR)

Batch Parameters

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

Parameter	Description	Default Value
Transaction Create User ID (DOC_CREA_USID)	Optional. If specified, it is used to find transactions with a	(No Default)

	matching create User ID during record selection.	
Transaction Last Date (DOC_LAST_DT)	Optional. If specified, it is used to find transactions last modified on or after this date during record selection. Valid format is MM/DD/YYYY.	(No Default)
Exception Report File Name (EXCEP_REP_FILE_NM)	Optional. If specified, output exception report files are produced using this value to form the file names.	(No Default)
Exception Report Indicator (EXCEP_REP_IND)	1 = Detailed, 2 = Failed_Transactions, 3 = Processed_Transactions, 4 = Failed_Lines 2 is used if the value is blank.	(No Default)
Transaction Status Code (EXPORT_DOC_STA_CD)	Optional. Parameter name (prefix EXPORT_ removed) and values are written in the parameter file used by child SMU jobs.	(No Default)
Prefix for Parameter Files (FILE_PREFIX)	Optional. If specified, the parameter file names are created using the prefix.	(No Default)
Selection Criteria (SELECT_CLAUSE)	Optional. Use SQL to specify how to select transactions for submission. Note that the job looks for transactions in draft phase in addition to the specified selection criteria.	(No Default)
SMU Batch Catalog ID (SMU_CTLG_ID)	Optional. Indicates the Catalog ID corresponding to the SMU job. 3 is used if value is blank.	3
Lag time (STAGGER_TIME)	Optional. Number of seconds between the spawning of each child process. 30 is used if value is blank.	10
Number of Jobs to Start (THREAD_COUNT)	Optional. Determines how many child SMU Submit jobs to spawn. 1 is used if value is blank.	2
Polling frequency (SLEEP_INTERVAL)	Optional. Time interval in milliseconds between checking the completion status of child jobs. 5000 is used if value is blank.	5000
Maximum prefetch count	Optional. If specified, it is used	2000

(MAX_PREFETCH_COUNT)	for Prefetch count. 2000 is used if value is blank.	
Progression counter size (PROG_CTR_SZ)	Optional. If specified, it is used for Progression message counter. 1000 is used if value is blank.	100
Listener Class Name (LISTENER_NAME)	Optional. This is the name of the Java class implementing the AMSSysManUtilListener interface or any interface extending AMSSysManUtilListener. If the class name is provided, then the value is added as input to the parameter file created, which is used as input to be spawned child SMU jobs. Syntax for providing the listener class name: <Package Name>.<Class Name>.	No Default

You can introduce any new parameter if required by child SMU jobs. To do so, you need to find the Multi Process Submit job on the Job Setup (BATSETUP) page, and then add the new parameter with a prefix of EXPORT_. Adding a new parameter does not involve any code changes but this parameter cannot be validated.

The parameter introduced with a prefix of EXPORT_ is used only to write the parameter name and its value to the Parameter file used by child SMU jobs. Note that the prefix EXPORT_ is removed while writing the parameter name to the parameter file.

For example, the EXPORT_DOC_STA_CD parameter is written to the parameter file as DOC_STA_CD = (value).

Major Output

- Spawned child SMU jobs - each with a parameter file specifying which transactions to submit.
- Transactions submitted by child SMU jobs.
- Exception report files (they can be used by other custom jobs as input to generate the desired report output).

Sort Criteria

Selected transactions are sorted by DOC_CAT, DOC_TYP, DOC_CD, DOC_DEPT_CD, DOC_ID, and DOC_VERS_NO.

Selection Criteria

- Transaction Phase = Draft
- If Transaction Create User ID job parameter is specified, match Created By user using case-insensitive comparison.

- If Transaction Last Date job parameter is specified, narrow down to transactions last modified on or after this date.
- If Selection Criteria job parameter is specified, also incorporate the specified conditions during record selection.

Problem Resolution

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

Step 1: Parameter Validation

Return Code	Condition
Successful (1)	Parameter validation is successful.
Warning (4)	N/A
Non-Fatal Error (8)	N/A
Failed (12)	Parameters are invalid.
Terminated (16)	The job is terminated by the user.
System Failure (20)	The job is terminated because of database server or network issues.

Step 2: Selection of Records:

Return Code	Condition
Successful (1)	One or more records were selected.
Warning (4)	No records matched selection criteria.
Non-Fatal Error (8)	N/A
Failed (12)	Selection criteria is not valid.
Terminated (16)	The job is terminated by the user.
System Failure (20)	The job is terminated because of database server or network issues.

Step 3: Record Processing/Parameter File Creation

Return Code	Condition
Successful (1)	All the records are written successfully to parameter file(s).
Warning (4)	N/A
Non-Fatal Error (8)	N/A
Failed (12)	Error while creating parameter files.
Terminated (16)	The job is terminated by the user.
System Failure (20)	The job is terminated because of database server or network issues.

Step 4: Spawn SMU Job(s)

Return Code	Condition
Successful (1)	Child SMU jobs spawn successfully.
Warning (4)	N/A
Non-Fatal Error (8)	N/A
Failed (12)	Error while spawning the child jobs.
Terminated (16)	The job is terminated by the user.
System Failure (20)	The job is terminated because of database server or network issues.

Step 5: Await Child Completion

Return Code	Condition
Successful (1)	Child SMU jobs complete successfully.
Warning (4)	The highest return code of child SMU jobs is Warning.
Non-Fatal Error (8)	The highest return code of child SMU jobs is Non-Fatal Error.
Failed (12)	The highest return code of child SMU jobs is Failed. Error while retrieving the status of child SMU jobs.
Terminated (16)	The job is terminated by the user.
System Failure (20)	The job is terminated because of database server or network issues.

Step 6: Concatenate Exception Files

Return Code	Condition
Successful (1)	Exception files are concatenated successfully.
Warning (4)	N/A
Non-Fatal Error (8)	N/A
Failed (12)	Error while concatenating exception files. Error while deleting temporary exception files.
Terminated (16)	The job is terminated by the user.
System Failure (20)	The job is terminated because of database server or network issues.

2.1.23 Multi-threaded Table Loader

Chain or Job Name	Multi-threaded Table Loader
Recommended Frequency	On Demand
Single Instance Required	No
Can be restarted?	No
Reports generated	No

Overview

The Multi-threaded Table Loader splits large Advantage table files into smaller sized table files. The application accepts only XML file as input, and does the process of splitting the input file. Splitting is based on following input parameters:

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

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

Unlike, the Multi-threaded Transaction Loader which can be run in three different modes, the Multi-threaded Table Loader can only be executed in one mode which splits and imports the related records.

Process Details

Logic:

STEP-1 Parameter Validation:

The job reads all related parameters, including the name of the related input XML file (containing table records), performs parameter validation, and if any parameters are invalid then the job terminates and issues related error messages regarding invalid parameters to the job log.

STEP-2 Splitting the Input File:

In this step the job will split the input file in to multiple smaller files. If the count of split files is greater than zero the following message will be logged in the job log: "File split done: count=<number>". If the count is not greater than zero then the job will terminate and issue an error stating "No records found for processing" within the job log.

STEP-3 Child Job processing:

As each child job is spawned the message "Setting up child jobs for processing" will be written to the job log as well as the message "Number of jobs to set :< number>". Therefore the user is informed of the number of jobs spawned (one job per file after the split).

STEP-4 Status of Child Job scheduled for processing:

The job will log the status of each child job scheduled for the processing as well as the status of each job once child job execution is complete.

STEP-5 Deletion of temporary Files created during processing:

Once the job completes successfully all temporary files created as a result of this job execution (i.e. each split input file) will be deleted by the job.

Job Not Restartable

If the process fails for any reason schedule a new instance of the Multi-threaded Table Loader job and submit the new job. Before doing so you can modify the original input file to not include those records already successfully imported during the failed run, or the new job may be executed against the same input file without modifications. If the input file is not modified a non-fatal error will be thrown for every record successfully imported during the failed run, however due to the fact that it is a non-fatal error the job will continue and will load all records not loaded during the failed run, resulting in a successful import overall.

Major Input

The input for this job would be the xml file that is to be loaded. The input files are specified in the Job Parameter.

Major Output

- The result of this job is the loading of Table records from the input file. Files are first split and then imported.
- After successful execution the jobstream will delete all temporary files created during execution.
- A parameter file is generated for each split XML file, and then the spawned jobs read those input parameter files in a sequential order.

Batch Parameters

Parameter Name	Internal Parameter Name	Description	Default Value
Block Size	BLOCK_SIZE	Number of records in each split segment of the input file. This parameter is not required.	1
Commit Block Size	COMMIT_BLOCK_SIZE	Number of records to commit at a time. This	The value for the property –

		parameter is not required.	XMLExportCommitBlockSize – in ADV30Params.ini
File Input Directory	FILE_INPUT_DIR	The location of the input file. This parameter is not required.	The value for the property – XMLExportFileLocation– in ADV30Params.ini
File List	FILE_LIST	Comma separated list of files to be uploaded in multi-threaded processing. This parameter is required.	None
File Output Directory	FILE_OUTPUT_DIR	Output location for the file segments. This parameter is not required.	The value for the property – XMLExportFileLocation– in ADV30Params.ini
File Prefix	FILE_PREFIX	Prefix used on the filenames for the output file segments. This parameter is not required.	None (blank string)
Log Status Interval	LOG_STATUS_INTERVAL	Logging frequency (in seconds) for controller thread reporting status of child threads to the system log. This parameter is not required.	300 Seconds (5 minutes). Cannot be less than 30 seconds. The system will use the default of 300 if a value less than 30 is specified.
Sleep Interval	SLEEP_INTERVAL	Polling frequency (in seconds) for internal controller thread for checking child processes. This parameter is not required.	5 seconds. A value less than 5 seconds is not used (default value used instead)
System Maintenance Utility Catalog ID	SMU_CTLG_ID	Catalog id of the System Maintenance Utility job which is spawned as the child process. This parameter is required.	3
Stagger Time	STAGGER_TIME	The lag time, in seconds, between the spawning of each child process. This parameter is not required.	30 seconds. Valid values >= 1 second- Default value used otherwise
Thread Count	THREAD_COUNT	Number of threads to use for processing. This parameter is not required.	1

Additional Job Parameters

For table import, optional parameters like MAX_ERRORS, ERROR_FILE_NM and EDITS_FL can be added for child jobs (System Maintenance Utility). These Additional Job Parameters that are specific to the children jobs are optional and hence not supplied as part of day zero setup on the Batch catalog. Use of these optional parameters is left to the client's discretion. If the client requires adding these additional parameters they would have to go to Batch catalog (BATSETUP) for the Multi-threaded Table Loader and add the desired parameter(s).

When these parameters are added they need to be prefixed with **I_SMU_.**

When they are passed to the children jobs, the prefix itself will be taken away and only the job parameter portion will be passed. For example, setting the following job parameters for the Multi-threaded Table Loader job:

```
I_SMU_EDITS_FL=1
I_SMU_MAX_ERRORS=10
```

will generate the job parameter file of the child jobs (System Maintenance Utility) as shown here:

```
**ACTN_CD= TBLIMPORT
_PARAM_LINE_
FILE_NM=/apps/advhr/3x/AMSADV35/RTFiles/ah351os2/ExportImport/XML_62100_3.xml
COMMIT_BLOCK_SIZE=10
EDITS_FL =1
MAX_ERRORS=10
```

Refer to the description of System Maintenance Utility for more details on these job parameters for the child jobs.

Job Return Code

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

Return Code	Condition
Successful (1)	All of the table records are imported successfully.
Warning (4)	Not applicable for this job
Non-Fatal (8)	This job will issue a non-fatal error for the following: • Error while importing the input record files during the child job processes (System Maintenance Utility). For example, if attempting to import a record that already exists in the target table.

Failed (12)	The job will fail under the following conditions: • Invalid Parameters • Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

There is not any sort criteria.

Selection Criteria

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

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non-Fatal (8)	Job Failed due to Non-fatal condition: Duplicate record being imported to target table	No action is needed. The job will continue and will import all other records (assuming they do not exist on the target table).	N/A
Failed (12)	Job failed due to Fatal conditions. Sample Message: "No input file to process"	In this step, the job can fail under the following conditions: 1) Job Parameter error 2) Encounters any runtime exceptions and 3) Failed during child job. If the Job Parameter input was causing the error, examine the error message	N/A

		in the job log, correct the job parameter. You must then reschedule the job. If the job fails because of runtime exceptions, investigate the exception reported by the process, resolve the error. You must reschedule the job. For child job errors, refer to the run-sheet for the System Maintenance Utility job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. When the reason for termination has been determined, you must reschedule the 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. When the reason for System Failure has been determined, you must reschedule the job.	N/A

2.1.24 Multi-threaded Transaction Loader

Chain or Job Name	Multi-threaded Transaction Loader
Recommended Frequency	On Demand
Single Instance Required	No
Can be restarted?	Yes
Reports generated	Yes. If any of the child jobs caused an exception, it will be merged in the Exception Report file for the job (specified in the Job Parameter).

Overview

The Multi-threaded Transaction Loader splits large Advantage Transaction files into smaller sized Transaction files. The application accepts only XML file as input, and does the process of splitting the input file. Splitting is based on following input parameters:

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

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

Process Details

The Multi-threaded Transaction Loader job can be executed in four process modes. Splitting the files is a common functionality across all four process modes. These process modes differ only in the action taken after splitting the files.

Process mode 1: Import

The Multi-threaded Transaction Loader can be used to split and import the transaction file into the Advantage system.

Process mode 2: Import & Submit

The Multi-threaded Transaction Loader in addition to steps performed in process mode 1, sets up the transactions for Submit action after import.

Process mode 3: Import & Transaction Other Action Mode

The Multi-threaded Transaction Loader in addition to steps performed in process mode 1, sets up the transactions for any other action after import. The 'Other Action' to be performed is specified as an input parameter to the Multi-Threaded Transaction Loader.

Process mode 4: Import & Validate

The Multi-threaded Transaction Loader in addition to steps performed in process mode 4, sets up the transactions for Validate action after import.

Logic:

STEP-1 Parameter Validation:

The job reads all related parameters, including the name of the related input XML file (containing transaction details), performs parameter validation, and if any parameters are invalid then the job terminates and issues related error messages regarding invalid parameters to the job log.

STEP-2 Splitting the Input File:

In this step the job will split the input file in to multiple smaller files. If the count of split files is greater than zero the following message will be logged in the job log: "File split done: count=<number>". If the count is not greater than zero then the job will terminate and issue an error stating "No records found for processing" within the job log.

STEP-3 Child Job processing:

As each child job is spawned the message "Setting up child jobs for processing" will be written to the job log as well as the message "Number of jobs to set :< number>". Therefore the user is informed of the number of jobs spawned (one job per file after the split).

STEP-4 Status of Child Job scheduled for processing:

The job will log the status of each child job scheduled for the processing as well as the status of each job once child job execution is complete.

STEP-5 Exception Report:

If there are any exceptions while loading the records the job will log a related exception message to the exception file if an exception file name has been provided in the job parameters (S_SMU_EXCEP_REP_FILE_NM). All exception files created by the individual child jobs are concatenated into one exception file (named as the exception file provided in the parameters) once all child jobs have completed.

STEP-6 Deletion of temporary Files created during the processing:

Once the job completes successfully all temporary files created as a result of this job execution (i.e. each split input file and all child exception files) will be deleted by the job.

Restartability Information

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

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

Major Input

The input for this job would be the xml file that is to be loaded. The input files are specified in the Job Parameter.

Major Output

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

Job parameters

Parameter Name	Internal Parameter Name	Description	Default Value
Block Size	BLOCK_SIZE	Number of records in each split segment of the input file. This parameter is not required.	1
Commit Block Size	COMMIT_BLOCK_SIZE	Number of records to commit at a time. This parameter is not required.	The value for the property – XMLExportCommitBlockSize – in ADV30Params.ini
Transaction	S_SMU_DOC_ST	This parameter allows the user to specify the	None

Status	A_CD	Transaction Status (1-Held or 2-Ready) for the loaded transactions. This parameter is not required.	
File Input Directory	FILE_INPUT_DIR	The location of the source file for the input records. This parameter is not required.	The value for the property – XMLExportFileLocation– in ADV30Params.ini
File List	FILE_LIST	Comma separated list of files to be uploaded in multi-threaded processing. This parameter is required.	None
File Output Directory	FILE_OUTPUT_DIR	Output location for the file segments. This parameter is not required.	The value for the property – XMLExportFileLocation– in ADV30Params.ini
File Prefix	FILE_PREFIX	Prefix used on the filenames for the output file segments. This parameter is not required.	None (blank string)
Log Status Interval	LOG_STATUS_INTERVAL	Logging frequency (in seconds) for controller thread reporting status of child threads to the system log. This parameter is not required.	300 Seconds (5 minutes). Cannot be less than 30 seconds. The system will use the default of 300 if a value less than 30 is specified.
Mode	MODE	Mode of operation. (1=Import, 2=Import and Submit, 3=Import and Other Action, 4=Import and Validate). This parameter is required.	No Default
Other Action	OTHER_ACTION	An 'action' considered valid for the System Maintenance Utility. This parameter is not required, unless MODE = 3.	No Default
Sleep Interval	SLEEP_INTERVAL	Polling frequency (in seconds) for internal controller thread for checking child processes. This parameter is not required.	5 seconds. A value less than 5 seconds is not used (default value used instead)
System Maintenance Utility	SMU_CTLG_ID	Catalog id of the System Maintenance Utility job which is spawned as the child process. This parameter is	No Default for this parameter.

Catalog ID		required.	
Stagger Time	STAGGER_TIME	The lag time, in seconds, between the spawning of each child process. This parameter is not required.	30 seconds. Valid values >= 1 second- Default value used otherwise
System Maintenance Utility Exception File for Import and Submit Mode	S_SMU_EXCEP_REP_FILE_NM	Name of exception file for System Maintenance Utility in Import and Submit Mode. This parameter is not required.	None
System Maintenance Utility Flag to indicate if failed error messages should be recorded in Import Mode	I_SMU_DTL_IMP_MSG_FL	Indicates whether error message details for each failed record should be recorded in the error file in Import Mode	True
Thread Count	THREAD_COUNT	Number of threads to use for processing. This parameter is not required.	1
File Polling Interval(Seconds)	FILE_POLLING_INTERVAL	The time interval taken (in seconds) by this batch job after every polling is committed to the Export-Import folder looking for the input file.	0
File Polling Max Time(Seconds)	FILE_POLLING_MAX_TIME	The maximum time (in seconds) until when the batch process will keep polling to the Export-Import folder looking for the input file.	0

Additional Job Parameters

- For Transaction import, optional parameters like MAX_ERRORS, ERROR_FILE_NM and EDITS_FL can be added for child jobs (System Maintenance Utility). These Additional Job Parameters that are specific to the children jobs are optional and hence not supplied as part of day zero setup on the Batch catalog. Use of these optional parameters is left to the client's discretion. If the client requires adding these additional parameters they would have to go to Batch catalog (BATSETUP) for the Multi-threaded Transaction Loader and add the desired parameter(s).
- When these parameters are added they need to be prefixed as follows:
  - **I_SMU_**: Parameters passed to SMU jobs in Import Mode.
  - **S_SMU_**: Parameters passed to SMU jobs in Import and Submit Mode.
  - **O_SMU_**: Parameters passed to SMU jobs in Import and Other Action Mode.

When they are passed to the child jobs, the prefix itself will be taken away and only the job parameter portion will be passed. For example, setting the following job parameters for the Multi-threaded Transaction Loader job:

```
MODE=1
I_SMU_DOC_CD=ADDR
I_SMU_DOC_DEPT_CD=001
```

will generate the job parameter file of the child jobs (System Maintenance Utility) as shown here:

```
**ACTN_CD=DocImport
_PARAM_LINE_
FILE_NM=/apps/advhr/3x/AMSADV35/RTFiles/ah351os2/ExportImport/XML_62100_3.xml
COMMIT_BLOCK_SIZE=10
DOC_CD=ADDR
DOC_DEPT_CD=001
```

Refer to the description of System Maintenance Utility for more details on these job parameters for the child jobs.

Job Return Code

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

Return Code	Condition
Successful (1)	All of the Transaction records are imported successfully.
Warning (4)	N/A
Non-Fatal (8)	This job will issue a non-fatal error for the following: ▪ Error while importing records that already exist in the application.
Failed (12)	The job will fail under the following conditions: • Invalid Parameters • Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

There is not any sort criteria.

Selection Criteria

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

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non-Fatal (8)	Job Failed due to Non-fatal condition: Duplicate record being imported to application	No action is needed. The job will continue and will import all other records (assuming they do not exist within the application).	N/A
Failed (12)	Job failed due to Fatal conditions. Sample Message: "Invalid value received for parameter MODE (expected values 1,2 or 3): aBcD" "No input file to process"	In this step, the job can fail under the following conditions: 1) Job Parameter error 2) Encounters any runtime exceptions and 3) Failed during child job. If the Job Parameter input was causing the error, examine the error message in the job log, correct the job parameter, then restart the job. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job. For child job errors, refer to the runsheet for the System Maintenance Utility job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be	If another instance of the

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

2.1.25 New Year Initialization Table Process

Chain or Job Name	New Year Table Initialization (NYTI) Process
Recommended Frequency	Yearly
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes

Overview

When a new year approaches the need to populate many reference tables driven by Fiscal Year or Budget Fiscal Year exists to enable transaction entry in the new year. Data already setup in the current year is the best source for data in the new year. Any minor modifications required can be manually made after an automatic creation of records by this, the NYTI Process chain job.

As many years are populated over time, the system accumulates data in past years that is no longer used by the application. The NYTI Process is designed with a delete mode that will archive off this older data in a fashion similar to how it populates data for a new year.

Not all tables that are keyed by Fiscal or Budget Fiscal Year are included in NYTI. Maintenance of these tables should not or cannot be done by the chain.

- Cannot Example: **Required Budget (REQ_BUD / REQBUD)** is a table that cannot be rolled forward from one year to the next because of a unique table structure. Records here must be manually added for the new year. Please ensure this table is not in any custom NYTI parameter file or the result will be that the source BFY records will be overlaid with the target BFY records, no matter what the Table Overlay parameter setting is.
- Should Not Example: The Appropriation Inference table (INF_APPR). It is loaded by the creation of budget lines on certain budget structures. For NYTI to populate this table would be wrong, thus it is not listed in the tables to populate. Additionally, it is not listed in the tables to purge. For this reason, purging of this table should be done outside of the NYTI program for a year to be deleted before running NYTI. If not, relationships on that table will cause certain records to not be deleted by NYTI.

The chain consists of four jobs.

NYTI Process **–** In populate mode, this job step reads the input parameter file for a listing of tables in a specific order. From this listing, the job step creates an XML file for each table with records to be added. In delete mode, this job step creates an inverted parameter file for later job steps.

Export/Import Tables **–** In populate mode, this System Maintenance Utility job imports each of the XML files created from the first job step. In delete mode, this job step creates XML files for the next job to delete records.

Delete Records **–** This job step deletes the records in the XML file from each table in the reverse order found in the parameter file. This job runs even in populate mode but does no processing.

NYTI Report **– This job step produces a report, listing by table, the number of records selected, successfully processed, and failed to process along with system messages.**

No dependencies on other batch jobs exist before the running of NYTI.

One manual pre-processing step would be for users to update records in the current year for an inactive status or Effective From and To Dates before running NYTI. Also, a review should be done to see if any records can be deleted in the current year that were not used before copying those into a new year.

A technical pre-processing step would be to pull any tables out of cache that are to be populated as part of NYTI.

When running in delete mode, the APPCTRL parameter "COA Deletion Prevention" must be set to 3 (Usage verification disabled so deletion is allowed).

There are several job dependencies after the running of NYTI.

Run the **Export From Advantage** job under Vendor Self Service in the Batch Catalog if the VSS application is used.

As this program updates the organizational chart-of-account tables, which are used in security, it is important to regenerate the MRT (memory resident tables) after population of new year data with the MRT Generation job under Administration/Utilities in the Batch Catalog.

Certain budget tables are also updated that are in memory. For these and the new MRT file, a bounce of the application is required for the NYTI changes to take effect.

When running in delete mode, the APPCTRL parameter "COA Deletion Prevention" should be restored if it was set to a value other than 3 (Usage verification disabled so deletion is allowed) prior to running NYTI.

Major Input

- NYTI_TABLE_PARM.txt
- Reference tables listed in above parameter file

Major Output

- XML files created for each reference table
- Reference tables listed in parameter file
- Report

Chain / Job Return Code

For chain jobs, use the following table to indicate the potential return codes with which the chain will end:

Return Code	Condition
Successful (1)	All of the jobs end successfully.
Warning (4)	One of the jobs in the chain ends with a return code of “Warning”.
Non Fatal Error (8)	One of the jobs in the chain ends with a return code of “Non Fatal Error”.
Failed (12)	One of the jobs in the chain ends with a return code of “Failed”.

Terminated (16)	One of the jobs in the chain ends with a return code of "Terminated".
System Failure (20)	One of the jobs in the chain ends with a return code of "System Failure".

Problem Resolution

Please refer to the individual job Problem Resolution section for more details.

NYTI Process Chain: NYTI Process Job Step

Job Name	NYTI Process job step
Recommended Frequency	Yearly
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

Parameter editing is the first step in this NYTI Process job step. Any errors found will stop the chain at this step. The next step is the creation of XML files for each table listed in the parameter file when in populate mode. On each file is a listing of records to be added in the second job step. The step also creates an NYTI_PARMS.txt file for use in the next job step. When in delete mode, XML files are not created but an inverted parameter file is for the next job step.

Shown below are small samples of each file used and created by this job step. The NYTI_TABLE_Parms.txt file is shown first. Only the first two tables of many are shown. The following is a list of values that are found for the various parameters defined in the txt file that are used by the NYTI Process job step: I – Increment, B – Blank, O – Optional, T – True, and F – False.

```
NYTI_TABLE_Parms.txt
TABLE_NAME=R_FY
KEY_SELECTION=FY
CALENDAR_DAYS=F
FY=I
FY_END_DT=I
CLSD_FL=F
CLSNG_PROC_RUN_FL=F
_PARAM_LINE_
TABLE_NAME=R_APD
KEY_SELECTION=FY
CALENDAR_DAYS=F
```

```
FY=I
APD_END_DT=I
CLSD_FL=F
CLSNG_PROC_RUN_FL=F
```

From the above file, the job step creates a NYTI_PARM.txt file with information populated for the year to be populated or deleted (population shown below). This file will be used by the next job step. Shown below are just the same two tables of many and one can see that Fiscal Year 2011 is going to be populated as the job puts that year into the name of each XML file. When the run mode is delete, the NYTI_PARM_DEL.txt file is created instead of the NYTI_PARM.txt, and it has a similar layout but with a different ACTN_CD of TBLDELETE.

NYTI_PARM.txt

```
**ACTN_CD=TBLIMPORT
_PARAM_LINE_
FILE_NM=R_FY2011.XML
COMMIT_BLOCK=100
RESTART_FL=true
GENERATE_STATS=true
**ACTN_CD=TBLIMPORT
_PARAM_LINE_
FILE_NM=R_APD2011.XML
COMMIT_BLOCK=100
RESTART_FL=true
GENERATE_STATS=true
```

Another output of this job step is an XML file for each table in the NYTI_TABLE_Parms.txt file. The following is the XML file for the new record for 2011 to be added to the R_FY table.

R_FY2011.xml

```
<?xml version=”1.0” encoding=”ISO-8859-1” ?>
  <AMS_DATAOBJECT_XML_EXPORT_FILE VERSION=”1.0” EXPORT_DATE=”2008-02-27 10:50:37.000000”>
    <R_FY AMSDataObject=”Y”>
      <FY Attribute=”Y”><![CDATA[2011]]></FY>
      <FY_END_DT Attribute=”Y”><![CDATA[2011-06-30 00:00:00.000000]]></FY_END_DT>
      <CLSD_FL Attribute=”Y”><![CDATA[false]]></CLSD_FL>
      <CLSNG_PROC_RUN_FL Attribute=”Y”><![CDATA[false]]></CLSNG_PROC_RUN_FL>
      <AMS_ROW_VERS_NO Attribute=”Y”><![CDATA[3]]></AMS_ROW_VERS_NO>
```

```
        <TBL_LAST_DT Attribute=”Y”><![CDATA[2007-07-16
17:12:41.402250]]></TBL_LAST_DT>
        <ARCH_FL Attribute=”Y”><![CDATA[null]]></ARCH_FL>
      </R_FY>
</AMS_DATAOBJECT_XML_EXPORT_FILE>
```

In the case of any custom tables, the NYTI_TABLE_Parms.txt file delivered with the application should be updated or a copy of that file made, updated, and setup as the default for this job step. The ordering of the file is critical as records are added (and deleted) in the order (reverse order) listed in the file with system edits being enforced. For this reason, a parent table must be listed before a child table. Our example of R_FY and R_APD is one such example. It doesn’t matter that R_APD follows directly after R_FY, but must be listed before any table that has APD values such as R_CLDT.

When all tables should not be populated or purged at once, multiple parameter files can be made to populate/purge data in stages, with care taken to list tables in the proper order. When making custom parameter files, be sure to keep the entry for the R_FY table at the beginning, even it will have been populated already. If this table is missing, the job will fail.

When a site is using the Human Resource Management (HRM) application, sharing data from Financial (FIN), it is important to realize that certain reference data is needed in HRM longer than in FIN. The Calendar Date (R_CLDT), Department Fiscal Year (R_DEPT_FY), and Unit (R_UNIT) are three tables that should not be deleted from FIN with the NYTI Process. One option would be to leave the tables in the delete, and then take the XML files created for the delete and add them back with a SysManUtil import.

When the ACFR Reporting feature is being used, the NYTI process can populate new year data within Advantage Administration that is visible within the Statement pages found in Advantage Financial. The ACFR Reporting parameter controls this action. The ACFR tables are all populated but the Statement External Data (STMT_EXT_DATA) table, as that table is in Financial and should not be rolled forward automatically between years. There are some records that should, but the identification of those is beyond NYTI capabilities. Such records have to be rolled manually through an upload, while making any necessary updates. The Statement Cell table is processed twice. The first is an insert of all data except the Cell Expression (CELL_EXP) field with the second being an update after the Statement Rule table is populated. In the update pass the Cell Expression (CELL_EXP) field is updated just as if the Table Overlay parameter was set to *True*. Because of this ‘double pass’, you will see the Statement Cell table listed twice in the parameter file.

Because of this Statement Cell dependency with Statement Rule on the Cell Expression field, the deletion of certain Statement Cell records has to be done in two passes. Because of dependency the Records Failed Report will list out many records on the first attempt as failing to delete, but in the second pass, those records will be deleted. The NYTI chain will still finish with a Return Code of Successful because all records were eventually deleted.

NYTI_TABLE_Parms.txt

```
TABLE_NAME=STMT_DEF
KEY_SELECTION=STMT_FY
_PARAM_LINE_
TABLE_NAME=STMT_COLUMN
```

```
KEY_SELECTION=STMT_FY
_PARAM_LINE_
TABLE_NAME=STMT_ROW
KEY_SELECTION=STMT_FY
_PARAM_LINE_
TABLE_NAME=STMT_CELL
KEY_SELECTION=STMT_FY
CELL_EXP=B
_PARAM_LINE_
TABLE_NAME=STMT_RULE
KEY_SELECTION=STMT_FY
_PARAM_LINE_
TABLE_NAME=STMT_CELL
KEY_SELECTION=STMT_FY
_PARAM_LINE_
TABLE_NAME=STMT_CONDITION
KEY_SELECTION=STMT_FY
_PARAM_LINE_
```

Process Steps	Messages
1. Parameter Validation	• Each parameter is listed. • Any that are invalid are listed with an error
2. XML Files Creation	• No messages unless creation fails
3. Parameter File Creation	• No messages unless creation fails

Major Input

- NYTI_TABLE_PARM.txt
- Reference tables listed in above parameter file (see file for complete listing by searching for TABLE_NAME)

Major Output

- XML files created for each reference table (populate mode)
- NYTI_PARM.txt or NYTI_PARM_DEL.txt
- If INFO_INST = Y, then STMT_CELL, STMT_ROW, STMT_DEF, STMT_COLUMN, STMT_RULE and STMT_CONDITION tables get updated.

Batch Parameters

Parameter Name	Description	Default Value
Parameter Location at NYTI Process Job (AMSPARM)	• Location for the PARM_FILE parameter value.	• (blank)
Base Year (BASE_YEAR)	• Base Year to be copied. Please enter as CCYY. • Required if the Run Mode is Populate.	• (blank)
Delete Year (DELETE_YEAR)	• Delete Year to be deleted. Please enter as CCYY. The value entered must be flagged as Closing Process Run on the Fiscal Year table. • Required if the Run Mode is Delete.	• (blank)
Effective Date Rule (EFFECT_DT_RULE)	• Effective Date Rule: 1 – Copy effective from and to dates with an incremented year or 2 – Blank out date values	• 1
Inactive Code Rule (INACT_CD_RULE)	• Inactive Code Rule: 1 – Copy all records (active and inactive) or 2 – Copy active records only	• 1
New Year (NEW_YEAR)	• New Year to be populated. Please enter as CCYY. • Required if the Run Mode is Populate.	• (blank)
NYTI Parameter File Name (PARM_FILE)	• Parameter file listing the tables to be processed along with critical settings for each.	• NYTI_TABLE_PARM.txt
Run Mode (RUN_MODE)	• Run Mode: 1 – Populate or 2 – Delete	• 1
Table Overlay (TABLE_OVERLAY)	• Table Overlay (To overlay the target year records. T - for	• F

	YES or F - for NO)	
infoAdvantage installed? (INFO_INST)	• ACFR Reporting used? (Y - for YES or N - for NO) • When set to Y data is rolled in the various STMT_* tables in the Administration database. If set to N, those tables are not rolled.	• N

Job Return Code

Return Code	Condition
Successful (1)	XML files successfully created for each table in the parameter file with data defined in the Base Year (populate mode) or Delete Year (delete mode).
Warning (4)	• Job does not end with this return code.
Non Fatal Error (8)	Job does not end with this return code.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • TXT or XML files could not be created When this job ends with a return of code Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return of code System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

- XML files sorted on primary keys

Selection Criteria

- Select records from each table for the specified Base Year or Delete Year for Populate or Delete Mode respectively
- If the mode is populate and Inactive Code Rule is 2, then select records for each table for the specified base year and with ACT_FL = true.
- When the CODE OVERLAY parameter is set to F, then if a record already exists in the new year, it will be reported as a unique constraint error (duplicate record) in the report. However, if the parameter is set to T, then the following will occur:

a. If record exists in current year but not the upcoming fiscal year, it will be selected and inserted into the upcoming fiscal year as normal.

b. If record exists in upcoming fiscal year, but not the current year, nothing will be done, and record should remain unchanged in upcoming fiscal year.

c. If record exists in both current year and upcoming year and the Last Date is greater on the current year record, then overlay field values from current year onto upcoming fiscal year. Here the 'source' record in the current year has been updated so the record in the upcoming year should be updated too.

d. If record exists in both current year and upcoming year and the Last Date is greater on the upcoming fiscal year record, then do nothing and the upcoming fiscal year record should remain unchanged. Here the upcoming record has already had something on it changed that should make it different than the current year record.

Problem Resolution

Possible Return Codes	**Condition**	**Recommendation**	**Other Instructions**
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	Job does not end with this return code.	N/A	N/A
Non Fatal Error (8)	Job does not end with this return code.	N/A	N/A
Failed (12)	Job failed due to Fatal conditions. In this step, the job can fail under the following two conditions. 1) Parameter is invalid 2) Delete Year is not closed 3) Encountered a run time exception	1) Use correct parameters in a subsequent run 2) Ensure CLSNG_PROC_RUN_ FL = 1 for Delete Year on R_FY. Can be set in the database if no accounting activity occurred in the year so that it doesn't need a formal closing. 3) If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	N/A

System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	N/A

NYTI Process Chain: Export/Import Tables Job Step

Job Name	Export/Import Tables
Recommended Frequency	Yearly
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

The Export/Import Tables job step takes each of the XML files created from a populate mode and the NYTI_PARMS.txt file and performs a Table Import, saving the records in the order specified. Any messages issued from a successful save (budget tables that issue: "In order for this change to take effect, all instances of the application server must be restarted to ensure proper functioning of the system (A667)) or an unsuccessful save are recorded in the job log for the NYTI Report job step.

When in delete mode, this job step takes the NYTI_PARM_DEL.txt file and creates XML files for the next job to delete records.

Process Steps	Messages
1. Parameter Validation	• Each parameter is listed. • Any that are invalid are listed with an error
2. XML File Importing	• Action TBL_IMPORT Processing Parameter Line: FILE_NM=[XML File Name], COMMIT_BLOCK=100, RESTART_FL=true, GENERATE_STATS=true • Rows Processed: ## • Rows Saved: ## • Rows In Error: ## • If any messages resulted from a successful or unsuccessful save, the messages are listed before the next table.
3. XML File Creation	• Action TBL_EXPORT Processing Parameter Line: FILE_NM=[Table NameCCYY.XML], TBL_NM=R_MAT_TS_CLS_INF, FM_KEY=FY=2011, TO_KEY=FY=2011

Major Input

- XML files created for each reference table (populate)

- NYTI_PARM.txt (populate)
- NYTI_PARM_DEL.txt (delete)

Major Output

- Reference tables listed in parameter file (populate)
- XML files created for each reference table (delete)

Batch Parameters

Parameter Name	Description	Default Value
Parameter file (PARM_FILE)	• NYTI Parameter File Name	• $$AMSPARM$$/NYTI_PARM.txt

Job Return Code

Return Code	Condition
Successful (1)	All XML files successfully imported without any records failing to load
Warning (4)	• One or more records failed to load to a table.
Non Fatal Error (8)	One or more records failed to load to a table.
Failed (12)	The job will fail under the following conditions: • Parameter is invalid • TXT or XML files could not be found or created When this job ends with a return of code Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return of code System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

- None

Selection Criteria

- None

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	One or more records failed to import from a table.	Review the job log or the report created from the NYTI Report job step for the records that failed. If records are required, address the reason(s) for failure and load the records manually or with another run of the NYTI chain, while expecting many duplicate record errors for records successfully loaded the first time.	N/A
Non Fatal Error (8)	One or more records failed to import from a table.	Review the job log or the report created from the NYTI Report job step for the records that failed. If records are required, address the reason(s) for failure and load the records manually or with another run of the NYTI chain, while expecting many duplicate record errors for records successfully loaded the first time.	N/A
Failed (12)	Job failed due to Fatal conditions. In this step, the job can fail under the following two conditions. 1) Parameter is invalid 2) Encountered a run time exception	1) Use correct parameters in a subsequent run 2) If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	N/A

System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	N/A

NYTI Process Chain: Delete Records Job Step

Job Name	Delete Records
Recommended Frequency	Yearly
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

The Delete Records job step takes each of the XML files created from a delete mode in the Export/Import Tables job step, the NYTI_PARM_DEL.txt file, and performs a Table Delete. The delete is done in reverse order of a populate. Any messages issued from a successful delete (budget tables that issue: "In order for this change to take effect, all instances of the application server must be restarted to ensure proper functioning of the system (A667)) or an unsuccessful delete are recorded in the job log for the NYTI Report job step.

Process Steps	Messages
1. Parameter Validation	• Each parameter is listed. • Any that are invalid are listed with an error
2. XML Table Deleting	• Action TBL_DELETE Processing Parameter Line: FILE_NM=[XML File Name], COMMIT_BLOCK=100, RESTART_FL=true, GENERATE_STATS=true • Rows Processed: ## • Rows Saved: ## (here saved means successfully deleted) • Rows In Error: ## • If any messages resulted from a successful or unsuccessful save, the messages are listed before the next table.

Major Input

- XML files created for each reference table
- NYTI_PARM_DEL.txt

Major Output

- Reference tables listed in parameter file

Batch Parameters

Parameter Name	Description	Default Value
Parameter file (PARM_FILE)	- NYTI Parameter File Name	- $$AMSPARM$$/NYTI_PARM_DEL.txt

Job Return Code

Return Code	Condition
Successful (1)	All records deleted successfully.
Warning (4)	- One or more records failed to delete from a table.
Non Fatal Error (8)	One or more records failed to delete from a table.
Failed (12)	The job will fail under the following conditions: - Parameter is invalid - TXT or XML files could not be found When this job ends with a return of code Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return of code System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

- None

Selection Criteria

- None

Problem Resolution

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

Warning (4)	One or more records failed to delete from a table.	Review the job log or the report created from the NYTI Report job step for the records that failed. If records should be deleted, address the reason(s) for failure and delete the records manually or with another run of the NYTI chain. This may required the addition of an R_FY record back with the Closing Process Run Flag set to 1.	N/A
Non Fatal Error (8)	One or more records failed to delete from a table.	Review the job log or the report created from the NYTI Report job step for the records that failed. If records should be deleted, address the reason(s) for failure and delete the records manually or with another run of the NYTI chain. This may required the addition of an R_FY record back with the Closing Process Run Flag set to 1.	N/A
Failed (12)	Job failed due to Fatal conditions. In this step, the job can fail under the following two conditions. 1) Parameter is invalid 2) Encountered a run time exception	1) Use correct parameters in a subsequent run 2) If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	N/A

NYTI Process Chain: NYTI Report Job Step

Job Name	NYTI Report
Recommended Frequency	Yearly
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes

Overview

In this is the final job step of the NYTI chain, a report is built summarizing the processing activity (delete or populate) of the chain. Each table in the parameter file is listed as a page in the report. Totals are given for the number of records selected to copy or delete, the number of records successfully copied or deleted, and the number of records that failed to save or delete. Any system messages issued are listed in the report.

Below is a sample report from the populate mode.

```
Report ID : NYTIRep                          City of Greenville                                PAGE : 1
DATE RUN : 02-27-2008
TIME RUN: 14:24:44                         NYTI Process Report
Run Mode 1
Base Year 2008
New Year 2011
-------------------------------------------------------------------
Table Name                                         R_FY
Records In Base Year                                 1
Records Selected for New Year                        1
Records Successfully Imported                        1
```

Records Failed To Import 0

Below is a sample report from the delete mode. The report shows an unsuccessful delete of the R_FY record because of a deletion failure on an earlier table – Required Budget (R_GN_BUD_RULE).

```
Report ID : NYTIRep                          City of Greenville                              PAGE : 141
DATE RUN : 02-27-2008
TIME RUN: 14:05:24                         NYTI Process Report
Run Mode 2
Delete Year 2011
-------------------------------------------------------------------
Table Name                                         R_FY
Records Selected for Delete Year                     1
Records Successfully Deleted                         0
Records Failed To Delete                             1
Component Name: R_FY Error Context :FY = 2011 Attribute ID :FY Message ID :A760 Severity Level :SEVERE
Message: This record cannot be deleted because it is used on at least one Required Budget record. (A760)
```

Process Steps	Messages
1. Parameter Validation	• No messages are issued for this step
2. Report Creation	• Reports output folder mapped

	• HTML report file path: [full path] \NYTIReport\Html\NYTIReport.html • PDF report file path: [full path] \NYTIReport\PDF\NYTIReport.pdf • Rendering report started • Rendering report completed

Major Input

- Job Log from the Import/Export Tables or Delete Records job steps
- Report Layout files

Major Output

- NYTI Process Report

Batch Parameters

Parameter Name	Description	Default Value
Client Name (CLIENT NAME)	• Client name to be used on report header	• (blank)

Job Return Code

Return Code	Condition
Successful (1)	Report was successfully generated.
Warning (4)	• Job step does not end with this status.
Non Fatal Error (8)	Job step does not end with this status.
Failed (12)	The job will fail under the following conditions: • Report directory not found • Report layout files not found When this job ends with a return of code Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return of code System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

- None

Selection Criteria

- None

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	Job step does not end with this status.	N/A	N/A
Non Fatal Error (8)	Job step does not end with this status.	N/A	N/A
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following condition. 1) Encounters any runtime exceptions If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	N/A

2.1.26 Online Forms Cleanup

Batch or Job Name	Online Forms Cleanup
Recommended Frequency	This job should be scheduled to run daily to purge select records from the View Forms Management (VW_FORMS_MGMT) table and related PDF files.
Single Instance Required	No
Reports generated	No

Overview

Online Forms Cleanup is a batch job in CGI Advantage Financial and is available under the **Financial** : **Utilities** : **Batch Jobs** folder.

This job loops through online View Forms Management records selecting those that have been previously accessed, or those that have expired, and deletes those records along with the corresponding PDF files. Generated PDF files will be purged nightly if the related View Forms Management records have been accessed, or per a predetermined period of time if it has not been accessed.

Batch Parameters

Parameter	Description	Default Value
CLEANUP_DT	Cleanup Date (Valid format is MM/DD/YYYY)	N/A
COMMIT_BLOCK	Commit block size Indicates the number of View Forms Management records that should be processed before transaction is committed. If value is not specified then default value of 1000 is assumed. Too low value for this parameter will increase processing time. Too high value would require more memory.	1000
PURGE_TYPE	Type of Purge,1 - Online forms, 2 - Offline forms, 3 - both Online and Offline forms	N/A

Steps in Running this Process

1. Parameter Validation - In this step the job will validate the batch parameters. If any parameter is invalid, the appropriate message will be displayed in the log and the job will return as “Failed”.

2. Selection of Records - View Forms Management records are selected if they satisfy these criteria:

   - Online (indicating whether the forms printing request was an online request) = true

- Accessed (indicating whether the PDF file has been accessed) = true OR (Expiration Date is not blank AND Expiration Date <= Cleanup Date parameter)
- Application (Advantage application where the forms request was initiated) = PrimaryApplication specified in AMS30Params.ini

3. Processing of Records

4. Removal of File and Directory

   - File is removed from the file location as well as the file location itself.

     Note: While processing forms entries for offline printing (records with Job ID populated), for each selected record, the job checks if navigating to the parent directory two levels up corresponds to the root folder “FormOutput”. If so, then only the PDF file mentioned on the selected record is deleted. This is applied to avoid deletion of all folders and files for non-standard offline forms printing where the default folder structure is not followed.

5. Removal of View Forms Management Record

Major Input

- View Forms Management table (VW_FORMS_MGMT)

Output

- View Forms Management table (VW_FORMS_MGMT)

Sort Criteria

N/A

Batch Job Return code

The following table shows the potential return codes for the Online Forms Cleanup job.

Return Code	Condition
Successful (1)	The jobs end successfully without any error.
Warning (4)	N/A
Non Fatal Error(8)	N/A
Failed (12)	The job will fail under the following conditions: • Parameters are invalid. • Run time exceptions for unexpected situations.
Terminated (16)	N/A
System Failure (20)	NA

Problem Resolution

Look into the job log for errors. Correct the problem and restart the job.

If the job fails for any batch parameter validation error then correct the erroneous batch parameter value(s) and re-schedule the job.

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
Failed (12)	Required Parameters are not entered **Sample Message:** The Purge Type parameter is required.	Enter the Purge Type and restart the job.	
	Entered Parameters are not valid **Sample Message**: Cleanup Date not entered properly. Valid format is MM/DD/YYYY. **Sample Message**: Purge Type should be a positive integer. **Sample Message:** Commit block size should be a positive integer **Sample Message:** Commit block size should be greater than zero **Sample Message**: The Type of Purge parameter is not valid. Enter "1", "2", or "3"	Enter the correct parameter and restart the job.	

	and resubmit the job.		
	Transaction commit failed **Sample Message:** Commit failed. Going to stop processing further.	The reason for the failure needs to be investigated before re-running the job.	
	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	

Troubleshooting

If the process fails due to any reason then:

- Check the Log file for any errors that may have occurred while the program was running.

2.1.27 Output File Maintenance

Chain or Job Name	Output File Maintenance
Recommended Frequency	On demand or as part of a chain
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

As generated files of all types often need to be moved from where they are originally written, this batch process does that move as a separate job run apart from the one that generated the file or as a job step configured into the chain that generated the file. Another feature in the job is to also create a copy of the input file with a date/timestamp appended to the filename for retention purposes as a backup.

The steps involved in this process are:

1. **Parameter Validation**: The process validates the batch parameters.
2. **Input File Checking**: The process checks whether the input file exists where indicated.
3. **File Archive**: The process archives the file to the indicated archive location.
4. **Move the File to Target Location**: The process moves the input file to target location.

The following table shows a sample of the messaging.

Process Steps	Messages
1. Parameter Validation	Run started • Each parameter is listed Validating parameters • If the parameters are invalid, a message indicating the error is displayed
2. Input File Checking	The import file does not exist • No files were moved or archived
3. File Archive	<input file name> was successfully archived
4. Move the File to Target Location	Run ended • <Input file name> was moved to the Target location

Restartability

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

Major Input

- Input File (see parameters)

Job Parameters

The following are the parameters for this job with delivered defaults.

Parameter	Description	Default Value
Location of input file (AMSIMPORT)	A required selection parameter that lists the folder location where the input file is located.	$$AMSROOT$$/ExportImport
Name of input file (IMPORT_FILE)	The required selection parameter of the name of the input file. There are two special values for this parameter when selecting Check Writer files where a job ID is included in the file names to make them unique and prevent overwrite by a subsequent job. The following are used so that the job does not need to knw the actual file name(s): • cwACHFile or cwACHFile.dat or cwACHFile.txt • cwPrintGenFile or cwPrintGenFile.xml or cwPrintGenFile.txt	No Default
Location where input file is moved (TARGET_DIR)	A required output parameter of where the file will be moved.	No Default
Location of Archive Directory (ARCHIVE_DIR)	An optional output parameter of where a copy of the file will be saved. When left blank, no copy will be made.	No Default

Major Output

- File moved
- Archive file created

Job Return Code

The following table shows the possible job Return Codes for the Output File Maintenance job:

Return Code	Condition
Successful (1)	Parameters were valid and the input file has been moved successfully to Target Location.
Warning (4)	The job gives a warning under the following two special conditions that coincide with Check Writer ACH File Generation and Check Writer Printing Generation processes: • If the IMPORT_FILE parameter value is ‘cwACHFile’ or ‘cwACHFile.dat’ or ‘cwACHFile.txt’ and the job didn’t find

	the input files which are beginning with ACH_CCD_ (or) ACH_PPD_ and end with .dat • If the IMPORT_FILE parameter value is ‘cwPrintGenFile’ or ‘cwPrintGenFile.xml’ or ‘cwPrintGenFile.txt’ and the job didn’t find the input files which are beginning with Check_ (or) RA_ and end with .xml
Non-Fatal Error (8)	This job does not issue this return code.
Failed (12)	The job fails under the following conditions: • The job fails if the parameters are invalid. • The job fails if the import file does not exist.
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 the database server or network issues.

Sort Criteria

N/A

Selection Criteria

When the input file parameter value is ‘cwACHFile’ or ‘cwACHFile.dat’ or ‘cwACHFile.xml’, all those *.dat files that start with ACH_PPD_ <jod id> or ACH_CCD_<job id> will be selected and moved.

When the input file parameter value is ‘cwPrintGenFile’ or cwPrintGenFile.xml’ or ‘cwPrintGenFile.txt’, all those *.xml files that start with Check_<job id> or RA_<job id> will be selected and moved.

Problem Resolution

The following table shows the various steps that the job goes through and the messages issued at each step:

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Parameter validations 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)	The job fails if the Parameters are invalid. **Sample Message:**	If the required parameters are not entered (or) not entered correctly, enter the	N/A

	"Invalid Archive directory supplied"	required parameters and submit a new job.	
Terminated (16)	The job is terminated manually by the user.	The reason for the termination needs to be investigated before submitting a new job.	N/A
System Failure (20)	The job is terminated because of the database server or network issues.	The reason for the System Failure needs to be investigated before submitting a new job.	N/A

Step 2: Input File Checking

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	The input file was found.	N/A	N/A
Warning (4)	One of two exceptions noted for Check Writer and Check Printing is found.	Investigate whether data should have been created by either process, possibly running either process after addressing errors within.	N/A
Non-Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	The job fails if the input file doesn't exist. **Sample Message:** "The Import File doesn't exist"	If the Input file doesn't exist, enter the valid input file name, and reschedule the job.	Review the results of the process that was to have created a file to see if it failed and needs to be submitted again.
Terminated (16)	The job is terminated manually by the user.	The reason for the termination needs to be investigated before submitting a new job.	N/A
System Failure (20)	The job is terminated because of the database server or network issues.	The reason for the System Failure needs to be investigated before submitting a new job.	N/A

Step 3: File Archive

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	The process created the archive file successfully.	N/A	N/A

Warning (4)	N/A	This step does not issue this return code.	N/A
Non-Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	The job fails if the input file doesn't exist as there are no files to archive. **Sample Message:** "No Files were moved or archived"	If the Input file doesn't exist, enter the valid input file name and submit a new job.	N/A
Terminated (16)	The job is terminated manually by the user.	The reason for the termination needs to be investigated before submitting a new job.	N/A
System Failure (20)	The job is terminated because of the database server or network issues.	The reason for the System Failure needs to be investigated before submitting a new job.	N/A

Step 4: Move the File to Target Location

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	The process moved the input file to the target location.	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)	The job fails if the input file doesn't exist as there are no files to move. **Sample Message:** "No Files were moved or archived"	If the Input file doesn't exist, enter the valid input file name and reschedule the job.	N/A
Terminated (16)	The job is terminated manually by the user.	The reason for the termination needs to be investigated before submitting a new job.	N/A
System Failure (20)	The job is terminated because of the database server or network issues.	The reason for the System Failure needs to be investigated before submitting a new job.	N/A

2.1.28 Paid Check Archiving

Chain Name	Paid Check Archiving
Recommended Frequency	On Demand
Single Instance Required	Yes
Can be restarted?	N/A
Reports generated	Yes

Overview

With the progression of time, the data built up in the Paid Checks (PDCHK) table of Advantage Financial becomes considerable. The Paid Archiving process uses standard SysManUtil (System Maintenance Utility) Table Archiving features to export selected Paid Check data to XML files and then purge the selected data. Once parameters are validated and Paid Check records are selected for a given execution, Archive Facilitator records are created and an Archive Facilitator step is invoked to spawn concurrent SysManUtil jobs to perform the archiving.

The following job steps comprise the Paid Check Archiving chain process:

- Archiving Preprocessor
- Archive Facilitator

The steps are both singleton jobs. This means that only one of their instances can be run at a time. The Archive Facilitator does have the ability to launch the System Maintenance Utility jobs in parallel.

The output from the process includes the Paid Check Archiving Report, which shows a list of selected records. When run in Report Only mode, the chain ends after the Archiving Preprocessor completes the report and does not archive or purge any records.

The acceptable job return code (configured in the Configure Chain Job section of the Job Setup in CGI Advantage) for the Archiving Preprocessor job in the Paid Archive chain is delivered to be set to Successful. As with all CGI Advantage chain jobs, this acceptable return code is configurable and may be changed to meet certain requirements.

This chain implements the Data Warehouse Archived Record Queue process, which is used to retain the table names and key values for records that are archived and deleted. The infoAdvantage reporting system uses this information to distinguish between records that have been archived (as valid historical data) before deletion and records that have simply been deleted from the system (as no longer needed). The process is enabled for most archiving processes when the value of Application Parameter Enable Data Warehouse Archived Record Queue (ENABLE_DW_ARCH_QUEUE) is *True*. Please see the *CGI Advantage System Administration Guide* for more information regarding this process and the Application Parameter.

Chain Job Return Code

The following table shows the potential return codes for the Paid Check Archiving chain. Note that the chain job will end with the highest return code across all of the jobs.

Return Code	Condition

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

Problem Resolution

Please refer to the individual job "Problem Resolution" section for more details.

Paid Check Archiving Chain: Archiving Preprocessor Job

Job Name	Archiving Preprocessor
Recommended Frequency	On Demand This job must be run as part of the Paid Check Archiving Chain.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Paid Check Archiving Report

Overview

This step first validates parameters. If parameters are valid, it then identifies the rows to-be-archived based on the selection criteria, and divides the selected rows into groups for each export file. For example, if PDCHK has 20,000 rows that meet the selection criteria and the Job Block Size parameter is 10,000, then two export files will be created. The Bank Account Code, Check/EFT Number, Amount, Status and Cleared Date of records eligible for archival are written to the Paid Check Archiving Report. If the run mode is set to 'Report Only', the process will end at this point and no PDCHK records will be archived or purged.

In 'Full Run' run mode, a System Maintenance Utility parameter file is created for each export file to be created. The parameter file contains the table name (for example, AP_PD_CHK), the identification of specific records to be archived, and other settings. A Facilitator table record referencing the SMU parameter file is created and saved. Facilitator records can be viewed in the Archive Facilitator Inquiry page (QARC) for the status. One Facilitator table record is created per SMU parameter file.

Process Steps	**Messages**

1. Parameter Validation	- Validating Batch Parameters - Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value is displayed in the log. - Batch Parameters are/are not valid.
2. Pre-processing	- Initializing report started. - Initializing report completed. - Record selection started. - Record selection completed. - Rendering report started. - Rendering report completed.

Major Input

- Paid Check table (AP_PD_CHK)

Batch Parameters

Parameter	Description	Default Value
AMSEXPORT	Export Location at Archiving Preprocessor Job Required. This must be a valid directory and will be used to determine where the export files will be written. (Depending on the volume of records, the size of the archived files may warrant specifying a location other than the default Export/Import directory.)	$$AMSROOT$$/ ExportImport
AMSPARM	Parameter Location at Archiving Preprocessor Job Required. This must be a valid directory containing the file designated by the Parameter File parameter.	$$AMSROOT$$/ Parms
ARCH_CHECK_WRITER	Archive Check Writer Payments Required. If the value is 'N', Check Writer Payment records will be excluded from selection. If the value is 'Y', no records are excluded.	Y
ARCH_STALE_ESCHEAT	Archive Stale Dated and Escheat Required. If the value is 'N',	Y

	records with Status 'Stale Dated' and 'Escheat' will be excluded from selection. If the value is 'Y', no records are excluded.	
ARCH_VOID_CHECKS	Archive Void Checks Required. If the value is 'N', records with Status 'Void' will be excluded from selection. If the value is 'Y', no records are excluded.	Y
BANK_ACCT_CD	Bank Account Code Optional. If provided, there can be only one value, and it must be valid in the Bank Account (BANK) table. Only records with the Bank Account Code will be selected.	No Default
BYPASS_DISB_DOC	Bypass Disbursement Transaction Archive Edit Required. If the value is 'Y', all records are considered for archive regardless of whether their associated Disbursement transaction (AD or MD) or associated Check Writer File has been archived. If the value is ‘N’, only records whose associated Disbursement Transactions (AD or MD) or Check Writer File has been archived will be eligible for selection. For records where Transaction Type is 'AD' and 'MD', the associated transaction will be verified in the transaction catalog. For Check Writer records (where Transaction Type is '**'), the associated record will be verified in the Check Writer Payment table.	Y
CLIENT_NM	Client Name for Reports Optional.	No Default
COMMIT_BLOCK_SIZE	Commit Block Size Required. This value will determine how often database transactions will be committed and can be used for performance tuning. The value must be a positive integer.	1000

FILE_PREFIX	File Prefix Required. Prefix used for archive file names.	No Default
JOB_BLOCK_SIZE	Job Block Size Required. The maximum number of lines in an output file.	10000
LAG_DAYS	Lag Days Required. This is used to determine how many days between the Cleared Date and the Application Date have passed in order to be selected for archive. The value must be a valid positive number.	1
RUN_MODE	Run Mode Required. Enter '1' for Report Only, '2' for Full Run.	1
SMU_FILE_PREFIX	SMU Facilitator Parameter File Prefix Required, not editable. This value will be used to determine a SysManUtil input parameter file name for each Facilitator job.	PCArch
SMU_REPORTS	SMU Reports Required. If this value is Y, each spawned SMU job will create a Table Archive Report that lists the record keys processed for the run.	N

Major Output

- Paid Check Archiving Report
- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page.
- ARCH_DW_QUEUE_REC – The Data Warehouse Archive Record table keeps track of the values for all primary keys of each record being archived and deleted. The table name and attribute names are stored in ARCH_DW_QUEUE_TBL and linked to this record by unique ID.
- ARCH_DW_QUEUE_TBL – The Data Warehouse Archive table keeps track of the table name and primary key attribute names of each table being archived. If a record already exists for the table being archived, it will reuse the record. If it does not yet exist, it will be created.

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	This return code is issued under the following conditions: • No records eligible for archive selection. • The Run Mode parameter is set to '1' (Report Only).
Non Fatal Error (8)	This step does not issue this return code.
Failed (12)	The job fails under the following conditions: One or more parameter(s) are invalid. • One or more required parameter(s) are not entered. • Run time exceptions for unexpected situations. When this job ends with a return of code Failed, subsequent jobs in the chain are set to Inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain are set to Inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return of code System Failure, subsequent jobs in the chain are set to Inactive.

Sort Criteria

Records are grouped according to the year and month of the Cleared Date for each record. Within these groups, the records are ordered by Bank Account Code and Check/EFT Number.

Selection Criteria

Paid Check records are selected by:

- If a Bank Account Code (BANK_ACCT_CD) parameter value is provided, then only records with a matching Bank Account Code are selected, otherwise all Bank Account Codes are eligible for archiving (but subject to the remaining criteria).

Once records are identified by Bank Code (if any) it is simpler to describe record selection in terms of how records are excluded from eligibility based on various job parameters:

- Records are **excluded** if their Cleared Date is **not** eligible per the LAG_DAYS parameter value.
- Check Writer Payments are **excluded** if the Archive Check Writer Payments (ARCH_CHECK_WRITER) parameter is 'N'.
    - PD_CHK (AP_PD_CHK) Check Writer payment records are identified as those having Transaction Type = '**' and Transaction Code <> '**' (i.e. Transaction Code = CW or CE).
- If the Archive Stale Dated and Escheat (ARCH_STALE_ESCHEAT) parameter is 'N', then records with Check Statuses of Stale Dated and Escheat will be **excluded**.
    - The AP_PD_CHK Check Status (CHK_STA) field is used to identify Stale Dated and Escheat checks.
- If the Archive Void Checks (ARCH_VOID_CHECKS) parameter is 'N', then records with Check Status of Void will be **excluded**.
    - The AP_PD_CHK Check Status (CHK_STA) field is used to identify Void checks.
- If the Bypass Disbursement Transaction Archive Edit (BYPASS_DISB_DOC) parameter is 'N', then records having associated non-archived Disbursement Transactions or Check Writer Payment files are **excluded**.
    - For Paid Check records directly related to Disbursement transactions (Transaction Types AD and MD), Paid Check Archiving will check if the referenced transaction still exists on the Transaction Catalog (DOC_HDR). If the referenced transaction still exists in the system (that is, has not been archived) then the Paid Check record will be excluded.
    - For Paid Check records related to Check Writer Payments, Paid Check Archiving will check if the referenced Check Writer file has been archived from the Check Writer Header (R_AP_CW_HDR) table. If the referenced Check Writer file still exists in the system (that is, has not been archived) then the Paid Check record will be excluded.

All remaining eligible Paid Check records are selected for archive.

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered.	Schedule a new job after entering a valid value for	N/A

	Sample Message: Parameter File Prefix is required.	the parameter.	
	Entered parameters are not valid. Sample Message: Parameter Job Block Size value 'ABC' is invalid; it must be a positive integer.	Schedule a new job after entering a valid value for the parameter.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Pre-processing

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	The Run Mode parameter is set to '1' (Report Only). This is a normal condition that sets any subsequent jobs to inactive.	N/A	N/A
	No records eligible for archive selection.	Confirm selection criteria before scheduling a new job.	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A

Failed (12)	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Paid Check Archiving Chain: Archive Facilitator Job

Job Name	Archive Facilitator
Recommended Frequency	On Demand This job must be run as part of the Paid Check Archiving Chain.
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	None

Overview

The Facilitator job spawns multiple System Maintenance Utility (SMU) jobs to export (archive) selected Paid Check records to files, then delete the records from the table. The Archive Facilitator step retrieves the records from the Facilitator table for the chain's Job ID. It then reads the file names from these records and spawns multiple SMU jobs, using the parameter file names from the Facilitator table as input for the SMU jobs. The parameter files specify the "Table Archive" command with other options set in the Archive Preprocess step.

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value is displayed in the log. • Batch Parameter validation completed
2. Facilitator Job Processing	• The Run Number for this archive/restore process = xxx (xxx being the chain's Job ID) • For each job in the Facilitator for the Run

	Number: • SMU Job - xxx - Spawned (xxx being the Job ID) • Each spawned job will have either of these results: • SMU Job - xxx – Processing completed successfully (xxx being the Job ID) • SMU Job - xxx - Failed (xxx being the Job ID) • The job slept a total of xxx times, for a Total Sleep Time of yyy seconds. (xxx and yyy being the counts)

Restartability Information

This job can be restarted if it fails due to any reason. After restart, the job will continue to process the records based on the status of each Facilitator record. Restarting this job step will restart all spawned jobs automatically.

Major Input

- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page.
- BS_AGENT – The job will consult the Job Manager for the status of jobs that have been spawned.
- AP_PD_CHK – Each SMU job spawned by the Facilitator job will export a subset of records, depending on the starting and ending keys specified in the parameter file supplied to the Facilitator record.

Batch Parameters

Parameter	Description	Default Value
ARCHIVE_RESTORE_ID	Archive Restore ID Required, not overrideable. This value tells the Facilitator to select the archive table process.	1 (Table Archive)
COMMIT_BLOCK_SIZE	Commit Block Size Required. This value will determine how often database transactions will be committed and can be used for performance tuning. This must be a positive integer.	1000

PROCESSOR_NO	Number of jobs for Facilitator to keep running Required. This value sets the number of jobs to launch and keep submitted/running. Exceeding the number of available VLS is acceptable. This must be a positive integer.	1
SLEEP_TIME	The number of seconds to wait between polling occurrences. Required. This value sets the length of time between each iteration, which checks the status of the jobs that are running and the jobs that are launching for unprocessed Facilitator records, as necessary.	5
UPDATE_STATUS	Update Status Required, not overrideable. This value tells the Facilitator to update the status of each Facilitator record based on the status of its corresponding job in the Job Manager.	Y

Major Output

- Exported files
- FACILITATOR – The Facilitator table is updated to reflect the Job Manager status.

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	This return code is issued under the following conditions: • No Facilitator records found for Run Number = xxx. Nothing to process. (xxx being the chain's Job ID) • SMU Job Parameter File NOT Found – (file name) Record skipped.
Non Fatal	This step does not issue this return code.

Error (8)	
Failed (12)	The job fails under the following conditions: One or more parameter(s) are invalid. • One or more required parameter(s) are not 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 Criteria

None

Selection Criteria

FACILITATOR records contain the location of input parameter files for use by spawned System Maintenance Utility batch jobs.

Problem Resolution

No database restore is required. Both the Archive Facilitator chain step and each spawned System Maintenance Utility job can be restarted.

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)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	No Facilitator records found for the Run Number (Nothing to process).	The reason for the missing records must be investigated before a new chain job is run.	N/A

Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. **Sample Message:** SLEEP_TIME is required to run the Facilitator.	Schedule a new job after entering a valid value for the parameter.	N/A
	Entered parameters are not valid. **Sample Message:** The SLEEP_TIME must be an integer and > 0.	Schedule a new job after entering a valid value for the parameter.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Facilitator Job Processing

Possible Return Codes	**Condition**	**Recommendation**	**Other Instructions**
Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	SMU Job Parameter File not found.	If all files are missing, the reason must be investigated before a new chain job is run. If the files can be located and moved to the expected location, the job can be restarted.	N/A

Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	This return code will be issued when any spawned SMU job fails. **Sample Message:** SMU Job – xxx – Failed	Determine the cause for the spawned SMU job to fail and resolve. The spawned job can be restarted independently if desired, and restarting the chain job will also restart any failed SMU jobs.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

2.1.29 Paid Check Table Restore

Job Name	Paid Check Table Restore
Recommended Frequency	On Demand
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	None

Overview

The Paid Check Archiving chain job archives Paid Check records to XML files. This job will reverse the process and add the records back to the table, given the Job ID of a Paid Check Archiving chain, along with its respective parameter and XML files in their original locations.

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value is displayed in the log. • Batch Parameter validation completed
2. Facilitator Job Processing	• The Run Number for this archive/restore process = xxx (xxx being the chain's Job ID) • For each job in the Facilitator for the Run Number: • SMU Job - xxx - Spawned (xxx being the Job ID) • Each spawned job will have either of these results: • SMU Job - xxx – Processing completed successfully (xxx being the Job ID) • SMU Job - xxx - Failed (xxx being the Job ID) • The job slept a total of xxx times, for a Total Sleep Time of yyy seconds. (xxx and yyy being the counts)

Restartability Information

This job can be restarted if it fails due to any reason. After restart, the job will continue to process the records based on the status of each Facilitator record. Restarting this job step will restart all spawned jobs automatically.

Major Input

- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page.
- BS_AGENT – The job will consult the Job Manager for the status of jobs that have been spawned.
- Parameter files created by the Paid Check Archive chain process.
- XML files created by the Paid Check Archive chain process.

Batch Parameters

Parameter	Description	Default Value
ARCHIVE_RESTORE_ID	Archive Restore ID Required, not overrideable. This value tells the Facilitator to select the archive table process.	2 (Table Unarchive)
COMMIT_BLOCK_SIZE	Commit Block Size Required. This value will determine how often database transactions will be committed and can be used for performance tuning. This must be a positive integer.	1000
PROCESSOR_NO	Number of jobs for Facilitator to keep running Required. This value sets the number of jobs to launch and keep submitted/running. Exceeding the number of available VLS is acceptable. This must be a positive integer.	2
SLEEP_TIME	Number of seconds to wait between polling occurrences Required. This value sets the length of time between each iteration, which checks the status of the jobs that are running and the jobs that are launching for unprocessed Facilitator records, as necessary.	5
UPDATE_STATUS	Update Status Required, not overrideable. This	Y

	value tells the Facilitator to update the status of each Facilitator record based on the status of its corresponding job in the Job Manager. Valid values are Y or N.	

Major Output

- FACILITATOR – The Facilitator table is updated to reflect the Job Manager status.

Job Return Code

The following table shows the potential job return codes.

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	This return code is issued under the following conditions: • No Facilitator records found for Run Number = xxx. Nothing to process. (xxx being the chain's Job ID) • SMU Job Parameter File NOT Found – (file name) Record skipped.
Non Fatal Error (8)	This step does not issue this return code.
Failed (12)	The job fails under the following conditions: One or more parameter(s) are invalid. • One or more required parameter(s) are not 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 Criteria

None

Selection Criteria

FACILITATOR records contain the location of input parameter files for use by spawned System Maintenance Utility batch jobs.

Problem Resolution

No database restore is required. Both the Facilitator and each spawned System Maintenance Utility job can be restarted.

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)	All the parameters are validated successfully.	N/A	N/A
Warning (4)	No Facilitator records found for the Run Number (Nothing to process).	The reason for the missing records must be investigated before a new chain job is run.	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. Sample Message: SLEEP_TIME is required to run the Facilitator.	Schedule a new job after entering a valid value for the parameter.	N/A
	Entered parameters are not valid. Sample Message: The SLEEP_TIME must be an integer and > 0.	Schedule a new job after entering a valid value for the parameter.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A

Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Facilitator Job Processing

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	SMU Job Parameter File not found.	If all files are missing, the reason must be investigated before a new chain job is run. If the files can be located and moved to the expected location, the job can be restarted.	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	This return code will be issued when any spawned SMU job fails. Sample Message: SMU Job – xxx – Failed	Determine the cause for the spawned SMU job to fail and resolve. The spawned job can be restarted independently if desired, and restarting the chain job will also restart any failed SMU jobs.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

2.1.30 Record-Specific Alerts

Chain or Job Name	Record-Specific Alerts
Recommended Frequency	The Record-Specific Alerts batch job can be run daily as part of the nightly cycle or on demand.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

The Record-Specific Alerts batch process in CGI Advantage Financial is a process which notifies/ sends emails to users based on entries on the Record-Specific Alerts (RSALRT) page. The job selects all eligible Record-Specific Alerts by comparing the Alert Date field found on RSALRT with the Alert Date job parameter. If an RSALRT record has a blank Alerted Date field and an Alert Date that is equal to or less than the job parameter, that record is selected.

For each selected record, an email is sent to the respective recipient (Recipient Email set on RSALRT) by using other email details provided on record. After sending emails successfully, the Alerted Date field is set to the system date on each selected record.

The job then goes on to do data maintenance with the Number of Days for Retaining Records parameter. If the RSALRT record has a past Alerted Date equal to or greater than the number of days specified, it is deleted. If the desire is not to delete records, then the job should be run with a very large number of days.

Since this job uses Alert Date and Alerted Date for selection of eligible records and at the end updates the Alerted Date to indicate the record has been processed, the same record will not be picked in the next run of the process. Hence this job is not restartable, but rather a new job is submitted if one should fail.

Of note are the two different kinds of emails generated by this job. There are some with an imbedded hyperlink that will take the user first to the login screen and then to the individual record. Other emails are generated without that hyperlink. The difference between the two stems from how the RSALRT records are created. When users add or upload them directly to the RSALRT page, there will not be a hyperlink. Those areas of the application where the RSALRT record is created by another action (for example, transaction processing) will have the hyperlink because that action has the logic to build the link.

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters • If Alert Date parameter is not provided then “Alert Date is defaulted to Application Date” will be displayed in log. • If Alert Date parameter is not valid on CLDT then ‘The Alert Date is not a valid date on the Calendar Date (CLDT) table’ will be displayed in log. • If Alert Date parameter is not provided in valid MM/DD/YYYY format then ‘The Alert Date must be in the mm/dd/yyyy format’ will be displayed in log. • If From Email Address parameter is not provided or is not a valid email address then ”Please enter a single valid From

	Email Address” will be displayed in log. • If Commit Size parameter is not provided then “Commit Size is defaulted to 1000” will be displayed in log. • If Commit Size parameter is not a valid integer greater than zero then ’Commit Size must be a positive whole number greater than zero” will be displayed in log. • If Number of Days for Retaining Records parameter is not provided or is not valid integer then ‘A valid Number of Days for Retaining Records is required’ will be displayed in log. • If Progression Counter parameter is not provided then “Progression Counter is defaulted to 1000” will be displayed in log. • If Progression Counter parameter is not a valid integer greater than zero then “Progression Counter must be a positive whole number greater than zero” will be displayed in log. • Parameter validation completed.
2. Selection of Records	• Selecting eligible records • Selection of eligible records completed
3. Processing of Selected Records	• Processing the selected records • If Current Application URL (CURR_APP_URL) is not mentioned on APPCTRL then ‘Current Application URL is not set on APPCTRL table’ will be displayed in log. • If Current Application URL (CURR_APP_URL) is not mentioned on APPCTRL then ‘Hyperlink to navigate on specific record is not generated and added on email’ will be displayed in log. • If Email creation or sending of email fails then ‘Unable to create emails’ and ‘Mail exception occurred while trying to send email to user’ will be displayed in log. • Process is terminated as at least 10 records failed to send emails. • Depending of progression Counter size the ‘<<Progression Count>> record(s) processed’ will be displayed in log. • If no records are selected to process then ‘No records selected for processing’ will be displayed in log. • Total <<Total Count>> record(s) are processed • Processing completed • Deleting record(s) from Record-Specific Alerts table prior to date <<Date calculated based on Number of Days for Retaining Records parameter >> • <<Delete Record Count>> " record(s) are deleted from Record-Specific Alerts table

Major Input

Tables

- Record-Specific Alerts (R_SPFC_ALERTS)

Batch Parameters

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

Parameter	Description	Default Value
ALRT_DT	Optional The Alert Date (valid format: MM/DD/YYYY) is used for record selection when the current Application Date is not desired for selection.	(blank)
EMAIL_MSG	Optional Email Salutation/Message to be printed at the bottom of a generated email. The salutation is limited to 300 characters and certain email programs may limit that further.	(blank)
FROM_EMAIL_AD	Required From Email Address that will be placed on generated emails. The format of this address has to match what is acceptable for the email program used.	(blank)
COMMIT_SIZE	Optional Commit Block Size is a performance parameter used to control the number of records saved/updated.	1000
NO_DAYS_RETAIN	Required The Number of Days for Retaining Records is used to age and delete processed alerts if they are as old as or older than this number of days. If set to 0, all processed records will be immediately deleted.	365
PROG_CTR_SIZE	Optional Progression Counter is a parameter that controls messaging to the job log stating the progress of the job.	1000

Major Output

- Record-Specific Alerts (R_SPFC_ALERTS)
- Generated emails

Job Return Code

Return Code	Condition
Successful (1)	All of the selected records are processed successfully by sending emails and updating records for the Alerted Date.
Warning (4)	No eligible records found. This could be because of the following reasons: • No records found on the Record-Specific Alerts (RSALRT) table. • No records found on the Record-Specific Alerts (RSALRT) table for the Alert Date parameter value.
Non Fatal Error (8)	Unable to update Alerted Date on processed Record-Specific Alerts (RSALRT) record.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid. • Unable to update Alerted Date on processed Record-Specific Alerts (RSALRT) record. • Unable to create/ send email for selected Record-Specific Alerts (RSALRT) record. • At least 10 records failed to send an email due to unavailability or invalid server. • 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

Alert Date (ALRT_DT) in ascending order.

Selection Criteria

- The records will be selected only if Alerted Date is not populated on 'Record-Specific Alerts' and
- The selection based on Alert Date will be as follows:
- If the Alert Date parameter is provided and this date is greater than the Application Date (on APPCTRL) then the records from 'Record-Specific Alerts' will be selected by matching (equal to) Alert Date on record with job parameter value.

- If the Alert Date parameter is provided and this date is less than or equal to Application Date (on APPCTRL) then the records from 'Record-Specific Alerts' will be selected where Alert Date on record is less than or equal to the Alert Date parameter value.
- If the Alert Date parameter is not provided then the records from 'Record-Specific Alerts' will be selected where Alert Date on record is less than or equal to the Application Date (on APPCTRL).

Problem Resolution

The following table shows the possible return codes and recommendations.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully and all of the selected Record-Specific Alerts records are processed successfully.	N/A	N/A
Warning (4)	Job ended with a Warning because no eligible record is selected for processing. **Sample Message:** No records selected for processing.	Provide Alert Date parameter such that at least one record exists on Record-Specific Alerts page with the Alert Date less than or equal to the job parameter value for which the Alert/ Notification is not yet sent.	Possibly a valid return code with no action required if there are no alerts for a given day.
Non Fatal Error (8)	Unable to update Alerted Date on processed Record-Specific Alerts (RSALRT) record	The reason for the failure needs to be investigated before scheduling a new job.	N/A
Failed (12)	Required Parameters are not entered. **Sample Message:** A valid Number of Days for Retaining Records is required.	Enter Number of Days for Retaining Records (valid numeric value)	N/A
	Unable to create or send emails. **Sample Message:** Unable to create emails. Mail exception occurred while trying to	The reason for the failure needs to be investigated before scheduling a new job.	N/A

	send email to user. Unable to send emails due to unavailability or invalid server. **Sample Message:** Process is terminated as at least 10 records failed to send emails.	The reason for the failure needs to be investigated before scheduling a new job.	N/A
	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

2.1.31 Submit Ready Transactions

Chain or Job Name	Submit Ready Transactions
Recommended Frequency	Daily
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

The Submit Ready Transaction batch job will search for combinations of Transaction Code and Create User ID that exist on the Transaction Catalog unprocessed. These could be interfaced, uploaded, system-generated, or manually entered transactions. The key is that the transactions have the Transaction Status of *Ready*. For transactions selected, the process schedules a System Maintenance Utility (SMU) batch job for each Transaction Code and Create User ID combination. Each SMU completes before the next is scheduled.

It is critical to note that the only Transaction Status selected is *Ready.* If there is a set of transactions in need of submitting but the Transaction Status is *Rejected*, then if the reason(s) for the rejection have been addressed, then a separate SMU must be run with the Mark for Processing action to change from *Rejected* to *Ready*. Alternatively, if the Automatic Transaction Correction process had run to correct the reason for rejection, the SMU as Mark for Processing is still required to change the Transaction Status.

As the batch job creates SMU jobs with the Submit Transaction action, all SMU parameters available for that action are available for this job. If the need arises to use a parameter not delivered with this job, use the Batch Setup page to add the parameter. Please ensure that the parameter name added matches that which SMU expects. Please see System Maintenance Utility documentation for those.

While an on demand run is possible, the anticipated run frequency will be after one or more transaction imports have occurred. When scheduled to run with a frequency, one or both of the selection parameters will be a default set defined on Batch Setup or in a job scheduler.

*Although delivered as a single instance job, change that setting in Batch Setup to allow multiple concurrent instances. Take care with selection parameters in this situation so no two jobs select the same set of transactions, as one will record many failures.

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters • Parameters are valid or invalid depending on the validation **Sample Message**: Create User ID is not valid within the application • Parameter validation failed (or) Parameters successfully validated
2. Selection / Submitting Transactions	• Submitting the ready transactions for the Transaction Code list passed.

	(or) No matching transactions found to submit. • For Each Create User ID and Transaction Code where Ready Transactions are found: • Creating submit parm file for Transaction Code <Transaction Code> and Create User ID <user ID>. • Setting up job for parameter file: <file name> • Job completed for child <JobID> and job return code is <Return Code>.

Major Input

- Transaction Catalog (DOC_HDR)

Batch Parameters

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

Parameter	Description	Default Value
Parameter Directory (AMSPARM)	A required Parameter file directory where files are saved for use by the SMU job(s) spawned.	$$AMSROOT$$/Parms
Created By User ID (CREA_USER_ID)	An optional selection parameter for the Created By header field. Multiple values are allowed, if separated by commas. A wild card (*) can be used, as long as a minimum of 3 characters is provided. See footnote	No Default
Transaction Code (TRANS_CD_LIST)	An optional selection parameter for Transaction Code. Multiple values are allowed, if separate by commas. A wild card character * can be used, as long as a minimum of 3 characters is provided. See footnote	No Default
Submit User ID (SBMT_USER_ID)	An optional parameter to specify a User ID in the submit process to become the Last Modified By user instead of using the User ID that submitted the batch job.	No Default
SMU Catalog ID (SMU_CTLG_ID)	The required Catalog ID for the SMU job to spawn.	3
Sleep Interval (SLEEP_CTR)	The required number of seconds to wait between SMU job status checks.	5
Progression Counter (PROG_CTR)	A required definition of a number of times that SMU job status is checked before a count is reported in the batch job. If -1 is	-1

	set, there will be no progress messages. If left blank, -1 will default.	

Although both selection parameters are not required, do not run the process without at least one if not both unless it is acceptable to pick up all transactions of all transaction types for all users. If run without the proper selection criteria, work-in-progress transactions entered online that are error free would be selected.

Major Output

The specific output is based on what transaction types were selected, whether or not the submit action was successful, and whether or not the submit action resulted in the transaction going to pending or final.

Job Return Code

Return Code	Condition
Successful (1)	All spawned SMU jobs complete successfully. The job also completes successfully when no eligible transactions are found.
Warning (4)	This job completes with a warning when the maximum return code of all SMU jobs is warning.
Non Fatal Error (8)	This job completes with as non-fatal when the maximum return code of all SMU jobs is non-fatal.
Failed (12)	This return code will be issued under the following conditions: • 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, application server, or network issues.

Sort Criteria

If the Transaction Code List parameter value is used, transactions submit in the same order as listed. If the Transaction Code List parameter value is not used, transactions submit in order by the Transaction Codes selected.

Selection Criteria

- Transaction Status (DOC_STA_CD) – Ready (2)
- Create User ID (DOC_CREA_USID) – per Create User ID parameter value, if provided
- Transaction Code (DOC_CD) – per Transaction Code List parameter value, if provided

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All SMU jobs finished successfully or no eligible transactions were found.	If there were anticipated transactions, check parameters or that the Transaction Status of the anticipated transactions is *Ready* and not *Rejected.*	N/A
Warning (4)	One or more SMU jobs returned Warning (4). **Sample Message:** Job completed for child <job id> and job return code is 4.	Determine the transaction with the problem from the SMU log message, correct the conditions, and either submit manually or reset the Status to *Ready* for inclusion in the next run.	N/A
Non Fatal Error (8)	One or more SMU jobs returned Non Fatal Error (8). **Sample Message:** Job completed for child <job id> and job return code is 8.	Determine the transaction with the problem from the SMU log message, correct the conditions, and either submit manually or reset the Status to *Ready* for inclusion in the next run.	N/A
Failed (12)	Job failed due to Fatal conditions. **Sample Message:** Invalid Submit User ID	If the job fails because of any runtime exceptions or parameter errors, investigate the exception/error reported by the process, resolve the error, and schedule another job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated, and a new job can be scheduled.	N/A
System Failure (20)	Job is terminated because of database server or network issues.	The reason for the system failure needs to be investigated, and a new job can be scheduled.	N/A

2.1.32 Sync Transaction ECM Reference

Job Name	Sync Transaction ECM Reference
Recommended Frequency	On Demand
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Exception report generated for transactions if a match is not found in the ECM application.

Overview

The Sync Transaction ECM Reference job creates an attachment reference on the Advantage transaction that is created after the attachment is loaded in the Enterprise Content Management (ECM) system outside Advantage and associated to the Advantage transaction by populating metadata attributes manually on the ECM attachment.

The process reads transactions based on the input parameters, searches for an attachment in the ECM system using transaction data, and creates an attachment record if a match is found in the ECM system. Attachment description on the Catalog table (IN_OBJ_ATT_CTLG) is set to the description provided as a parameter to indicate that the attachment reference was added by the batch process.

Prior to searching for an attachment in the ECM system, the process first checks for an existing attachment on the transaction with a matching attachment description provided as a batch parameter. If a match was found, the transaction is skipped from the processing to avoid duplicate attachment references.

An exception report and a CSV file are generated (when the batch parameter set to Y-yes) for the transactions that do not have a matching attachment in the ECM system. Sites can use the report and/or the CSV file to identify discrepancies, correct the metadata in ECM system, and then run the batch process again to create attachment references correctly in the Advantage application.

Warning: Sites should NOT use this process if their business model does not meet the criteria described in the overview section. Not doing so may result in unwanted attachment references on transactions and corrupted application data.

Steps

Process Steps	Messages
1. Parameter Validation	• Parameters are valid or invalid depending on the validation. If a parameter is invalid, an appropriate message issued.
2. Read and process transactions	• Read matching transactions from transaction table provided in the batch parameter. • If the selection returns 0 records, then the following message will be issued: “No matching records found for processing”. • At the end, the following messages are issued: • “Number of <Transaction Code> Transactions

Process Steps	Messages
	processed:" <count> • "Number of <Transaction Code> Transactions linked:" <count> • "Number of <Transaction Code> Transactions already linked and skipped:" <count>
3. Verify attachment reference already created	• Read existing attachments for the transaction from IN_OBJ_ATT_DOC_REF table. • If there are no existing attachments, continue searching for attachment in ECM system for the transaction. • For each attachment found, read entry from attachment catalog (IN_OBJ_ATT_CTLG) table matching OBJ_ATT_UNID value. • Match the attachment description on the attachment catalog entry with the value provided in the input parameter. • If description matches, increment the skipped count, and skip the transaction from further processing. • If description doesn't match, continue searching for attachment in ECM system for the transaction.
4. Search for attachment in ECM	• Read ECM type configured for Adv Type provided in parameter value. • Read Advantage Type to ECM Type attribute mapping details for Advantage Type and Transaction Header Component table derived based on the parameter value. • Build a search request populating search attribute with data values extracted from transaction. • Initiate search request. • If the response fails, the following message is logged: "Error occurred while searching in ECM:"
5. Add Attachment reference	• Parse the ECM search request response. • If an attachment is available for the transaction, the following message is logged: "Attachment found for "<Transaction Details> "Object ID: "<ECM Object ID> "File Name: "<Attachment File Name> • Derive the transaction header component and read transaction header record. • Create attachment entry on the transaction header component with the response received from ECM system.
6. Generate Exception Report and CSV file	• If generate Report and CSV parameter set to true, Exception report generated for transactions attachment is not available in the ECM system. • A CSV file generated for the exception transaction and included all the columns provided in batch parameter. • Standard messages are issued in the log for exception report generation.

Major Input

- Transaction query component table provided in batch parameter.
- Attachment Transaction Reference table (IN_OBJ_ATT_REF).
- Attachment catalog table (IN_OBJ_ATT_CTLG).

Batch Parameters

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

Parameter	Description	Default Value
ATT_DESCRIPTION	The required Attachment Description matched against existing attachment records to avoid duplicate attachment references. This value is on new references added into the Advantage. **Note**: Sites must configure this value on the Batch Setup (BATSETUP) once and must NOT change the value between the batch run.	Transaction submitted through ECM
GEN_EXCP_CSV_RPT	A required Generate Exception/CSV report parameter where (Y - Yes N - No)	N
LOOKUP_ECM_ADV_TYPE	A required Advantage ECM Transaction Type parameter used for search attachments in ECM system. This parameter used to derive corresponding ECM type mapped for searching attachments.	(No Default)
REPORT_NAME	A required Report Name parameter to appear on the exception report.	Sync ECM Reference Exception Report
SELECT_COLUMNS	A required Select Columns parameter used for record selection. Enter comma separated values and must include DOC_CAT, DOC_TYP, DOC_CD, DOC_ID, DOC_DEPT_CD, DOC_VERS_NO for successful execution.	DOC_CAT, DOC_TYP, DOC_CD, DOC_ID, DOC_DEPT_CD, DOC_VERS_NO, DOC_CREA_USID, DOC_CREA_DT
TRAN_CODE	A required Transaction Code(s) parameter used for	No Default

	record selection. Enter comma separated for multiple values.	
TRAN_CREA_FROM_DT	A required Transaction From Date (inclusive) parameter used for record selection. Valid format is MM/DD/YYYY.	No Default
TRAN_CREA_TO_DT	A required Transaction To Date (inclusive) parameter used for record selection. Valid format is MM/DD/YYYY.	No Default
TRAN_PHASE_CD	A required Transaction Phase parameter used for record selection where (1 - Draft, 2 - Pending, 3 - Final). Enter comma separated for multiple values.	1,2,3
TRAN_QUERY_COMP	A required Transaction Query Component parameter used for record selection.	No Default

Major Output

- Attachment Transaction Reference table (IN_OBJ_ATT_REF).
- Attachment catalog table (IN_OBJ_ATT_CTLG).
- Attachments storage table (IN_OBJ_ATT_STOR).

Sort Criteria

None

Selection Criteria

The process uses the following selection criteria.

From the transaction table specified in the transaction query component (TRAN_QUERY_COMP) parameter where:

- Transaction create date (DOC_CREA_DT) >= batch parameter (TRAN_CREA_FROM_DT) value.
- Transaction create date (DOC_CREA_DT) <= batch parameter (TRAN_CREA_TO_DT) value.
- Transaction Phase (DOC_PHASE_CD) matches batch parameter (TRAN_PHASE_CD) values.

Problem Resolution

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

Return Code	Condition
Successful (1)	The job ends as successful when matching records found and all the records processed successfully.
Warning (4)	A warning results when there no records selected for the matching criteria.
Non-Fatal Error (8)	No scenario.
Failed (12)	The job fails under the following conditions: • Parameters are invalid. • Unable to invoke Enterprise Content Management (ECM) search request. • Unable to parse the search request response. • Unable to add an attachment reference to the transaction record. • Run time exceptions for unexpected situations.
Terminated (16)	The job is terminated by the user.
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues.

2.1.33 System Maintenance Utility

System Maintenance Utility (SysManUtil) is the primary utility used to maintain table data and transactions in CGI Advantage Financial.

For detailed information on this job (such as when to run, input, output, and process parameters), refer to the “System Maintenance Utility” section in the *CGI Advantage – Administration Utilities Run Sheets* guide available on the User Guide Downloads page in the online help.

Please refer to the “Working with System Maintenance Utility” section in the *CGI Advantage – System Processing User Guide* available on the User Guide Downloads page in the online help for more information on the features, maintenance of tables, maintenance of transactions, and setting up parameters.

2.1.34 Temporary Table Purge Process

The Temporary Table Purge process purges the COA Usage Verification, Transaction Reference Lifecycle, and Vendor Transaction History temporary tables, restoring them to an empty state after each run. Normally the data in the Transaction Reference Lifecycle and Vendor Transaction History tables will be removed for a user session each time a new search is performed on the LINQ and VTH pages respectively. This job is used to remove accumulated records in the tables to immediately reclaim the space and help ensure best available performance.

- Each record on the COA Usage Verification table (COA_USG_VER_TMP) tracks information about whether a single COA element has been successfully used in an accounting-based or budget transaction. This table is only used if the COA Deletion Prevention feature is enabled. When the feature is enabled, any attempt to delete a COA element will create a record on this table and invoke the COA Usage Verification batch process to update the record after researching the COA element's usage. Records on the COA Usage Verification table typically expire (are no longer considered valid) soon after the verification batch process completes (see APPCTRL "COA Deletion Verification Minutes" parameter).
- The Transaction Reference Lifecycle table (R_DOC_RF_TEMP) is a temporary table that will provide the user with a snapshot of a related transaction within a transaction chain. Each search action (Backward or Forward) performed on the Transaction Lifecycle Inquiry (LINQ) page will add records to this table temporarily.
- The Vendor Transaction History table (VEND_TRAN_HIST) is a temporary table used to store records from the accounting line tables for Transaction Types ABS, AD, CR, DC, JV, MD, PO, PR, PYRL, RE, and RQ during queries on the Vendor Transaction History (VTH) page. The records are stored indefinitely to allow sorting and downloading query results. Each query deletes any existing rows for a given session, but the records from the most recent query remain until the session is reused and another query is made.

When to Run

This job should be run daily.

Major Input

N/A

Output

- COA_USG_VER_TMP table
- R_DOC_RF_TEMP table
- VEND_TRAN_HIST table

Parameters

N/A

Sort Criteria

N/A

Selection Criteria

All rows in the COA_USG_VER_TMP, R_DOC_RF_TEMP, and VEND_TRAN_HIST tables are selected for deletion.

Troubleshooting

No database restore is required. Correct the problem and rerun the job executing the program. No restoration of data sets or files from backups is required for this program.

2.1.35 Transaction Archiving

Description

Transaction Archiving is an extension of the SMU DOCARCHIVE action. Transaction Archiving makes it more flexible for users to archive several transactions without having to specify individual transactions. In addition, users can specify multiple sets of search criteria for transactions to be archived.

The Transaction Archive chain job can alternatively be used to purge transactions instead of archiving them to XML. The system provides two different parameters on Application Parameter (APPCTRL) to control XML creation and Archived Transaction Catalog update. Allow Transactions to Write to XML File (ALLW_TRSNCTN_WRT_XML_FL) can be set to *Y* for XML file creation. Without this file, transactions cannot be restored. Allow Transaction to Archived Transaction Catalog (ALLW_TRSNCTN_ARCH_CTLG_FL) can be set to *Y* to update the Transaction Archive Catalog. Without this update and the XML file, a user cannot preview the archived transaction.

Allow Transaction to Archive Transaction Catalog	Allow Transaction to Write to XML File	Implementation Considerations
Y	Y	Archived transactions will be listed on the Archived Transaction Catalog, support the View action there, allow for the Restore action, and remain in Advantage Business Intelligence.
Y	N	Archived transactions will be listed on the Archived Transaction Catalog and remain in Advantage Business Intelligence.
N	Y	This is not a recommended setting as the XML file without the Archive Transaction Catalog provides no features.
N	N	These settings are used by those that do not want any of the Archive Transaction Catalog features nor use Advantage Business Intelligence.

Unlike several other archiving jobs, this chain does not implement the Data Warehouse Archived Record Queue process, which is used to retain the table names and key values for records that are archived and deleted. The Advantage Business Intelligence reporting system uses this information to distinguish between records that have been archived (as valid historical data) before deletion and records that are simply deleted from the system (as no longer needed). When transactions are archived, the Transaction Archive Catalog (DOCARCH) is used to determine which transactions have been archived rather than relying on the additional Data Warehouse Archived Record Queue tables.

While running the job for purging transactions it is recommended to run the job initially in report mode before running the job in update mode to review the selected transactions since purging a transaction will make it irrecoverable, unlike archival.

When to Run

On request.

Major Input

Input Parameter file. The format of the file is:

```
BLOCK_SIZE=100000
ACTN_CD=DOCARCHIVE
_PARM_LINE_
DOC_CD=ESMT
DOC_DEPT_CD=010
DOC_LAST_DT<=01/01/2000
_PARM_LINE_
DOC_CD=ADDR
DOC_LAST_DT<=SYSTEM_DT
_PARM_LINE_
DATA_OBJ=ABS_DOC_ACTG
DOC_CD=GAX
DOC_DEPT_CD=010
DOC_LAST_DT<=05/01/2000
```

Note: If the reserved word DATA_OBJ is used then any field that is listed on that corresponding table can be included in the parameter file. If a Data Object is not specified within a PARM_LINE, then the system will select based on values on the Transaction Header. Any field that is listed on the General Transaction Catalog (DOC_HDR) table can be included in the parameter file:

- Transaction Code
- Transaction ID
- Transaction Department Code
- Transaction Version Number – (applicable when Action Code = DOCARCHIVE)
- Transaction Unit
- Transaction Function
- Transaction Phase – Valid values are Final or Historical Final
- Transaction Status
- Transaction Date

Examples of both options are illustrated in the above parameter file example.

Valid values for ACTN_CD are DOCARCHIVE and DOCARCHIVEHIST. (If the ACTN_CD in the input parameter file is DOCARCHIVEHIST, then the Archive Restore ID parameter in the Facilitator step must be changed to 4). In addition to the input parameter file, several batch parameters exist. General rules for the parameters are as follows:

1. The Selector program uses Reserved Words to insert values into the Input Parameter. For Date fields the Reserved Words of SYSTEM_DT, APPLICATION_DT, and PARM_DT may be used. Additional Reserved Words PARM_VAL, PARM_VAL2, and PARM_VAL3 can be used in Non-Date fields.
    a. SYSTEM_DT - This reserve word sets the selection date field to the System date.

b. APPLICATION_DT - This reserve word sets the selection date field to the Application date.
c. PARM_DT - This reserve word sets the date to a selection date field that is entered in the Online parameter.
d. PARM_VAL - This reserve word sets the field to a value that is entered in on the Online parameter.
e. PARM_VAL2 - This reserve word sets the field to a value that is entered in on the Online parameter.
f. PARM_VAL3 - This reserve word sets the field to a value that is entered in on the Online parameter.

2. An additional reserved word DATA_OBJ is used to specify the data object which will be used to make the selection of transactions.
3. The Selector process handles <, <=, =, >=, >, and <> clauses for the Date parameter only.
4. The Selector program does addition and subtraction calculations for the Date parameter only.
5. If the Bypass Flag parameter is set to N, the system will invoke processing to validate the transactions selected by the input parameter file against transaction-specific archive rules, e.g., that the transaction is fully referenced and eligible to be archived. If the Bypass flag is set to Y, then these rules will be bypassed. (Note: In this case, it is possible that the report produced from the Selection process will include more transactions than were actually archived). In order for the transaction-specific validations to be invoked when the Bypass Flag parameter is set to N, a transaction Data Object must be specified in the parameter file, e.g., DATA_OBJ=ABS_DOC_HDR. If no data object is specified, the transaction validations will not be called, regardless of the Bypass Flag parameter value.
6. The File Location parameter specifies the location where both the SMU parameter file generated by the selection process, as well as the transaction XML files, will be written. Note: When run in purge mode, the XML files will not be created since the transactions would be completely purged.
7. The SMU parameter file created by the Selector Program includes specific lines for each transaction.

Output

When run in Archive mode, one XML file will be created for each transaction. Related transaction attachments and comments will be saved, and both file types will be compressed into a single Zip file per transaction. Once the data is exported all of the files are stored to a system archive folder. Additionally, two reports are generated by the selector process – one for transactions selected to be archived and the other for transactions that were selected but failed the archive eligibility validation. This report will only be produced if the Bypass Flag is set to N. The reports will be produced even when run on purge mode for users to review the transactions selected before running the process in update mode.

Parameters

Batch Parameters

Job	Parameter	Description	Default Value
Transaction Selector	PARM_DT	PARM_DT. Enter the date for the transactions to be archived. The format should be	Null

		MM/DD/YYYY.	
Transaction Selector	PARM_FILE	PARM_FILE. Required. Enter the name and exact location of the parameter file for Transaction archiving.	Null
Transaction Selector	PARM_VAL	PARM_VAL. Enter the value for the transactions to be archived.	Null
Transaction Selector	PARM_VAL2	PARM_VAL2. Enter the value for the transactions to be archived.	Null
Transaction Selector	PARM_VAL3	PARM_VAL3. Enter the value for the transactions to be archived.	Null
Transaction Selector	BYPASS_FL	BYPASS_FL. Required. Specify if the validation for transactions needs to be bypassed. Valid values are Y or N.	N
Transaction Selector	REPORT_ONLY	REPORT_ONLY. Required. Specify if the job needs to run in report only mode. Valid values are Y or N.	N
Transaction Selector	FILE_LOCATION	Output File Location. Required. Enter the file location for the output parameter file and XML files to be read by SMU.	Null
Archive-Restore Facilitator	ARCHIVE_RESTORE_ID	<Non Overridable> Archive Restore ID	3
Archive-Restore Facilitator	COMMIT_BLOCK_SIZE	<Required> The commit block size	1000
Archive-Restore Facilitator	PROCESSOR_NO	<Required> The number of	2

		processors. This determines the number of SMU jobs to be started simultaneously.	
Archive-Restore Facilitator	SLEEP_TIME	<Required> The Sleep time for the facilitator in milli seconds.	5
Archive-Restore Facilitator	UPDATE_STATUS	<Non Overridable> Update the facilitator table.	Y

Note: PARM_VAL, PARM_VAL2, and PARM_VAL3 cannot be used with date columns. This is because if the parameter file has something like this 'DOC_REC_DT = PARM_VAL2', the system will just ignore this filter because it knows DOC_REC_DT is a date field and the only parameters that are checked against a date column are APPLICATION_DT, SYSTEM_DT, and PARM_DT.

Sort Sequence

DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO

Selection Criteria

The selector program will generate a where clause based on the input parameter file and retrieve all the transactions matching the selection criteria. Again, if the Bypass Flag is set to N, then some of the transactions initially selected may be eliminated if they do not pass transaction-specific archive eligibility rules.

Additionally, the system will include certain assumptions in its selection logic, meaning that the following items do not need to be specified in the input parameter file:

- Transactions will not be selected if they are in one of the following phases:
  - Draft Phase
  - Pending Phase
  - Final Phase with a subsequent draft or pending transaction existing
- Transactions will not be selected if they are a Template or if a transaction is pending Future Transaction Triggering processing.
- Only transactions which have completed journal posting will be selected if the Posting Line Catalog exists on their database. If the Posting Line Catalog does exist, the program will only select transactions where the Journal Posting Indicator is set to true.

Problem Resolution

If the Transaction Archive job runs unsuccessfully, correct the problem and start a new job.

2.1.36 Transaction Message Update

Chain or Job Name	Transaction Message Update
Recommended Frequency	On Demand
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	None

Overview

The Transaction Message Update batch job extracts message information from transactions to the Transaction Message (DOCMSG) table. Messages (the Severe, Error, Warning, and Informational messages that are saved with transactions) for selected transactions are extracted from the transaction header tables where they are stored in a packed text format that is difficult to query. The message information is parsed into individual records and fields, which are then copied to the corresponding fields of a new DOCMSG record and saved providing a normalized representation of the messages that can be easily queried.

The output (accessed via the DOCMSG page) can be used to support the process of troubleshooting large numbers of rejected transactions. For example, suppose a chain job is responsible for generating and submitting 10,000 transactions in one execution. 9,500 of those transactions go through OK, but 500 fail due to some unexpected situation. The Transaction Message Update batch job can be used to extract and normalize the messages from those rejected transactions to expedite analysis and corrective action among the rejected transactions.

This job is not intended to provide a means of system-wide extraction of transaction message data on a regular basis. While the batch parameters do allow execution against all transactions in one execution, this should be reserved for extraordinary circumstances and will require several hours to complete with just moderate data sizes. The job is intended to be used for more focused analysis of transaction messages, whether on an ad hoc basis, or on a recurring basis if there are particular kinds of transactions that require frequent attention to large volumes of rejected transactions.

The job provides several parameters for flexible selection of transactions. When the job selects a transaction for which messages have previously been extracted, it will refresh the DOCMSG records by replacing the existing records (for the transaction) with the latest transaction message information from the selected transaction.

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters • Parameters are valid or not valid depending on the Validation.
2. Processing Transactions	• If no records are selected: • No Transaction Types met Include/Exclude selection criteria. • For each Transaction Type selected:

	• Deleting all existing Transaction Message records for Transaction Type '<Transaction Type>' (Run Mode Full only). • Updating from <Transaction Type> Transaction Header table. • Records processed: 1000 (continues by 1000) • Total records processed: <record count> • Updates complete.

When to Run

This job offers a high degree of flexibility in selection parameters to meet a diverse set of research and analysis needs. It can be run on an ad hoc, on-demand basis when need arises to research a particular subset of transactions. It can also be run on a periodic basis if there are identifiable subsets of transactions known to require research and resolution on a frequent basis. When run on a periodic basis for the same selection criteria, consider using Incremental mode with a recent Run Date.

As discussed above, the output from this job (accessed via the DOCMSG table) is intended to be used to analyze messages that occur frequently for known sets of transactions. Running this job against all transactions, or with any parameters that target a large set of Transaction Types, is strongly discouraged except for extraordinary circumstances.

Restartability Information

This job can be restarted if it fails due to any reason, as long as it reached the first database commit before interruption. During each database commit, the identifiers for the last transaction in the Transaction Message commit block are saved as a checkpoint.

A restarted job will begin by reprocessing all messages for the transaction specified in the checkpoint. This is because there may have been some Transaction Message records saved with the last successful commit, while others for the same transaction were to be included with the next.

Frequency

This job should be run on demand as needed for transaction message analysis.

Major Input

- All transaction header tables (<Transaction Type>_DOC_HDR) and DOC_ERR_OFLOW.

Batch Parameters

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

Parameter	Description	Default Value
COMMIT_BLOCK_SIZE	Commit Block Size Optional.	100
EXCL_DOC_PHASE	Exclude Transaction Phase List Optional. Comma delimited list of Transaction Phase values to be excluded from transaction selection. Enter 0 (No Phase), 1 (Draft), 2 (Pending), 3 (Final), 5 (Historical Final), 6 (Conflict Draft), and/or 7 (Template).	No Default
EXCL_DOC_STATUS	Exclude Transaction Status List Optional. Comma-delimited list of Transaction Status values to be excluded from transaction selection. Enter 1 (Held), 2 (Ready), 3 (Rejected), and/or 4 (Submitted).	No Default
EXCL_DOC_TYP_CD	Exclude Transaction Type/Code List Optional. Comma-delimited list of Transaction Type values and/or Transaction Type/Transaction Code values to exclude from transaction selection. Each Transaction Type and Transaction Code value should be delimited with a period. Example 1: DT1, DT2 Example 2: DT1/DC1.DC2.DC3, DT2/DC4.DC5.DC6	No Default
EXCL_SEVERITY	Exclude Severity List Optional. Comma-delimited list of Message Severity values to be excluded from transaction selection. Enter 0 (Informational), 1 (Warning), 2 (Error), and/or 3 (Severe).	No Default
FROM_DT	From Date Required for Run Mode 3 (Interval); otherwise prohibited.	No Default

	Use format MM/DD/YYYY.	
TO_DT	To Date Required for Run Mode 3 (Interval); otherwise prohibited. Use format MM/DD/YYYY.	No Default
INCL_DOC_TYP_CD	Include Transaction Type/Code List Optional. Comma-delimited list of Transaction Type values and/or Transaction Type/Transaction Code values to include in transaction selection. Each Transaction Type and Transaction Code value should be delimited with a period. Example 1: DT1, DT2 Example 2: DT1/DC1.DC2.DC3, DT2/DC4.DC5.DC6	No Default
PURGE_RECS	Purge Records Optional for Run Mode 2 (Incremental); otherwise prohibited. Enter true or false.	false
RUN_DT	Run Date Optional for Run Mode 2 (Incremental); otherwise prohibited. Use format MM/DD/YYYY.	No Default
RUN_MODE	Run Mode Required. Enter 1 (Full), 2 (Incremental), or 3 (Interval). If Incremental mode is selected, only transactions are included that have been edited, validated, or submitted after the Run Date parameter value. If Interval mode is selected, only transactions are included that have been edited, validated, or submitted between the From Date and To Date parameter values.	2

Major Output

- DOC_MSG – Records are deleted from and inserted into the Transaction Message (DOCMSG) table.

Job Return Code

Return Code	Condition
Successful (1)	All of the selected transactions are processed successfully.
Warning (4)	This step does not issue this return code.
Non Fatal Error (8)	This step does not issue this return code.
Failed (12)	The job will fail under the following conditions: • 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 Criteria

None. Transactions are processed by Transaction Type, Transaction Code, Transaction Department Code, and Department ID so that the checkpoint can be stored upon each commit to the database.

Selection Criteria

Transactions are selected by inclusion by Transaction Type and Transaction Code (within Transaction Type). If the Include Transaction Type/Code List parameter value is blank, all Transaction Type values will be included.

Transactions are selected by exclusion for Transaction Type, Transaction Code (within Transaction Type), Transaction Phase, Transaction Status, and Message Severity.

In Run Mode 1 (Full), all records are deleted from the Transaction Message (DOCMSG) table for each Transaction Type and/or Transaction Code selected, then records from each Transaction Type's header table are added to DOCMSG.

In Run Mode 2 (Incremental), only transactions that have been edited/validated/submitted on or after the Run Date parameter are selected. If the Run Date parameter is blank, then the value is defaulted for each Transaction Type to the lowest value among all highest Last Modified Date values by Transaction Code. For example, if the highest Last Modified Date for a selected Transaction Type with two Transaction Codes is November 6 of last year for the first and December 8 of last year for the second, then November 6 becomes the default for Run Date. When the Purge Records parameter is set to true, all records in the DOCMSG table are deleted for the selected Transaction Type and Transaction Code values. When the Purge Records parameter is set to false, any existing DOCMSG records are deleted as each transaction is processed for that transaction only.

In Run Mode 3 (Interval), only transactions that have been edited/validated/submitted between the required From Date and To Date parameters are selected. Any existing DOCMSG records are deleted as each transaction is processed for that transaction only.

Problem Resolution

No database restore is required. The job can be restarted if it reaches the first database commit before it is interrupted, otherwise a new job must be started.

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)	All the parameters are validated successfully.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. Sample Message: Parameter Run Mode value must be '1' (Full) or '2' (Incremental).	Schedule a new job after entering a valid value for the parameter.	N/A
	Entered parameters are not valid. Sample Message: One or more values in the Exclude Transaction Status List parameter not valid (0 - Informational, 1 - Held, 2 - Ready, 3 - Rejected, 4 - Submitted).	Schedule a new job after entering a valid value for the parameter.	N/A

	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated before scheduling a new job.	N/A

Step 2: Processing Transactions

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the transactions are processed successfully.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following two conditions. • Encounters any runtime exceptions • Failed during restart • If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	N/A
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.	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 either be restarted or schedule a new job.	N/A

2.1.37 Transaction Unarchive

Description

Transaction Unarchive will restore the transactions from the xml file if the earlier Transaction Archive process was run in Archive mode.

Note: If the Transaction Archive process was run in purge mode, there were no XML files created so the transactions cannot be unarchived. For more information on running the process in Purge mode please refer to the Transaction Archive chain job run sheet.

When to Run

On request.

Major Input

XML files created by the Archive process.

Output

The transactions will be archived back to the database.

Parameters

Batch Parameters

Job	Parameter	Description	Default Value
Transaction Unarchive	ARCHIVE_RESTORE_ID	<Non Overridable> Archive Restore ID	5
Transaction Unarchive	COMMIT_BLOCK_SIZE	<Required> The commit block size	1
Transaction Unarchive	PROCESSOR_NO	<Required> The no of processors. This determines the number of SMU jobs to be started simultaneously.	2
Transaction Unarchive	SLEEP_TIME	<Required> The Sleep time for the facilitator in milli seconds	5
Transaction Unarchive	UPDATE_STATUS	<Non Overridable> Update the facilitator table.	Y

Transaction Unarchive	RUN_NO	<Required> The run no for which the XMLS will be restored	NULL

Sort Sequence

N/A

Selection Criteria

The Facilitator will read the records from the Facilitator table for a Run Number indicated by a parameter. The Facilitator will generate SMU processes to unarchive the records that have been identified as “Transaction Archive Complete". After the restore process is complete, the status will change to "Transaction ready for archiving". The amount of SMU processes that will be generated at one time will be dictated by PROCESSOR_NO parameter.

Problem Resolution

If the Transaction Unarchive job runs unsuccessfully, correct the problem and restart the job.

2.1.38 Vendor Customer Archiving

Chain Name	Vendor Customer Archiving
Recommended Frequency	On Demand Before running, all sync cycles jobs should have finished their execution before this chain process is executed.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Vendor Customer Archiving Statistics Report for Financial Vendor Customer Archiving Facilitator Report

Overview

With the progression of time, the buildup in the vendor customer setup and associated transaction data in Advantage Financial becomes considerable. While such data is necessary for reporting and online inquiry, if the vendor customer has not been used after a period of time and there are no outstanding transactions, the data is no longer needed in the online application. The Vendor Customer Archiving process identifies eligible records based on a set of selection criteria and archives and deletes data from multiple tables in Advantage Financial only. Once the eligible records have been determined, they are written to the Vendor Customer Archive (VCARCH) table in Financial. If the Vendor Self Service (VSS) application is in use, the VCARCH table is used to communicate the list of Vendor/Customer codes to be archived between the Financial and VSS applications, and to indicate whether the records archival and deletion have completed in the Financial and VSS applications. The data remains in VSS until a separate job, Vendor Customer Archiving VSS Updates, is completed for a full run mode synced to VSS. Even after records have been archived and deleted from the online application, the data still remains in infoAdvantage.

The following job steps comprise the Vendor Customer Archiving chain process:

- Archiving Preprocessor
- Archive Facilitator
- Post Archiving Process

The steps are all singleton jobs. This means that only one of their instances can be run at a time. The Archive Facilitator does have the ability to launch System Maintenance Utility jobs in parallel. It is recommended to perform a database backup before the chain is run in *Full Run* mode.

Before any archive of data, it is assumed that research has been done to see that all activity has completed and that there will be no further need to record any more activity against a Vendor/Customer. The VCARCH page can be used to review Vendor Customer codes, Taxpayer IDs, or 1042-S Recipient Numbers deemed eligible to be archived after the Vendor Customer Archiving chain is run in Report Only mode, and it is just one tool that can be used in this review. This can provide the end user the opportunity to review the identified records before they are archived or removed from the system.

The Archiving Preprocessor step of the chain takes into account the user-specified parameters with other pre-defined selection criteria when searching for records that are eligible to be archived. It searches for Vendor/Customer codes, 1042-S Recipient Account Numbers, Taxpayer ID and TIN Type combinations, and User IDs eligible to be archived. In order for a Vendor/Customer code to be eligible, the corresponding Vendor/Customer (VCUST) record must satisfy all of the following conditions:

- Never Archive = No

  - Last Usage Date < Last Usage Date parameter value
  - Internal Account = No or Exclude Internal Account parameter value = No
  - Miscellaneous Account = No or Exclude Miscellaneous Account parameter value = No
  - Does not have any related Vendor User Information (R_VEND_USER) records, or if they exist, all of them have a Last VSS Login Date < Last Usage Date parameter value

If the Vendor/Customer code passes all of the above conditions, it is a candidate, but the job still needs to verify if the vendor customer has any outstanding activities yet to be completed. It must also answer “no” to all the following conditions:

- Is there any Disbursement Request (DISRQ) record where the Vendor Code or Payee Code matches the candidate?
- Does the candidate have any retainage not yet released or forfeited?
  - Is there any Retainage Summary by Commodity Line (RTGSUM) record where the Vendor Code or 3rd Party Vendor matches the candidate and the Total Retainage Outstanding is greater than zero? –or-
  - Is there any Collateral Management (COLLM) record with Active = Yes where the Award Vendor Code matches the candidate?
- Does the candidate have any Check Writer payments or intercepts not yet completed?
  - Is there any Check Writer Payment (CWPYMT) where the Vendor Customer matches the candidate?
- Is there any Unprocessed Procurement Card Activity (PRCUU) record with Pay flag = Yes, Delete flag = No, hidden Process flag = No where the Merchant ID is specified and matches the candidate’s Merchant ID?
- Does the candidate have any intercept in progress?
  - Is there any Intercept Disbursement Request (INCT_DISB_RQST) record where the Vendor Customer Code matches the candidate? –or-
  - Is there any Pending Intercept Payment (INTP) record where the Vendor matches the candidate? –or-
  - Is there any Externally Intercepted Debt (EXINCTDB) record where the Vendor/Customer matches the candidate?
- If the site is not a Dual HRM/Financial site, is there any Trip Detail (TRIPD) record where the Traveler ID matches the candidate and the Due to Vendor or Due to Traveler is greater than zero?
- Is there any Customer Account (CUSTA) record where the Customer matches the candidate and there is a nonzero value in Earned Unliquidated Rec, Unearned Unliquidated Rec, Refund Unliquidated Rec, Deposit Unliquidated Rec, Unreserved Credit Balance, or Reserved Credit Balance?
- Is there any Receivable History and Reference Query (RDET) record where the Customer matches the candidate and the Total Outstanding is nonzero?
- Is there any FHWA Rejection Records (FHWACLN) whose Customer Identifier matches the candidate?
- Is there any Electronic Billing Inquiry (EBIT) where the Vendor Code matches the candidate and the Payment Request Generated Date is null?
- Does the candidate have any FA Shell yet to complete, i.e., can we find a Fixed Asset Payment Request (FAPR) and child Generated Shell (FASHELL) record satisfying all of the following?

  - FAPR Vendor/Customer matches the candidate. –and-
  - FAPR Purge Shell Record = No –and-
  - FAPR Shell Generation Date is null, or FAPR has a child FASHELL with Shell Acceptance Date of null.

- Is there any Issue Queue Inquiry (ISSQ) where the Vendor Customer matches the candidate?

- If the Check Tax Reporting parameter value is equal to Y, is there any 1099 Reported Income (1099R) record where the Taxpayer ID and TIN Type combination matches the VCUST record's TIN and TIN Type combination, and the Calendar Year is on or after (Current Calendar Year – Tax Info Retention Years)?

- If the Check Tax Reporting parameter value is equal to Y, is there any 1099 External Reported Income (1099ER) record where the Taxpayer ID and TIN Type combination matches the VCUST record's TIN and TIN Type combination, and the Calendar Year is on or after (Current Calendar Year – Tax Info Retention Years)?

- If the Check Tax Reporting parameter value is equal to Y, is there any 1042-S Reported Income (1042R) record where the Recipient's Account Number matches the VCUST record's 1042-S Recipient Account Number, and the Calendar Year is on or after (Current Calendar Year – Tax Info Retention Years)?

- Does the candidate have any open accounting activities that should prevent archiving?

  - Is there any accounting line satisfying all of the following?

    - The transaction type is not JV and event type has Open Activity for Vendor Archiving equal to Yes on the Event Type (ETYP) table. Do not check any TRVL transaction type unless the DUAL_HRM_FIN_SITE parameter is set to false on the APPCTRL table. –and-

    - The vendor customer code matches the candidate. –and-

    - The transaction is a New or Modification transaction in Final Phase where Line Amount is not equal to the Closed Amount, or the transaction is in Pending Phase.

  - If a site has custom event types that have an open activity concept that deals with vendors or customers and does not update the Disbursement Request table, the Open Activity for Vendor Archiving flag must be set to Yes in order for the open accounting line activities selection criteria to consider the custom event types.

- Is there any Active Final Master Agreement (MA) that is in effect whose primary vendor matches the candidate?

- Does the candidate have any debt or loan in progress?

  - Is there any Debt Registry (BOND, LOAN, or LEASE) record satisfying all of the following?

    - The Debt Instrument Status is Application Started, Loan Started, Loan in Review, Lease Started, Lease in Review, Bond Started, Bond in Review, Application Ready for Loan, Active Loan, Active Lease, or Active Bond. –and-

    - The Vendor Code under the Payment Agent/Lender Information matches the candidate, or the Customer Code under the Borrower Information matches the candidate.

- Is there any Escrow Definition (ESCROW) with Completed = No where the Escrow Agent matches the candidate?

- Is there any Collection Agency Agreement (COLLA) record where the Vendor matches the candidate and the Outstanding Amount is nonzero?

- Is there any Coupon Return (CRT) record with Marked for Delete = No where the Customer matches the candidate?
- Is there any Protest (PRTST) record where the Vendor/Customer matches the candidate and the hidden Final indicator corresponding to the Protest Status has a value of No.
- Is there any Cash Sweep (SWEEP) record with Sweep Status = Active with an External Participant Vendor Code matches the candidate?
- If the candidate is a Headquarters Account, are all the subsidiary vendor customer records eligible for archiving?

Once all the Vendor/Customer codes eligible to be archived have been determined, the job checks each of the selected Vendor/Customer codes to see if any associated 1042-S Recipient Account Numbers can be archived.

- Examine each selected Vendor/Customer code. If it has 1042-S Recipient Account Number populated on the VCUST table, and there are no other VCUST record using this 1042-S Recipient Account Number or if all other VCUST records using this 1042-S Recipient Account Number are also selected for archiving, then the 1042-S Recipient Account Number is eligible for archiving.

Once all the 1042-S Recipient Account Numbers eligible to be archived have been determined, the job checks each of the selected Vendor/Customer codes to see if any associated TIN and TIN Type combinations can be archived.

- Examine each selected Vendor/Customer code. If it has TIN and TIN Type populated on the VCUST table, there are no other VCUST records using this TIN and TIN Type or if all other VCUST records using this TIN and TIN Type are also selected for archiving, and there are no other 1042I record using this TIN and TIN Type or if all 1042I records using this TIN and TIN Type are also selected for archiving, then the TIN and TIN Type combination is eligible for archiving.

Finally, the job examines the Vendor User Information (R_VEND_USER) and the Vendor Customer Archive table entries to determine if there are any Vendor/Customer codes associated with the User ID that are not archived. If there are none, then the User ID associated with the R_VEND_USER is eligible for archiving.

Running the Vendor Customer Archiving chain in *Report Only* mode is a useful tool to find out which records are being identified by the eligibility criteria and how many records in the expanded selection will be archived and deleted when the chain is run in *Full Run* mode.

This chain implements the Data Warehouse Archived Record Queue process, which is used to retain the table names and key values for records that are archived. The infoAdvantage reporting system uses this information to distinguish between records that have been archived (as valid historical data) before deletion and records that have simply been deleted from the system (as no longer needed). The process is enabled for most archiving processes when the value of the Application Parameter Enable Data Warehouse Archived Record Queue (ENABLE_DW_ARCH_QUEUE) is *True*. Please see the *CGI Advantage System Administration Guide* for more information regarding this process and the Application Parameter.

Details

One group of tables is archived to either CSV or XML files using export functionality and then purged. Another group of tables is just purged. The second group consists of tables that are considered peripheral tables to the other tables being archived. Not all tables that could be considered as having Vendor/Customer data are covered by this archive.

- Transactions still require Transaction Archiving. A transaction archive for Vendor Creation and Vendor Modification transactions may be paired with a Vendor Customer Archive. The order of the two is not dependent on any cross-editing.
- Journals & Ledgers have a separate archive and can be archived before or after a vendor customer archive.
- The Procurement subsystem has a Procurement Folder Historical Update process for marking old data as historical, so vendor procurement data follows that process.
- If the Check Writer functionality is used, the CW Archive Chain should be run before the Vendor Customer Archive is run to archive records on the Check Writer tables. This reduces the number of CW records that the Vendor Customer Archive needs to verify in order to determine if a vendor can be archived.
- If the Payment Hold functionality is used, the Vendor Customer Archive should be run to archive vendors off VCUST before running the Payment Hold Archive Process, which identifies and archives all the records on the Payment Hold Maintenance table where the Hold Level is “Vendor,” having no corresponding Vendor records on VCUST.
- Vendor Customer data in other areas such as Debt Management, Travel, Intercepts, Retainage, Procurement Card, or Grant Lifecycle Management is not handled by the Vendor Customer Archive process because data in these areas is logically organized under a different set of attributes other than vendor customer code. Only when archive processes are built for these areas will records be archived in these areas.
- A list of other tables containing data that has to be addressed manually can be found at the end of the chain description in a Manual Intervention section.

Other tables may retain vendor customer inquiry data even though the corresponding vendor customer setup data has been archived. Those tables where orphaned vendor/customer data is not directly used in system processing have had their referential integrity edits relaxed to allow other information in those records to be updated without getting any missing vendor/customer data error. (A list of these is found at the end of the chain description in a Relaxed Referential Enforcement section.)

As with all jobs, parameters are validated first. Once that validation is successful, the first step in selection is to determine the eligible Vendor/Customer codes, 1042-S Recipient Account Numbers, Taxpayer ID and TIN Type combinations, and User IDs as discussed in a previous section. If the chain is run in *Report* Only mode, the results of the selection are saved to the Vendor Customer Archive (VCARCH) table so it is available for review.

The second step is to form additional search criteria to find matching records on related tables using the list of tables for export and deletion from the instruction file.

Once the final set of records is selected, facilitator records are submitted to the Job Manager, where the records from certain tables (mainly from the Accounts Payable and Accounts Receivable areas) are exported (archived) using a parallel processing. Once the export process has successfully completed, the system will purge records for the archived and the non-archived tables.

It is important to schedule this job in proper placement with relation to jobs typically found in a nightly cycle. The recommended relative positioning of the Advantage Financial to VSS Synchronization Processes, the Vendor Customer Archiving chain, and the Vendor Customer Archiving VSS Updates job within a typical nightly cycle is as follows:

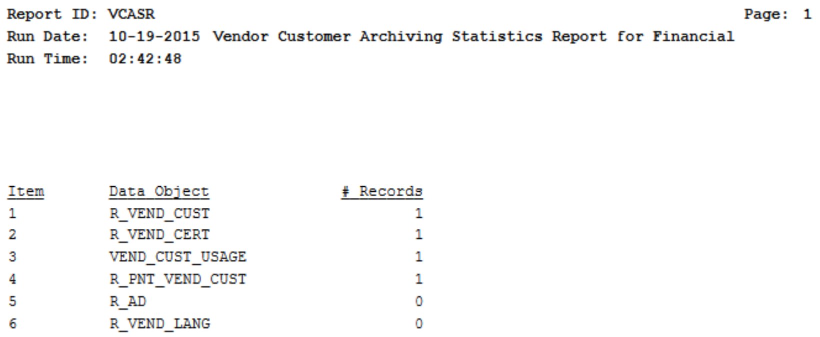
1. Export from VSS
2. Load to Advantage
3. Create Certification Transactions
4. DPC Table Export
5. VCM Tracking Table Load
6. Tracking Tables to VCM
7. Vendor Customer Archiving Chain
8. Export from Advantage
9. Load to VSS
10. Catalog Export from Advantage
11. Load Catalog to VSS
12. VCM Tracking Table Export
13. DPC Table Load
14. Vendor Customer Archiving VSS Updates
15. Catalog Indexing
16. Email Letter Generation (in Financial)
17. Email Letter Generation (in VSS)
18. DPC Purge
19. SA13 jobs

Furthermore, the following dependencies should be taken into account when determining when to schedule the Vendor Customer Archiving process:

- If the Open Activity Roll or Open Activity Lapse functionality is used, the Vendor Customer Archive should be run after year-end activities have completed. This ensures vendors or customers are still available in the system in order to complete year-end activities successfully.
- If there are any System Assurance 13 out-of-sync issues involving vendors who are eligible to be archived, it is recommended to resolve these out-of-sync issues prior to running the Vendor Customer Archiving process to avoid further complications.
- Calendar Date (CLDT), Fiscal Year (FY), and Accounting Period (APD) records should not be archived until all processing (including archival processing) has completed. In particular, Vendor Customer Archiving will run into errors if the system cannot determine the date, FY, or accounting period information while performing parameter edits and record selection.
- If a long time has passed after the Vendor Customer Archiving chain was run in Report Only mode, you may need to run the chain in *Report Only* mode again to get an updated listing of the codes that are eligibility for archiving.

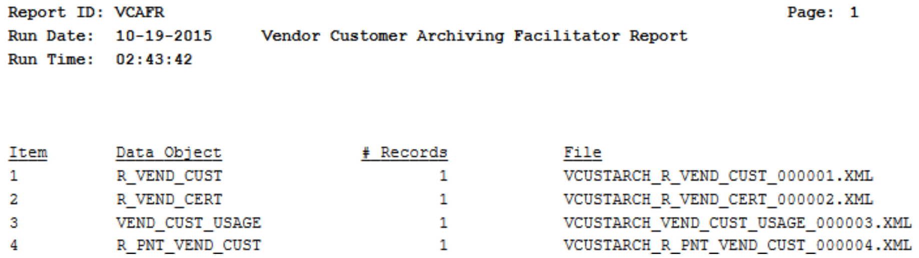
The output from the process includes two reports:

1. Vendor Customer Archiving Statistics Report for Financial showing a count of selected records per table. The sample below is abbreviated as there are many more items.



```
Report ID: VCASR                                                        Page: 1
Run Date:  10-19-2015  Vendor Customer Archiving Statistics Report for Financial
Run Time:  02:42:48



Item        Data Object              # Records
1           R_VEND_CUST                      1
2           R_VEND_CERT                      1
3           VEND_CUST_USAGE                  1
4           R_PNT_VEND_CUST                  1
5           R_AD                             0
6           R_VEND_LANG                      0
```

2. Vendor Customer Archive Facilitator Report listing each archive file created and the count of records per file. When run in *Report Only* mode, this report is not created as the chain ends after the Archiving Preprocessor completes the Statistics report.



```
Report ID: VCAFR                                                          Page: 1
Run Date:  10-19-2015     Vendor Customer Archiving Facilitator Report
Run Time:  02:43:42



Item        Data Object              # Records           File
1           R_VEND_CUST                      1           VCUSTARCH_R_VEND_CUST_000001.XML
2           R_VEND_CERT                      1           VCUSTARCH_R_VEND_CERT_000002.XML
3           VEND_CUST_USAGE                  1           VCUSTARCH_VEND_CUST_USAGE_000003.XML
4           R_PNT_VEND_CUST                  1           VCUSTARCH_R_PNT_VEND_CUST_000004.XML
```

Chain Job Return Code

The acceptable job return codes (configured in the Configure Chain Job section of the Job Setup in CGI Advantage) for the jobs in the Vendor Customer Archiving chain are delivered to be set to *Successful*. As with all CGI Advantage chain jobs, these acceptable return codes are configurable and may be changed to meet certain requirements.

The following table shows the potential return codes for the Vendor Customer Archiving chain. Note that the chain job will end with the highest return code across all of the jobs.

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

Problem Resolution

Please refer to the individual job “Problem Resolution” section for more details.

Advanced Selection Guidelines

The parameter file listing all the tables which are part of the archive can be modified within limits. Each table is listed with an export or a delete parameter line which can be changed, allowing you to retain some table data delivered as delete only or to simply delete data delivered to archive. Custom tables can be added in the same manner as those delivered in the file. The archive will not work with any table added that contains JRNL, LDGR, BUD_STRU, or DOC in the table name. The addition of other tables to the file is possible, but something that should receive close attention and testing before attempting in a production environment.

Manual Intervention

The following is a list of tables that need to be examined for vendor customer usage. They are not included in the Vendor Customer Archiving selection criteria when examining if any outstanding activities exist. If the vendor customer is eligible for archiving but it is used on one of these tables, you need to determine if the vendor customer should be prevented from archiving by setting the Never Archive flag to Yes on the VCUST record, or if the setup on these tables needs to be updated as described below under the Comments column. The following list contains only the baseline tables and you may have other custom ones as well.

Table	Field	Comments
Advantage HRM Application		
Deduction Plan (DPLN)	Vendor (VEND_CUST_CD)	If payment vouchers will be generated for deductions, choose another vendor if applicable.
Advantage Financial Application		
Payroll type transactions	Vendor Line: Vendor Customer (VEND_CUST_CD)	Check for any draft or pending payroll type transactions. Process them to final or reject them.
ITI, ITA transactions	Header: Vendor Customer (VEND_CUST_CD) Vendor Line: Vendor Customer (VEND_CUST_CD)	Check for any open ITI and ITA transactions if you are archiving an internal vendor or customer.
Billing Profile (BPRO – R_BPRO)	Collection Agreement (CLAGCY_AGRMT_NO)	Each collection agreement is tied to a vendor to whom past due receivables are sent for collection. If the vendor will be archived, the corresponding Collection Agency Agreement (COLLA) record would also be archived, and any BPRO using the COLLA would need to be updated to use another collection agreement.
Vendor/Customer (VCUST –	Third Party Code	If a vendor eligible to be archived is set up as a third party vendor

R_VEND_CUST)	(THRD_PTY_CD) Third Party Address ID (THIRD_PTY_AD_ID)	for another vendor who is ineligible for archiving, you can update the latter by removing the third party vendor setup or choosing another third party vendor setup. If both vendors are eligible for archiving, no updates are required. This scenario would not prevent the third party vendor from archiving because the usage check tells you that the third party vendor has not been used for a long time. If the third party relationship has been used, the first vendor would not have been selected for archiving in the first place.
Customer Account Option (CACT – R_CUST_ACCT_OPT)	Third Party code and Address ID (THRD_PTY_CD_1, THRD_PTY_AD_ID_1, THRD_PTY_CD_2, THRD_PTY_AD_ID_2, … THRD_PTY_CD_10, THRD_PTY_AD_ID_10)	If a customer eligible to be archived is set up as third party for another customer who is ineligible for archiving, you can update the Billing Location setup for the latter or choose another third party customer setup. If both customer and all third parties are eligible for archiving, no updates are required. This scenario would not prevent any third party customer from archiving because the usage check tells you that the third party customer has not been used for a long time. If the third party customer is set up solely for the purpose of billing, then the Never Archive flag should be set because billing activities do not update the Last Usage Date.
1099 Reporting Payer Information (1099RP - R_1099_RPT_PYRINFO)	Payer TIN and TIN Type (PYR_TIN and TIN_TYP)	This table sets up reporter payer information for IRS reporting.
1099 Transmitter Information (1099TI - R_TRNSMTR_INFO)	Transmitter TIN and TIN Type (TIN and TIN_TYP)	This table sets up transmitter information for IRS reporting.
Payment Hold by TIN (PHLDTN – R_PYMT_HLD_TIN)	Vendor (VEND_CUST_CD) Address ID (AD_ID)	This table identifies taxpayers with owed or settled judgment debts.

	Taxpayer ID and Type (TIN and TIN_TYP)	
Payment Hold Maintenance (PHLDM - R_PYMT_HLD_MNTN)	Vendor (VEND_CUST_CD) Address ID (AD_ID) TIN and TIN Type (TIN and TIN_TYP)	This table stores information regarding hold requests, removal requests, and allow bypass requests. If the Hold Level is Vendor and the VCUST record does not exist, you can run the Payment Hold Archive Process to archive the PHLDM records.
Disbursement Hold Exclusion (DISBMR – R_DISB_MGMT_RSTR)	Vendor (VEND_CUST_CD) Tax Payer ID and Type (TIN and TIN_TYP)	This table identifies exclusions to undisbursed payment requests that could be held during disbursement. If Vendor is the only exclusion criterion, the record can be deleted; otherwise update the record to reflect the applicable criteria.
Intercept External Allocation (INTEA – R_INT_EA)	Payment Vendor (VEND_CUST_CD) Address ID (AD_ID)	Choose another payment vendor setup.
Intercept Request (INTR - R_AP_INCT_RQST)	TIN and TIN Type (TIN and TIN_TYP)	
Intercept Disbursement Exception (INTDBEX - R_AP_DISB_EXCP)	TIN and TIN Type (TIN and TIN_TYP)	This table identifies what is exempt from having an intercept claim held against it.
Receivable Intercept Exception (INTREX – R_RCVB_INT_EXCP)	Customer (VEND_CUST_CD) TIN and TIN Type (TIN and TIN_TYP)	This table identifies receivables that are deemed eligible for intercept to be bypassed by the intercept portion of the Disbursements process. Update the record to reflect the applicable criteria.
Federal Offset Activity (FOTA – R_FED_OFST_ACTV)	TIN and TIN Type (TIN and TIN_TYP)	When the Federal agency refunds previously intercepted debt, a record is put here pending reversal. As you would need to manually modify the transaction that recorded the initial offset, and there is no definite timeframe for when reversal might happen, this table will be manually managed.

Travel Administration (TRADM – R_TRVL_ADM)	Miscellaneous Vendor Customer (TRVL_MISC_VEND_CD) Billing Address Code (AD_ID)	When both HRM and Financial are in use, the Miscellaneous Vendor Customer and Billing Address Code on the TRADM table are used to default the vendor information.
Electronic Federal Tax Payment System (EFTPS - ELEC_FED_TAXP_SYS)	Taxpayer ID Number (TIN)	
Inventory Maintenance (INVN – R_INVN)	Suggested Vendor (VEND_CUST_CD)	Blank out the Suggested Vendor or choose another vendor as the suggested vendor.
Inventory Maintenance Detail (INVND – R_INVN_DET)	Warranty Vendor (WTY_VEND_CUST_CD) Warranty Address ID (WTY_AD_ID)	Blank out the Warranty Vendor and Address ID or choose another combination.
Inventory Replenishment (IREP)	Suggested Vendor (SUGG_VEND_CUST_CD)	Blank out the Suggested Vendor or choose another vendor as the suggested vendor.
Major Program (MJPRG – R_MJR_PROG)	Internal Vendor (CUST_ID)	Choose another vendor to be internally reimbursed on the internal transactions generated by the Reimbursement Output process.
Funding Line (FPRFLST – R_FLINE)	Customer ID (CUST_ID) Unit (UNIT_CD) Billing Profile (BPRO_CD)	Used to populate the generated Reimb Output transactions. Since you cannot change the Customer ID, if another Customer ID is desired, you may need to delete and then insert a funding line to replace the existing one.
CMIA Interest Parameters (CMIAINTR – R_CMIA_INTR_PARM)	Customer (CUST_ID)	This table stores the parameters for the CMIA Interest Calculation Process. Update the Customer selection criterion as needed.
Reimbursement Generation Parameters (REIMGEN – R_REIM_GEN_P_CUST)	Customer ID (CUST_ID)	This table stores the parameters for the Reimbursement Generation Process. Update the Customer selection criterion if the run has not occurred yet.
Reimbursement OutPut Parameters (REIMOTPT – R_OTPT_PARM_DET)	Customer (CUST_ID)	This table stores the parameters for the Reimbursement Output Process. Update the Customer

		selection criterion as needed.
Internal Cost Rate (INCR – R_INT_CST_RT)	IVAD Vendor Code (IVAD_VEND_CUST_CD)	Choose another Internal Vendor Accounting Data (IVAD) table record that will be used to create a revenue line for indirect costs.
Grant Opportunity (GTOPSF – R_GRNT_OPTY)	Funding Agency (FNDG_AGCY_CD) Agency (PASS_THRU_AGCY_CD)	
Grant Award (GTAWSF – R_GRNT_AWD)	Funding Agency (FNDG_AGCY_CD)	

Relaxed Referential Enforcement

The following list of tables may retain data related to Vendor/Customer records that have been archived. This presence is allowed by relaxing the referential enforcement between the listed table and Vendor/Customer (R_VEND_CUST), Address (R_AD), or Account Users (R_VSS_USER). Custom objects may require similar treatment if they require data to be maintained in the system that references archived Vendor/Customer data. The following list contains only the baseline tables and you may have other custom ones also.

- Collateral Management (COLL_MGMT)
- Matching Status (MATCH_STA_AWD)
- Bid Deposit (R_BID_DPS)
- Procurement Bond (R_BOND)
- List of Insured (R_CERT_INSRD)
- Claim Tracking (R_CLMTRK)
- Collateral Management by Instrument (R_CL_MGMT_INST_HDR)
- Contract Assignment (R_CNTRC_ASGN)
- Contract Goal Header (R_CNTRC_GOAL_HDR)
- Contract Goal Line (R_CNTRC_GOAL_LN)
- Vendor Correspondence (R_CORSPD)
- Borrower Information (R_DEBT_BRWR)
- Lender Information (R_DEBT_LNDR)
- Historical Vendor Information (R_HIST_VEND_INFO)
- Insurance Certificates (R_INSU_CERT)
- Inventory (R_INVN)
- Inventory Maintenance Detail (R_INVN_DET)
- Inventory Replenishment (R_IREP)
- Letters of Credit (R_LTRCR)
- Mandatory Source Master Agreement Lines (R_MSMALS)
- Protest (R_PRTST)
- Recipient Vendor Quarterly Information (R_RCPNT_VEND_QTRLY)
- Retainage Summary by Commodity Line (R_RTG_SUM)
- Solicitation Question & Answer (R_SO_QA)
- Sub-Recipient Quarterly Award Information (R_SRCPNT_QTRLY_AWD)
- Sub-Recipient Vendor Quarterly Information (R_SRCPNT_VENDQTRLY)
- Vendor Solicitation List (R_VENDSLST)

Post Archive

Most find that moving the files created from the archive out of the Advantage file system gain the most in overall system file size. However, this is not a requirement but common place as there is not an automatic restore job for this data that would look to the initial location for the files.

Vendor Customer Archiving Chain: Archiving Preprocessor Job

Job Name	Archiving Preprocessor
Recommended Frequency	On Demand This job must be run as part of the Vendor Customer Archiving Chain.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Vendor Customer Archiving Statistics Report for Financial

Overview

This step first validates parameters. If parameters are valid, it then identifies the rows to be archived based on the eligibility criteria. Once the Vendor/Customer codes, 1042-S Recipient Account Numbers, Taxpayer ID and TIN Type combinations eligible to be archived have been determined, records are written to the Vendor Customer Archive (VCARCH) table.

The Instructional File parameter identifies a text file that is required to be present in the parameter directory (PARMS). For each table listed, one key selection value must be specified. The five possible selection keys are Vendor/Customer Key Selection (VCUST_KEY_SELECTION), Headquarters Key Selection (HQ_KEY_SELECTION), 1042I Key Selection (1042I_KEY_SELECTION), 1099I Key Selection (1099I_KEY_SELECTION), or VSS User Key Selection (USER_KEY_SELECTION). It may also optionally contain DEP_CHILD values as well for dependent child table record selection. Depending on the type of key used, it will use that information to determine which column to select from the indicated table.

Once the job has determined matching records on related tables using the selection key criteria, it divides the selected rows in each table to be archived into groups for each export file. For example, if R_VEND_CUST has 200 rows and the Job Block Size parameter is 100, then two export files are created.

The Vendor Customer Archiving Statistics Report for Financial lists the total number of records to be archived or deleted from each table. The VCARCH page shows the specific Vendor/Customer codes, 1042-S Recipient Account Numbers, and Taxpayer ID and TIN Type combinations that will be archived based on the user-specified parameters with other pre-defined selection criteria. If the run mode is set to *Report Only*, the process will end at this point.

In *Full* run mode, a System Maintenance Utility parameter file is created for each export file to be created. The parameter file contains the table name, the identification of specific records, and other settings. A Facilitator table record referencing the SMU parameter file is created and saved. Facilitator records can be viewed in the Archive Facilitator Inquiry page (QARC) for the status. The file names for the archived records are written to the chain parameter file for use by the Post Archiving Process job of the chain.

Process Steps	Messages
1. Parameter Validation	• Run Started Each parameter is listed • Validating Batch Parameters If any is found to be invalid an error will also be issued • Batch Parameters are valid. *or* • Batch Parameters are not valid.
2. Pre-processing	• Purged <number of records> Report Only records from the VCARCH table. • Non-HQ Record selection started. • Non-HQ Record selection completed. • Processing Non-HQ VCUST records and generating VCUST_KEY_SELECTION keys. • HQ Record selection started. • HQ Record selection completed. • Processing HQ VCUST records and generating VCUST_KEY_SELECTION keys. • Updated <number of updated records> VCARCH records for 1042-S Recipient Account Number archive eligibility. • Updated <number of updated records> VCARCH records for TIN and TIN Type archive eligibility. • Generating HQ_KEY_SELECTION keys. • Generating 1042I_KEY_SELECTION keys. • Generating 1099I_KEY_SELECTION keys. • Generating USER_KEY_SELECTION keys. • Processing Expanded Record Selection • Processing table: <table name> • Expanded Record Selection completed. • Rendering Statistics report started. • Reports output folder mapped (*followed by the HTML & PDF locations)* • Rendering Statistics report completed. • Number of records deleted from temporary table: <number of deleted records> • Record selection completed. • Rendering Statistics report started. • Reports output folder mapped (*followed by the HTML & PDF locations)*. • Rendering Statistics report completed. • Run Ended

	Full Run Only will contain messages: • Records from <table name> identified for Facilitator table: <number of records> • Writing chain parameter file. • Chain parameter file (Chain Parameter File Name) written in directory (CGI Parameter Directory).

Major Input

- Vendor/Customer (R_VEND_CUST)
- Vendor/Customer Usage (VEND_CUST_USAGE)
- Instruction File Table List (Tables are based on contents in the file)

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

Batch Parameters

Parameter	Description	Default Value
AMSEXPORT	Export Location at Archiving Preprocessor Job. Optional. If the default value is changed, it must be a valid directory and will be used to determine where the export files will be written. (The size of archived files may be so great that a location other than the default Export/Import directory should be used.)	$$AMSROOT$$/ExportImport
AMSPARM	Parameter Location at Archiving Preprocessor Job. Required. This must be a valid directory containing the file designated by the Parameter File parameter.	$$AMSROOT$$/Parms
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) Required, not overrideable. This file is created during the run to convey parameters to the subsequent steps in the chain.	VCUSTArchParams.txt
CLIENT_NM	Client Name for reports. Optional.	No Default
COMMIT_BLOCK_SIZE	Commit Block Size. Required. This value will determine how often database transactions will be committed and can be used for performance tuning. The value	1000

	must be a positive integer.	
EXP_FILE_TYP	Export File Type. Required. Enter '1' for XML, '2' for CSV.	2
FILE_PREFIX	File Prefix. Required. Prefix used for archive file names. Job will use VCUSTArch if left blank.	VCUSTArch
JOB_BLOCK_SIZE	Job Block Size. Required. The maximum number of lines in an output file. If left blank the process will assume 10000.	10000
RUN_MODE	Run Mode. Required. Enter '1' for Report Only, '2' for Full. If left blank then 1 will default.	1
SMU_FILE_PREFIX	SMU Facilitator Parameter File Prefix. Required, not editable. This value will be used to determine a SysManUtil input parameter file name for each Facilitator job.	VCUSTArch
TBL_PARM_FILE	Instructional file with table listing (txt). Required, not editable. This file will be used to identify tables for selection expansion.	VCUSTArchListParams.txt
LAST_USAGE_DT	Last Usage Date cutoff. Eligible for archiving if vendor customer was last used before this date. Enter in mm/dd/yyyy format. Required, editable	No Default
CHK_TAX_RPT	Check tax reporting (Y/N) Required, editable	No Default
EXCL_INT_ACCT	Exclude Internal Account (Y - Exclude, N - Do not exclude) Required, editable	N
EXCL_MISC_ACCT	Exclude Miscellaneous Account (Y - Exclude, N - Do not exclude) Required, editable	Y
EXP_ATT_FL	Export with Attachments (true/false) Conditionally required. Required when running full mode.	True
EXP_ATT_TYP	Attachments Export Type Required if EXP_ATT_FL entered. Enter '1' for XML, '2' for CSV.	

Major Output

The following are output in both the *Report Only* and *Full* modes:

- Vendor Customer Archiving Statistics Report for Financial
- Vendor Customer Archiving Temporary (VEND_ARCH_TBL) table which contains a record for each table and a record for each selected record in that table
- Vendor Customer Archiving (R_VEND_CUST_ARCH) table which shows the Vendor/Customer codes, 1042-S Recipient Account Numbers, and Taxpayer ID and TIN Type combinations that are eligible to be archived based on the user-specified parameters with other pre-defined selection criteria.

The following are only output in *Full* mode:

- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page.
- VCUSTArch##_#.txt – File created to be passed to SMU to perform the archive. The ## is the chain job ID and the # is an incrementing number for each table archived.
- ARCH_DW_QUEUE_REC – The Data Warehouse Archive Record table keeps track of the values for all primary keys of each record being archived. The table name and attribute names are stored in ARCH_DW_QUEUE_TBL and linked to this record by unique ID.
- ARCH_DW_QUEUE_TBL – The Data Warehouse Archive table keeps track of the table name and primary key attribute names of each table being archived. If a record already exists for the table being archived, it will reuse the record. If it does not yet exist, it will be created.

Minor Output

These are only output in *Full* mode:

- VCUSTArchParams.txt – To pass information from the first job in the chain to later jobs.
- Folder in ExportImport / VendCustArchive

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	This return code is issued under the following condition: • The Run Mode parameter is set to '1' (Report Only). When this job ends with a return of code Warning, subsequent jobs in the chain are set to Inactive.
Non Fatal Error (8)	This return code is issued under the following condition: • No eligible records found for the selection file. When this job ends with a return of code Non Fatal, subsequent jobs in the chain are set to Inactive.
Failed (12)	The job fails under the following conditions: • One or more parameter(s) are invalid. • One or more required parameter(s) are not entered. • The Instructional File is not found in the directory specified by the

	Parameter Location at Archiving Preprocessor Job parameter. • Run time exceptions for unexpected situations. When this job ends with a return of code of Failed, subsequent jobs in the chain are set to Inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain are set to Inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return of code System Failure, subsequent jobs in the chain are set to Inactive.

Sort Criteria

Vendor/Customer Code

Selection Criteria

The Instructional file parameter identifies a text file that is required to be present in the parameter directory. Within the Instructional file, one key selection must be specified for each table where records will be archived or deleted. Each table has its own set of selection criteria that must start with the _PARAM_LINE_ separator. Selection key criteria can be one of five values: VCUST_KEY_SELECTION, HQ_KEY_SELECTION, 1042I_KEY_SELECTION, 1099I_KEY_SELECTION, or USER_KEY_SELECTION. It may also optionally contain DEP_CHILD values as well for dependent child table record selection. Selection Key criteria values in the input file may use commas to specify more than one column.

Input File Example:

```
_PARAM_LINE_
TABLE_NAME=R_TIN_1099_INFO
1099I_KEY_SELECTION=TIN,TIN_TYP
**ACTN_CD=TBLEXPORT
_PARAM_LINE_
TABLE_NAME=R_PYMT_SCHED
VCUST_KEY_SELECTION=VEND_CUST_CD
DEP_CHILD=R_PYMT_SCHED_DET;DOC_CD,DOC_CD;DOC_DEPT_CD,DOC_DEPT
_CD;DOC_ID,DOC_ID;UNID,UNID
**ACTN_CD=TBLEXPORT
_PARAM_LINE_
TABLE_NAME=R_CR_BAL_DET
VCUST_KEY_SELECTION=VEND_CUST_CD
**ACTN_CD=TBLDELETE
```

Within each _PARAM_LINE_ group, that particular selection key indicates which table and fields to use for selecting eligible records from that table. For example, if a 1099I_KEY_SELECTION value is provided, that means the selection process will use the TIN and TIN_TYP from the

VCARCH record to map to the columns provided in this parameter to determine the eligible records on the TABLE_NAME table defined. It will then execute the action indicated, which, in the example above, is TBLEXPORT. The job determines matching records on related tables based on the selection key criteria as follows:

- For VCUST_KEY_SELECTION, the field specified needs to match any vendor customers from the VCARCH records written by this chain.
- For HQ_KEY_SELECTION, the field specified needs to match any vendor customers from the VCARCH records written by this chain where the Headquarters Account flag is equal to Yes.
- For 1042I_KEY_SELECTION, the field specified needs to match any 1042-S recipient account numbers from the VCARCH records written by this chain where the Archive 1042-S Recipient Account Number flag is equal to Yes.
- For 1099I_KEY_SELECTION, the field specified needs to match any TIN and TIN Type combinations from the VCARCH records written by this chain where the Archive TIN and TIN Type flag is equal to Yes.
- For USER_KEY_SELECTION, the job examines the Vendor User Information (R_VEND_USER) and the VCARCH table entries to determine if there is any Vendor/Customer codes associated with the User ID that are not archived. If there are none, then the User ID associated with the R_VEND_USER can also be archived, and that User ID is used to find records by matching the field specified by the USER_KEY_SELECTION.
- If there is a DEP_CHILD line specified in the table parameter file under a given table, then the system also identifies related child records for each parent record by matching each of the parent-child attributes specified in the DEP_CHILD line of the table parameter file.

Input File Rules:

1. The column values provided in the key selection must exist on the table specified.
2. One key selection value must be provided for any table indicated.
3. The table name must be provided for any key selections indicated.
4. The **ACTN_CD must be TBLEXPORT or TBLDELETE and must be present.
5. 1099I_KEY_SELECTION must have the selection keys mapping to TIN and TIN_TYP in that order. For example, if a table had two differently named columns VEND_TIN and VEND_TIN_TYP, the selection key must have VEND_TIN,VEND_TIN_TYP in that order.
6. The DEP_CHILD functionality can be used to delete records from the child table in situations where the parent table record will be deleted based on one of the KEY_SELECTION, but the child table does not have any fields that can be identified using one of the KEY_SELECTION. DEP_CHILD should be in the format of <dependent child table name>;<parent column name x>,<child column name mapping to x>;<parent column name y>,<child column name mapping to y>.

For example:
DEP_CHILD=R_PYMT_SCHED_DET;DOC_CD,DOC_CD;DOC_DEPT_CD,DOC_DEPT_CD;DOC_ID,DOC_ID;UNID,UNID

Problem Resolution

If the job ends with a return code other than Successful or Warning after completing parameter validation, a new job should be scheduled after the job log has been reviewed (the failed job should not be restarted) and the problems have been addressed. If the job was run in *Full Run* mode, it is possible that the job has already identified some records for archival and written them to the VCARCH table using Run Mode equal to Full Run and Financial Completed equal to No. If this happens, the Archiving Preprocessor job will issue an error if you try to run it again, because there is an edit to ensure that a Full Run needs to run to completion before another run can proceed. There are two options to clean up the VCARCH table in order to allow the chain to be rescheduled. Note that these two options are applicable if the chain is interrupted during the Archiving Preprocessor job. If the chain is interrupted during the Archive Facilitator or Post Archiving Process job, refer to the “Problem Resolution” section under the corresponding job for more details.

Option 1: You can use the System Maintenance Utility (SMU) job under Utilities > Batch Jobs to remove the partially-written list of eligible records from the VCARCH table. To do this, you need to create a parameter file, place it on the server, and run the SMU job specifying the Parameter File Name. If this option is used, you also need to dispose of the file generated by SMU properly, once the cleanup is completed.

Here is a sample parameter file. The AGNT_ID value needs to be replaced with the Job ID of the chain.

```
**ACTN_CD=TBLEXPORT
_PARAM_LINE_
TBL_NM=R_VEND_CUST_ARCH
FM_KEY=AGNT_ID=123456;RUN_MODE_IND=2;FIN_CMPLT_FL=0
TO_KEY=AGNT_ID=123456;RUN_MODE_IND=2;FIN_CMPLT_FL=0
FILE_NM=$$AMSROOT$$/ExportImport/VCARCH_DEL.xml
**ACTN_CD=TBLDELETE
_PARAM_LINE_
FILE_NM=$$AMSROOT$$/ExportImport/VCARCH_DEL.xml
EDITS_FL=false
```

Option 2: If a database backup was taken before the chain was run, you can restore the R_VEND_CUST_ARCH table using the database backup.

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)	All of the parameters are validated successfully.	N/A	N/A

Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. **Sample Message:** Parameter Run Mode value must be ‘1’ or ‘2’.	Schedule a new job after entering a valid value for the parameter.	N/A
	Entered parameters are not valid. **Sample Message:** Parameter Job Block Size value 'ABC' is invalid. It must be a positive integer.	Schedule a new job after entering a valid value for the parameter.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Pre-processing

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	The Run Mode parameter is set to '1' (*Report Only*). This is a normal condition that sets any subsequent jobs to inactive.	N/A	N/A
Non Fatal Error (8)	No tables need to be archived because no records match eligibility criteria.	Confirm eligibility criteria before scheduling a new job.	N/A
Failed (12)	Failed during attempt to write the chain parameter file.	Verify that the file system is not full, has the correct access, and so forth, before scheduling a new job.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A

Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Vendor Customer Archiving Chain: Archive Facilitator Job

Job Name	Archive Facilitator
Recommended Frequency	On Demand This job must be run as part of the Vendor Customer Archiving Chain.
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	None

Overview

The Facilitator job spawns multiple System Maintenance Utility (SMU) jobs to export (archive) selected Vendor Customer related records to files. This step does **not** delete records from the Vendor Customer related tables. The Archive Facilitator step retrieves the records from the Facilitator table for the chain's Job ID. It then reads the file names from these records and spawns multiple SMU jobs, using the parameter file names from the Facilitator table as input for the SMU jobs. The parameter files specify the "Table Export" command with other options set in the Archive Preprocess step. Each SMU job is responsible for exporting (archiving) table records from a single table. Multiple SMU jobs may be spawned to export data from a single table (depending on the Archiving Preprocessor Job Block Size parameter value), but one SMU job will never be responsible for exporting from more than one table.

Process Steps	Messages
1. Parameter Validation	• Run Started *Each parameter is listed*
2. Facilitator Job Processing	• The Run Number for this archive/restore process = xxx (xxx being the chain's Job ID. Restore is not a feature of Vendor Customer Archive.) • For each job in the Facilitator for the Run Number: • SMU Job - xxx - Spawned (*xxx being the Job ID)* • Each spawned job will have either of these results: • SMU Job - xxx – Processing completed successfully (*xxx being the Job ID)* • SMU Job - xxx - Failed (xxx being the Job ID)

	• The job slept a total of xxx times, for a Total Sleep Time of yyy seconds. *(xxx and yyy being the counts)* • Run Ended

Restartability Information

This job can be restarted if it fails due to any reason. After restart, the job will continue to process the records based on the status of each Facilitator record. Restarting this job step will restart all spawned jobs automatically.

Major Input

- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page.
- BS_AGENT – The job will consult the Job Manager for the status of jobs that have been spawned.
- Vendor Customer related tables – Each SMU job spawned by the Facilitator job will export a subset of records, from a single table, depending on number of starting and ending keys specified in the parameter file supplied to the Facilitator record.

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

Batch Parameters

Parameter	Description	Default Value
ARCHIVE_RESTORE_ID	Archive Restore ID Required, not overrideable. This value tells the Facilitator to select the archive table process for Vendor Customer related tables.	13 (Vendor Customer Archive)
COMMIT_BLOCK_SIZE	Commit Block Size Required. This value will determine how often database transactions will be committed and can be used for performance tuning. This must be a positive integer.	1000
PROCESSOR_NO	Number of jobs for Facilitator to keep running Required. This value sets the number of jobs to launch and keep submitted/running. Exceeding the number of available VLS is acceptable. This must be a positive integer.	1
SLEEP_TIME	Number of seconds to wait between polling occurrences Required. This value sets the length of time between iterations, which checks	5

	the status of the jobs that are running, and the jobs that are launching for unprocessed Facilitator records as necessary.	
SMU_CTLG_ID	SMU Catalog ID Required, not overrideable. This value tells the Facilitator to create a specific SMU job.	3
UPDATE_STATUS	Update Status Required, not overrideable. This value tells the Facilitator to update the status of each Facilitator record based on the status of its corresponding job in the Job Manager.	Y

Major Output

- Exported CSV or XML files
- FACILITATOR – The Facilitator table is updated to reflect the Job Manager status.

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	This return code is issued under the following conditions: • No Facilitator records found for Run Number = xxx. Nothing to process. (xxx being the chain's Job ID) • SMU Job Parameter File NOT Found – (file name) Record skipped. When this job ends with a return of code Warning, subsequent jobs in the chain are set to Inactive.
Non Fatal Error (8)	This step does not issue this return code.
Failed (12)	The job fails under the following conditions: • One or more parameter(s) are invalid. • One or more required parameter(s) are not entered. • Parameters are invalid. • Run time exceptions for unexpected situations. When this job ends with a return of code Failed, subsequent jobs in the chain are set to Inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain are set to Inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return of code System Failure, subsequent jobs in the chain are set to Inactive.

Sort Criteria

None

Selection Criteria

FACILITATOR records contain the location of input parameter files for use by spawned System Maintenance Utility batch jobs.

Problem Resolution

No database restore is required. Both the Archive Facilitator chain step and each spawned System Maintenance Utility job can be restarted.

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)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	No Facilitator records found for Run Number (Nothing to process).	This is acceptable in the event that there are no records to process. A new chain can be started with only the Post Archiving Process enabled. Otherwise, the reason for the missing records must be investigated before a new chain job is run.	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. **Sample Message:** SLEEP_TIME is required to run the Facilitator	Schedule a new job after entering a valid value for the parameter.	N/A
	Entered parameters are not valid. **Sample Message:** The SLEEP_TIME must be an integer and > 0	Schedule a new job after entering a valid value for the parameter.	N/A
	Failed because of runtime exceptions for an unexpected	Failure reason needs to be investigated before scheduling a new job.	N/A

	situation.		
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Facilitator Job Processing

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	SMU Job Parameter File not found.	If all files are missing, the reason must be investigated before a new chain job is run. If the files can be located and moved to the expected location, the job can be restarted.	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	This return code will be issued when any spawned SMU job fails. **Sample Message:** SMU Job – xxx – Failed	Determine the cause for the spawned SMU job to fail and resolve. The spawned job can be restarted independently if desired, and restarting the chain job will also restart any failed SMU jobs.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

<u>Vendor Customer Archiving Chain: Post Archiving Process Job</u>

Job Name	Post Archiving Process

Recommended Frequency	On Demand This job must be run as part of the Vendor Customer Archiving Chain.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Vendor Customer Archive Facilitator Report

Overview

The parameter directory and the chain parameter file are the only two parameters for this step. All of the user parameters that this step needs are stored in the designated file by the Preprocessor. The first action of this step of the chain is to open the chain parameter file and read these parameters. The second action is to confirm that all facilitator records were processed successfully in step 2 of the chain job. If any facilitator job did not finish successfully, the process ends. The third action is to delete the selected records from the Vendor Customer related tables, perform secondary count and relational data updates, and to write the Vendor Customer Archive Facilitator Report.

Vendor Customer Archive Facilitator Report Sample

Process Steps	Messages
1. Parameter Validation	• Run Started Each parameter is listed • Validating Batch Parameters If any is found to be invalid an error will also be issued • Batch Parameters are valid.
2. Post-processing	• Confirming Facilitator status for Run ID (Run ID of the current Chain Job). • Facilitator completed (number) of (number) job(s) for Run ID (Run ID of the current Chain Job). • If the number of jobs completed is not the same as the total number of jobs: • Post Archive deletion cannot proceed until all Facilitator jobs for Run ID (Run ID of the current Chain Job) have been completed. (Process ends here in this case) • Facilitator status check completed. • Retrieving table information for archived and deleted tables. • Retrieving table information completed. • Deletion of selected records started. • At this point, a series of progression messages will appear. Sample messages:

	- Deleted <number of records> records from <table>. - Deletion of selected records completed. - Rendering Facilitator report started. - Reports output folder mapped (*followed by the HTML & PDF locations)* - Rendering Facilitator report completed. - Number of records deleted from temporary table: (number of records) - Run Ended

Restartability Information

This job cannot be restarted. If the job failed due to any reason, a new job can be scheduled after correcting errors that caused the job to fail – either individually, or as part of a new chain job by disabling the first two jobs in the chain and specifying the same set of parameters.

Major Input

- FACILITATOR – The Facilitator table keeps track of System Maintenance Utility (SMU) jobs to be processed, and is updated by the Facilitator batch job to reflect the Job Manager status. These records can be viewed on the Archive Facilitator Inquiry (QARC) page. This job will check the table to see if any records remain that have not completed successfully.
- BS_AGENT – The job will consult the Job Manager for the status of jobs that have been spawned.
- Chain Parameter File – The Preprocessor step writes this file to communicate all of the settings that this step needs.

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

Batch Parameters

Parameter	Description	Default Value
AMSPARM	Parameter Location at Archiving Preprocessor Job Required. This must be a valid directory containing the file designated by the Parameter File parameter.	$$AMSROOT$$/Parms
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) Required, not overrideable. This file is created during the first job in the chain to convey parameters to this process.	VCUSTArchParams.txt

Major Output

- Vendor Customer Archiving (R_VEND_CUST_ARCH) table - all rows inserted to this table by the pre-processor are updated to mark the Financial Completed indicator to completed (1).
- Vendor Customer Archiving Temporary (VEND_ARCH_TBL) table – all rows inserted to this table by the pre-processor step are deleted in this step.
- Instruction File - Selected records are deleted from these tables:
- Vendor/Customer (R_VEND_CUST)
- Address (R_AD)
- Languages (R_VEND_LANG)
- Vendor Business Type (R_VEND_BUS_TYP)
- Vendor Service Area (R_VEND_SRVC_AREA)
- Vendor Commodity (R_VEND_COMM)
- Authorized Dept. (R_RSTR_USE)
- Prevent Spending (R_PRVNT_SPND)
- Vendor Certification (R_VEND_CERT)
- Vendor User Information (R_VEND_USER)
- Vendor Attachments (R_VEND_ATT)
- Business Types by Commodity (R_BUS_TYP_COMM)
- Vendor Notes (R_VEND_NOTE)
- W-8 Form (R_VEND_W8_FORM)
- Vendor Customer Usage (VEND_CUST_USAGE)
- VCM Tracking Table (VCM_TRCK_TBL)
- Internal Vendor Accounting Data (R_INT_VEND_ACTG)
- System Crosswalk (R_SYS_CROSWALK)
- External Commodity Crosswalk (R_EXT_COMM_XWK)
- Internal Commodity Crosswalk (R_INT_COMM_XWK)
- Electronic Account Profile (R_UTLY_ACTG_PRFL)
- Electronic Billing Inquiry (R_UTLY_BILL_INQ)
- Vendor Invoice Registry (R_VEND_INV_RGSTRY)
- Vendor Notification (R_VENDNOT)
- Collection Agency Agreement (R_CLAGCY_AGRMT)
- Customer Account Options (R_CUST_ACCT_OPT)
- Customer Account Information (R_CUST_ACCT) and Detailed Transaction Listing (R_CUST_ACCT_DET)
- Customer Account Assurance shadow table (SA_CUST_ACCT)
- Escrow Definition (R_ESCROW_DEF) and Escrow History (R_ESCROW_HIST)

- Invoice (R_INV_PRN_INQ)
- Invoice Print History (R_INV_PRN_HIST)
- Payment Plan (R_PYMT_SCHED and R_PYMT_SCHED_DET)
- Payment Plan Print (R_PYMT_SCHED_PRN)
- Payment Plan Print History (R_PYM_SCH_PRN_HIST)
- Receivable History and Reference Query (R_PYMT_DET_INQ_HDR and R_PYMT_DET_INQ_DET)
- Statement Detail (R_STMT_DET)
- Statement (R_STMT_PRN_INQ)
- Statement Print History (R_STMT_PRN_HIST)
- Accounts Receivable Correspondence History (R_RCVB_CORS_HIST)
- Headquarters Account (R_PNT_VEND_CUST)
- Master Addresses (R_MSTR_AD)
- Master Contacts (R_VEND_CNTAC)
- Account Users (R_VSS_USER)
- 1042-S Reporting Information (R_1042S_RPT_INF)
- 1099 Reporting Information (R_TIN_1099_INFO)
- Vendor List (R_VEND_LST_VEND)
- Vendor History (R_VEND_HIST)
- Payment Integration (DISB_INTG)
- Transform Catalog (R_CATFILE)
- Coupon Return (R_COUPON_DET)
- Credit Balance (R_CR_BAL_DET)
- Pending Credit Refund (R_PEND_CR_RFND)
- Pending Collection Agency Referral (R_COLL_RFRL_INPT)
- Pending Write Off (R_PUNR_INPT)

Minor Output

- Headquarters (R_PNT_VEND_CUST) – Vendor counts (VEND_CUST_CT) are updated to reflect the number of remaining vendor's associated to that particular headquarters.
- VSS User (R_VSS_USER) – Vendor user counts (VEND_USER_CT) are updated to reflect the number of remaining vendor user accounts associated to that particular VSS user. Also the Vendor/Customer code associated to the VSS User record may be updated to a remaining Vendor User's Vendor/Customer code associated to the same VSS user, if the previous value had been archived.

Job Return Code

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

Return Code	Condition
Successful (1)	The SMU jobs were confirmed to have completed, and all selected records were deleted.
Warning (4)	This step does not issue this return code.
Non Fatal Error (8)	This step does not issue this return code.
Failed (12)	The job fails under the following conditions: • One or more parameter(s) are invalid. • One or more required parameter(s) are not entered. • Parameters are invalid. • Run time exceptions for unexpected situations. When this job ends with a return of code Failed, subsequent jobs in the chain are set to Inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When this job ends with a return of code Terminated subsequent jobs in the chain are set to Inactive.

Sort Criteria

None

Selection Criteria

VEND_ARCH_TBL records identify the tables and their individual records to be deleted. The temporary records are inserted by the Preprocessor step during the record selection process and are keyed by the Chain Job ID from the Job Manager. This step also removes all temporary records associated with that Chain Job ID after the archive deletions have been completed successfully.

Problem Resolution

Since the job cannot be restarted, if the job ends with any return code (Failed, Terminated or System Failure) a new chain job with the first two jobs in the chain disabled should be scheduled with the same parameters.

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)	The SMU jobs were confirmed to have completed, and all selected records were deleted.	N/A	N/A

Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. **Sample Message:** Parameter AMSPARM directory is required and must be a valid directory.	Enter the correct directory for Parameter Location and schedule a new chain job with only step 3 enabled.	N/A
	Entered parameters are not valid. **Sample Message:** Parameter Chain Parameter File is required to have file extension "txt".	Enter the correct file name for Chain Parameter File and schedule a new chain job with only step 3 enabled.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Post-processing

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new	N/A

		chain job with only step 3 enabled.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

2.2 Utilities Report Processes

The Advantage Utilities report run sheet included in this section is:

- Audit Log Report
- Pre-Archive Budget Report

2.2.1 Audit Log Report

Description

The Audit Log Report captures all logs that are generated as part of the process of monitoring and recording activity of selected tables. Users can monitor changes to tables such as insert, update and delete actions. Two types of Audit Log Reports can be generated – Summary and Detail. A detailed report contains the new and old values for the data columns being updated by the action.

To generate an Audit Log Report, the user must first set up audit logging for any table. It can be done in the "Setup Audit Log Control" page. Then the user must create a new Audit Log Report job and specify the parameters discussed below. After specifying the parameters the job must be submitted to generate the report.

When to Run

As needed.

Major Input

- Tables setup for Audit Log (IN_AUD_LOG_CTL)
- Audit Log table (IN_AUD_LOG)

Other Input

None

Output

Audit Log Report (Summary or Detailed)

Parameters

Batch Parameters

Job	Parameter	Description	Default Values
Audit Log Report	RSRC_NM	List of Table Names for which the report has to be generated. The format of table list is comma-separated tables such as TABLE1, TABLE2, TABLE3 and so on. For example, you can setup table list as "**BS_AGENT, R_MSG, R_UNIT**". You can leave the field as BLANK to include all tables for which audit logging is enabled.	
	START_DT	Date from which the report should be generated. The format of start date is MM/DD/YYYY. For example, you can set up the start date as "**05/01/2002**".	
	END_DT	Date till which the report should be	

		generated. The format of end date is MM/DD/YYYY. For example, you can setup the end date as “**05/31/2002”**.	
	USID	User login Id of the online user or batch job Id for offline processes. The format of user id is the comma-separated user ids or job ids such as USERID1, USERID2 and so on or JOBID1, JOBID2 and so on. For example, you can setup the user id as “**sa”**. You can leave the field as BLANK to include for all user ids or job ids for offline jobs.	
	REPORT_TYP	Summary or Detail. A detailed report contains the new and old values for the data columns being updated by the action. The format of this report type is either 1 – for Summary Type or 2 – for Detailed Type. For example you can setup the report type as “**1”** or “**2”**.	1

Sort Criteria

None

Selection Criteria

- Select records from Audit Log table IN_AUD_LOG_CTL where:

  Resource Name = Table List specified as batch parameter and
- User Id running the online batch job or Job Id for offline process = User Id specified as batch parameter
- For each table select records inserted, updated and deleted between Start Date and End Date.
- If (Report Type is Summary) then display only new values.
- If (Report Type is Detail) then display both old and new values.

Troubleshooting

No database restore is required. Correct the problem and rerun the job executing the program. No restoration of datasets or files from backups is required for this program.

2.2.2 Pre-Archive Budget Report

Job Name	Pre-Archive Budget Report
Recommended Frequency	On Demand
Single Instance Required?	No
Can be restarted?	No
Reports generated	Pre-Archive Budget Report

Overview

This report helps to identify unfinished budget activity. The most common use for the report is before running the Budget Archiving chain job, which archives data from budget tables. An alternative use for the report is at year end to find pending activity for which a decision should be made to complete the activity or it will fail when the final approval is applied. Lastly, the report can be run on an ad hoc basis to review one or more budget lines.

No budget line should be archived unless all current activity against the line has finished. While such a decision may be initially made by knowing no new activity will be allowed by BFY Staging, or by knowing that a task, work order, project, grant or other cost accounting entity has been completed, these are only the first level of identification – new activity. Unfinished activity is any non-zero amount for a pending bucket or one of the other buckets identified with a batch parameter of the report.

The selection logic of the report is the same as that of the Budget Archiving process when it comes to the selection of budget lines and activity records. The report will list unfinished budget lines by key fields, listing any transactions identified for such activity. Should all selected budget lines for a level have no unfinished amounts then the report will state: All lines are finished.

The report will not just look at budget lines but also the activity records to catch the situation where multiple transactions could sum to a zero impact at the budget line level. These activity records will be listed in the report by Transaction Code, Department and ID.

This chain implements the Data Warehouse Archived Record Queue process, which is used to retain the table names and key values for records that are archived and deleted. The infoAdvantage reporting system uses this information to distinguish between records that have been archived (as valid historical data) before deletion and records that have simply been deleted from the system (as no longer needed). The process is enabled for most archiving processes when the value of Application Parameter Enable Data Warehouse Archived Record Queue (ENABLE_DW_ARCH_QUEUE) is *True*. Please see the *CGI Advantage System Administration Guide* for more information regarding this process and the Application Parameter.

Process Steps	Messages
1. Parameter Validation	• Run Started Each parameter is listed. • Validating Batch Parameters If any is found to be invalid an error will also be issued. • Batch Parameters are valid. or

	• Batch Parameters are not valid.
2. Report Generation	• Record selection started. • Record selection completed. • Rendering Pre-Archive report started. • Reports output folder mapped (followed by the HTML & PDF locations) • Rendering Pre-Archive report completed. • Number of records deleted from temporary table: (number of records) • Run Ended

Major Input

- Budget Level 1 of a Budget Structure (BUD_STRU_##_LVL_1)
- Audit/Activity Level for Budget Structure (BUD_STRU_##_LVL_#)

Batch Parameters

Parameter	Description	Default Value
AMSPARM	Parameter Location at Archiving Preprocessor Job Required. This must be a valid directory containing the file designated by the Parameter File parameter.	$$AMSROOT$$/ Parms
BFY	Budget Fiscal Year Conditionally Required. If the budget structure is keyed by BFY, then this parameter is required. For Multi-Year budgets, this parameter is prohibited.	No Default
BKT_ID	Bucket IDs Optional. Enter one or more Bucket IDs for evaluation.	11,12,13,14,22,23,25
BUD_STRU_ID	Budget Structure ID Required. Must be a valid ID in the Budget Structure (BUDST) table.	No Default
CLIENT_NM	Client Name for Reports Optional.	No Default
COMMIT_BLOCK_SIZE	Commit Block Size	1000

	Required. This value will determine how often database transactions will be committed and can be used for performance tuning. The value must be a positive integer.	
PARM_FILE	Input Selection File Required. Enter the name of the parameter file for budget archiving; this file must exist in the directory specified by the Parameter Location at Archiving Preprocessor Job parameter.	BudArchiveSelect.txt
JOB_TYP	Job Type Required, not overrideable. Indicates to the system whether to run shared logic as the Pre-Archive Report job or as part of the Budget Archiving chain job. This must always be set to Pre-Archive for the Pre-Archive Budget Report process.	Pre-Archive

Major Output

- Budget Pre-Archive Report
- Budget Archiving Temporary (BUD_ARCH_TMP) table
- ARCH_DW_QUEUE_REC – The Data Warehouse Archive Record table keeps track of the values for all primary keys of each record being archived and deleted. The table name and attribute names are stored in ARCH_DW_QUEUE_TBL and linked to this record by unique ID.
- ARCH_DW_QUEUE_TBL – The Data Warehouse Archive table keeps track of the table name and primary key attribute names of each table being archived. If a record already exists for the table being archived, it will reuse the record. If it does not yet exist, it will be created.

Job Return Code

The following table shows the potential job return codes for the Budget Pre-Archive Report job.

Return Code	Condition
Successful (1)	The report was generated successfully.
Warning (4)	This job does not issue this return code.
Non Fatal Error (8)	This return code is issued under the following conditions: - No eligible records found in the level 1 table of the budget

	structure for the parameters entered.
Failed (12)	The job fails under the following conditions: One or more parameter(s) are invalid. - One or more required parameter(s) are not entered. - The Input Selection File is not found in the directory specified by the Parameter Location at Archiving Preprocessor Job parameter. - 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 Criteria

The report is sorted by COA elements for Budget Level 1 of the Budget Structure (BUD_STRU_##_LVL_1), Transaction Code, Transaction Department Code, and Transaction ID.

Selection Criteria

The Input Selection File parameter identifies a text file that is required to be present in the parameter directory. It can be empty when the budget is keyed by BFY, but at least one set of selection criteria must be specified for no-year budgets (those without a BFY in their line definition). Budget lines that are defined to BFY 9999 will require the entry of 9999 in the BFY batch parameter.

Each set of selection criteria must start with the _PARAM_LINE_ separator. Selection criteria values in the input file may use commas to specify more than one value per COA element (but only one comma-separated list of values is allowed per PARAM LINE section). The "_" (underscore) may be used as a wildcard to match any one character. The "%" (percent sign) may be used as a wildcard to match 0 to *n* characters.

While the parameter file does allow for complex selection criteria, the easiest method to build a parameter file (and avoid parameter file edits for what is not allowed) is to run a series of database queries to get a list of UNID values from level 1 of the budget structure being reported on. Each budget line is uniquely defined by this UNID value and it is very straightforward to put into the file under one parameter line. Please exercise care not to exceed any limits for SQL IN conditions for the type of database. Such limits are often 1000, which is most likely higher than the manageable number of budget lines selected from level 1 of a budget structure using COA instead of BFY.

Input File Example:

```
_PARAM_LINE_
FUND_CD=010
DEPT_CD=010
APPR_CD=0028,0029,0030
_PARAM_LINE_
FUND_CD=150
DEPT_CD=020
APPR_CD=0001
_PARAM_LINE_
```

FUND_CD=20_
DEPT_CD=010,020
APPR_CD=1%
_PARAM_LINE_
UNID=22390
_PARAM_LINE_
UNID=22441

Within each _PARAM_LINE_ group, the selection criteria are grouped as AND statements. Then, the _PARAM_LINE_ groups are grouped as OR statements to form the full selection criteria. The above example would be interpreted as:

(Fund = '010' **and** Department = '010' **and** (Appr Unit is '0028' **or** '0029' **or** '0030'))

or (Fund = '150' **and** Department = '020' **and** Appr Unit = '0001')

or (Fund matches '20_' **and** (Department is '010' **or** '020') **and** Appr Unit matches '1%')

or (UNID = 22390)

or (UNID = 22441)

It is acceptable to have a given record satisfy the criteria for multiple _PARAM_LINE_ groups. This may occur depending on how broad or specific the selection criteria are defined for each group. When this situation occurs, the Pre-Archive Budget Report will properly select the record just once.

Input File Rules:

1. The input file cannot contain BFY, because the process requires a BFY parameter to be specified where applicable, and is otherwise prohibited.
2. The input file must contain at least one PARAM LINE section with one key field for selection and at least one value for that key field if the budget structure is not keyed by BFY.
3. The input file can only include budget level 1 COA elements list or UNID.
4. If the selection criteria key is not UNID, then the full key to a COA element must be specified (except FY). For example, running with just Program will fail if Department is not also specified. Selecting on Unit will fail if Department is not also specified, but there does not have to be an FY, which is a key to each Unit.
5. The PARAM LINE section will be controlled several ways to prevent performance degradation but allow for some flexibility in selection when a COA list is specified:
    a. Wildcard and comma features cannot be combined into the same line, so the following is not allowed:

        _PARAM_LINE_

        DEPT_CD=010,02%

        Multiple lines may use the wildcard feature, so the following is allowed:

        _PARAM_LINE_

        DEPT_CD=01_

        MJR_PROG=100%

        Multiple wildcards are allowed in a single line, so the following is allowed:

        _PARAM_LINE_

        DEPT_CD=010

MJR_PROG=%100%

b. Only one line can use the comma feature, so the following is not allowed:

_PARAM_LINE_

DEPT_CD=010,02

0MJR_PROG=10011, 10012

Instead, the same can be achieved with multiple lines, for example:

_PARAM_LINE_

DEPT_CD=**010**

MJR_PROG=10011,10012

_PARAM_LINE_

DEPT_CD=**020**

MJR_PROG=10011,10012

6. Wildcard characters are prohibited when selecting by UNID.

Problem Resolution

If the job ends with a return code other than Successful after completing parameter validation, a new job should be scheduled after the job log has been reviewed (the failed job should not be restarted).

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)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	Required parameters are not entered. Sample Message: Parameter BFY is required for Structure 29.	Schedule a new job after entering a valid value for the parameter.	N/A

	Entered parameters are not valid. Sample Message: Parameter Job Block Size value 'ABC' is invalid, it must be a positive integer.	Schedule a new job after entering a valid value for the parameter.	N/A
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A

Step 2: Report Generation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	The report is generated successfully.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non Fatal Error (8)	No records found.	Confirm selection criteria before scheduling a new job.	N/A
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling 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 before scheduling a new job.	N/A