CGI Advantage® 4

General Accounting
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 General Accounting Batch and Chain Processes

In CGI Advantage Financial, General Accounting enables the entering of accounting transactions, processing those transactions to update the journals and ledgers, customizing ledger inquiries to meet informational needs, and setting account balance and fund balance controls.

The General Accounting processes are:

- ABAL Process
- Accounting Period Close
- Accounting Template Mass Change Load
- Accounting Template Mass Update
- Accounting Template Validation
- Activity Override Batch
- Annual Close Chain
- Annual Financial Reporting Data Load
- Automated Accrual Process
- Automated Accrual Clearing Process
- Automated Bank Account Transfer
- Begin Day Balance
- Cash and Fund Balance Synchronization Process
- Clearing Account Maintenance
- Contract Roll Process
- Department Transaction Listing
- Identify and Archive Stale Gaps
- JV Unclaimed Property
- Journal Engine
- Journal Rebuild
- Ledger Engine
- Open Activity and Budget Roll

- Open Activity and Budget Update
- Open Activity Lapse
- Open Activity Options by Department
- Open Activity Roll
- Partial Transaction Ledger Engine
- Payroll Transaction Generator
- Populate FY Beginning Balance
- Pre-Annual Close Sweep
- Rebuild Ledger
- Receiver Accrual
- Revolving Fund Replenishment
- Treasurer Interface Process
- Treasury Cash Balance

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

2.1.1 ABAL Process

Chain or Job Name	ABAL Process
Recommended Frequency	This job should be run daily, after the day's transactions have processed, but it cannot be run more than once a day. On days when the Outstanding Check process, the Deposit Reconciliation process, or any other process that produces cash entries is run, it is important to make sure the ABAL Process runs after these jobs complete, as these jobs update tables that feed into the ABAL Process.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

The Bank Account Balance (ABAL) page provides the user with a snapshot of the Estimated Bank Balance in a bank account at a specific point in time. The ABAL batch job updates this page with a single record per bank account for the Balance Date provided as a parameter.

A record is inserted to the ABAL page for each 'Active' or 'Closing' bank account code on the Bank Account (BANK) page. Banks that are marked as 'Closed' on the Bank Account page are not processed by the ABAL job. The Beginning Balance amount is calculated by carrying forward the Ending Book Balance for that bank account from the previous cycle (that is, the last time this offline process was executed). The Credits and Debits amount fields are calculated by totaling all of the new debit and credit records from the Cash Journal added since the last run of the ABAL process. The Outstanding Checks amount is obtained from Check Reconciliation (CHREC) and the Outstanding Deposits amount is obtained from Deposit Reconciliation (DPREC). The rest of the fields on the page, Ending Book Balance and Estimated Bank Balance, are formula-driven fields.

The ABAL Process keeps track of the last record it read on the Cash Journal (JCASH) using the Journal Log (JLOG) page. This enables the process to only select the new records on the Cash Journal so that old records are not recounted.

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 will be displayed in the log. • Batch Parameter validation completed
2. Selection of Records & ABAL Updates	• Selecting eligible records • If the selection returns 0 records, then the following message will be issued: "No eligible record found".

	• Number of records (count) selected will be displayed • Run Ended.

Restartability Information

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.

For performance reasons, the Identify and Archive Stale Gaps job should be run periodically to remove gaps in journal records being tracked and processed by this program. Please refer to the Identify and Archive Stale Gaps run sheet for more details on scheduling and recommended parameters.

Major Input

- Cash Journal (JRNL_CASH)
- Bank (R_AP_BANK_ACCT)
- Check Reconciliation (R_AP_CHK_RECON)
- Deposit Reconciliation (R_AP_DPS_RECON)
- Journal Log (JRNL_LOG)
- Bank Account Balance (R_ABAL)

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
DY_DATA_RTND	(Optional) Number of days that the old data will be retained. This field does not have a default value. If the value is left blank, then no historical records are deleted. If a value is entered, then all records prior to the balance date minus this value will be deleted for each bank.	(blank)
BALANCE_DT	(Required) The parameter is used as the Balance Date for all of the ABAL records created. Please enter as mm/dd/yyyy. The Date entered in this field should be less than or equal to the Application Date and this date must be greater than the Balance Date of any existing ABAL record.	(blank)

COMMIT_BLOCK_SIZE	(Required) A block size used for performance reasons to control the number of records created at once. If left blank, 1000 is assumed.	1000
MAX_PREFETCH_COUNT	(Required) A block size used for performance reasons to control the number of records retrieved from the Cash Journal, Deposit Reconciliation, and Check Reconciliation tables. If left blank, 1000 is assumed.	1000

Major Output

- Journal Log (JRNL_LOG) – records last record read from Cash Journal for next run
- Bank Account Balance (R_ABAL)

Job Return code

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

Return Code	Condition
Successful (1)	All the selected records are processed successfully
Warning (4)	This job does not use this return code
Non-Fatal Error (8)	This job does not use 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 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

- BANK_ACCT_CD

Selection Criteria

- JRNL_LOG

Selects the last record entered on this table with the process ID of 'ABAL'. This enables the process to only select the new records on the Cash Journal so that old records are not recounted.

- JRNL_CASH
  - Records selected with REC_NO greater than the Ending Journal Record Number on the JRNL_LOG record for the last run of the ABAL process, having BANK_ACCT_CD where the corresponding R_AP_BANK_ACCT record has Bank Account Status of 'Active' or 'Closing'.
  - The REC_NO from the last JRNL_CASH record selected from the current run is used as the Ending Journal Record Number on the JRNL_LOG record created for the current run.
- From the R_AP_CHK_RECON
  - CHK_EFT_ISS_DT <= Balance Date parameter entered or defaulted
  - CHK_STA = OUTSTANDING
- From the R_AP_DPS_RECON
  - DPS_DT <= Balance Date parameter entered or defaulted
  - DPS_STA = OUTSTANDING

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the parameters are validated successfully	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	Job failed due to Fatal conditions: 1) Encounters any runtime exceptions and 2) Invalid parameters	If the job fails for parameter edits, correct the parameter and run a new job. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and run a new job.	If another instance of the job has already been run for a subsequent day, the date skipped cannot be recreated.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. The job can either be restarted or	If another instance of the job has already been scheduled and ran successfully, then

		schedule a new job.	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.

2.1.2 Accounting Period Close

Job Name	Accounting Period Close
Recommended Frequency	On demand as necessary to limit accounting and budget activity by accounting period.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes

Overview

This program can update either the Closed indication (often referred to as a soft close) or the Closing Process Run indication (often referred to as a hard close), or both on the Accounting Period (APD) page. While the soft close can be updated manually by those with adequate security, this process can be scheduled to run automatically to avoid missing the manual soft close.

The following steps are performed by this process.

Process Steps	Messages
1. Parameter Validation	• Run started. • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value is displayed in the log.
2. APD Update	• No messages are issued for this step.
3. Report Generation	• Report output folders mapped • Rendering report started • Rendering report completed • Run ended

Input

Accounting Period (APD / R_APD)

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
Accounting Period (ACTGPRD)	A required selection parameter to specify a single period close. Depending on the Close Type parameter, more than one period may be closed.	No default
Client Name	An optional name of client for the	No default

(CLIENT_NM)	report header.	
Open Period Method (CLSPREVPRD)	A required parameter to control whether only the period specified in the Accounting Period parameter is closed or all periods prior to that period in the Fiscal Year. Valid values are *0* (no) and *1* (yes). When running with *0*, if the program encounters a prior period in the Fiscal Year, the job will fail listing that as the reason.	No default
Close Type (CLS_TYP)	A required parameter to control the type of closing. Valid values are *1* (hard close), *2* (soft close), or *3* (both). If left blank, the default is *1*.	1
Fiscal Year (FY)	A required selection parameter to tie the Accounting Period parameter to a fiscal year. Must be entered as CCYY.	No default

Output

- Accounting Period (APD / R_APD)
- Report (Actg Period Close Program)

Job Return Codes

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

Return Code	Condition
Successful (1)	The job ends as successful when all parameters are valid, and one or more periods are closed.
Warning (4)	The job does not use this return code.
Non-Fatal Error (8)	The job does not use this return code.
Failed (12)	The job fails under the following conditions: • Parameters are invalid. • Invalid accounting period selected. • Open periods prior to the one being closed with a Close Period Method of 0.
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.

Sort Criteria

PER (Accounting Period from R_APD)

Selection Criteria

PER = ACTGPRD parameter and FY = FY parameter

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.

2.1.3 Accounting Template Mass Change Load

Chain or Job Name	Accounting Template Mass Change Load
Recommended Frequency	On demand
Single Instance Required	Yes
Can be restarted?	No
Reports generated	None

Overview

As Accounting Templates are setup as of a point in time to provide all or some of the necessary chart of accounts (COA) for one or more accounting events, over time a template may no longer contain a partial or complete COA string. While the Accounting Template Validation process can identify templates with invalid and missing required COA, it is incumbent upon a user to update each template on the Accounting Template (ACTPL) page using the report produced. As soon as the update is made, the updated template provides different COA. This may cause issues if the old form of the template should be used in the prior year and the new form in the new year. While users can manually account for the differences, interfaces cannot as the Accounting Template ID does not vary between years. The process of maintaining year templates often requires a copy of the prior year's template to create a new year template with a different ID.

Alternatively, a two-step approach can be taken: Accounting Template Mass Change Load process and Accounting Template Mass Update process. The primary advantage of this approach is that templates are not 'officially' updated until the second process is run, allowing authorized users time to plan for and update templates outside of the Accounting Template page, using the Accounting Template Mass Change (ACTPLMC) page instead.

All templates or a subset using selection criteria can be loaded to the ACTPLMC page. Subsets are useful when a department is undergoing a re-org or when templates are defined to a particular fiscal year or budget fiscal year, but there is no reference to that year in the Accounting Template ID.

The Accounting Template Mass Change Load process loads templates to Accounting Template Mass Change (ACTPLMC) page loads with COA values in both Current and Next Year COA fields. Users then have to only update those templates that need to change in the Next Year COA fields.

Process Steps	Messages
1. Parameter Validation	• Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value is displayed in the log. • When Validation Year is blank, the following message is logged: "Current FY <YYYY> will be used as Validation Year" • Batch Parameter validation completed
2. Verifying records do not exist in ACTPLMC	• After validating parameters, the following message is logged: "Begin Accounting Template Mass Change process"

	• If the template records to be selected from ACTPL already exists in ACTPLMC following error is issued: "Records exist in ACTPLMC matching selection criteria specified. Please purge existing Mass Change records" • If records do not exist in ATCPLMC the process proceeds after logging the following message: "No records exist in ACTPLMC matching selection criteria specified"
3. Selection of ACTPL records and Loading of Records to ACTPLMC	• If the selection returns 0 records, then the following message is issued: "No Accounting Template record found meeting the selection criteria". • After loading the records following message is logged: "Added (count) records into Accounting Template Mass Change" • After committing the loaded records, the following message is logged: "Successfully Committed Accounting Template records to Accounting Template Mass Change" • Where the commit fails following message is logged: "Failed to Commit Accounting Template records to Accounting Template Mass Change" • At the end, the following message is logged: "Completed Accounting Template Mass Change process"

Implementation Considerations

1. Please note that information on the Accounting Template Mass Change (ACTPLMC) page can be left for a period of time to provide an audit trail of changes. Before running a new round of updates, a System Maintenance Utility (SMU) job should be run after successfully loading the updated records to the Accounting Template (ACTPL) page through Accounting Template Mass Update process to purge the table. Failure to do so results in the entire load rejecting, as at least one Accounting Template ID already exists. Alternatively, the data can be purged prior to running next instance of Accounting Template Mass Change process.
2. If the Accounting Template Mass Change Load process is run for individual Departments, you can run the process without purging the data.

Major Input

- Accounting Template (ACTPL / R_ACTG_TMPL)

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

Selection Year Flag (SEL_YEAR_FL)	Required type of year for selection. Valid values are FY or BFY. Required even if not using the Selection Year parameter.	FY
Selection Year (SEL_YEAR_VL)	Optional year used for selection from the Accounting Template year field identified in the Selection Year Flag parameter. Please enter as YYYY.	No Default
Selection Department (SEL_DEPT)	Optional selection criteria for one or more departments using the security department of a template and not the department inferred. Please separate multiple codes with commas.	No Default
Validation Year (VAL_YEAR_VL)	Required fiscal year used for COA validation on the Accounting Template Mass Change records of the Next Year fields of COA keyed by Fiscal Year. Please enter as YYYY. If left blank, the defined default Fiscal Year of Application Date defaults.	No Default

Restartability Information

None

Major Output

- Accounting Template Mass Change (ACTPLMC / R_ACTG_TMPL_MC)

Job Return Code

This table shows the potential job return codes.

Return Code	Condition
Successful (1)	All of the selected accounting template records are loaded successfully.
Warning (4)	No eligible records found. This could be because of the following reason:

	• No active accounting template records exist matching selection criteria
Non-Fatal Error (8)	N/A
Failed (12)	The job fails under the following conditions: • Parameters are invalid. • At least 1 Accounting Template ID already exists on ACTPLMC • 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

N/A

Selection Criteria

Only the Active Accounting templates get selected based on the parameters specified.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully and records were found and copied.	N/A	N/A
Warning (4)	Job ended with a Warning because there is no active Accounting Template table record. **Sample Message:** No Accounting Template record found meeting the selection criteria.	Please verify whether the selection criteria specified is correct and there are Active Accounting templates meeting the selection criteria.	Alternatively, the job can be rescheduled with a different set of parameters.

Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	Job failed due to Fatal condition. Job failed because the Accounting Template records already exist on Mass Change (ACTPLMC). **Sample Message:** Records exist in ACTPLMC matching selection criteria specified. Please purge existing Mass Change records	In this step, the job can fail under the following two conditions. • Entered parameters are invalid. • Encounters any runtime exceptions If the job fails because of the runtime exceptions, investigate the error logs and address the issues. If not repeating parameters accidentally, purge the existing records from Mass Change (ACTPLMC) using System Maintenance Utility (SMU) and then run the job again.	Take appropriate action based on the failure conditions. Alternatively, the job can be rescheduled with a different set of parameters.
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.	N/A
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of database server or network issues.	Investigate the reason for the System Failure before scheduling a new job.	N/A

2.1.4 Accounting Template Mass Update

Chain or Job Name	Accounting Template Mass Update
Recommended Frequency	On demand
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

As Accounting Templates are set up as of a point in time to provide all or some of the necessary Chart of Accounts (COA) for one or more accounting events, over time a template may no longer contain a valid or complete COA string. While the Accounting Template Validation process can identify templates with invalid and missing required COA, it is incumbent upon a user to update each template on the Accounting Template (ACTPL) page using the report produced. As soon as the update is made, the updated template provides different COA. This may cause issues if the old form of the template should be used in the prior year and the new form in the new year. While users can manually account for the differences, interfaces cannot if the Accounting Template ID does not vary between years. The process of maintaining yearly templates often requires a copy of the prior year template to create a new year template with a different ID.

Alternatively, a two-step approach can be taken: Accounting Template Mass Change Load process and Accounting Template Mass Update process. The primary advantage of this approach is that templates are not 'officially' updated until the second process is run, allowing authorized users time to plan for and update templates outside of the Accounting Template page, using the Accounting Template Mass Change (ACTPLMC) page instead.

All templates or a subset using selection criteria can be loaded to the ACTPLMC page. Subsets are useful when a department is undergoing a re-org or when templates are defined to a particular fiscal year or budget fiscal year, but there is no reference to that year in the Accounting Template ID.

The Accounting Template Mass Update process can load all templates from ACTPLMC back to the Accounting Template (ACTPL) page or a subset using selection criteria. Subsets are useful when a department is undergoing a re-org.

Of special note is a feature to update templates with an optional Template FY and BFY parameters. When the original template was defined with either or both a Fiscal Year or Budget FY and there is no need to maintain the old template (see earlier discussion about different template IDs), these optional parameters are used to update either or both years. Additionally, if not all templates contain a year, that is another reason for breaking up the ACTPLMC data into separate updates. For mid-year re-organizations, the Template FY and BFY parameters can be left blank to retain the existing values on the Accounting Templates. For templates setup with multi-year BFY (9999), it is advisable to run through Accounting Template Mass Change Load and Accounting Template Mass Update in a separate cycle.

Process Steps	Messages
1. Parameter Validation	• Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value is displayed in the log with the following message:

	"Problem with one or more of the Batch Job Parameters" • Batch Parameter validation completed
2. Selection of Records from ACTPLMC	• After validating parameters, the system begins processing ACTPLMC records after logging the following messages: "Begin processing set of Actg Template Mass Change records." "About to query for Actg Template Mass Change records." • When no records are found in ACTPLMC that meet the selection criteria, the following message is logged and the job terminates: "No Records Selected from Actg Template Mass Change." • When a Template ID is found in ACTPLMC that does not exist in ACTPL, the system logs the following messages and proceeds to the next record: "Actg Template ID <Template ID> not found in ACTPL." "Problem encountered trying to migrate data for Template ID <Template ID>"
3. Updates to ACTPL	• The process then loads the records to ACTPL and for each block of records, as defined in Progression Counter, logs the following message: "Processed <n> Mass Change records. Last Processed Template ID: <Template ID>" • After completion of the job, the process logs the following messages: "Completed processing of Actg Template Mass Change records." "Total Records Picked from Actg Template Mass Change: <n>" "Total Records updated in Actg Template: <n>"

Restartability Information

None

Major Input

- Accounting Template Mass Change (ACTPLMC / R_ACTG_TMPL_MC)

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
Fiscal Year (TEMPL_FY)	Optional Fiscal Year for updating Accounting Templates. Please enter as YYYY.	No Default
Budget Fiscal Year (TEMPL_BFY)	Optional Budget Fiscal Year for updating Accounting Templates. Please enter as YYYY.	No Default
Selection Department (SEL_DEPT)	Optional selection criteria for one or more departments using the security department of a template and not the department inferred. Please separate multiple codes with commas.	No Default
Modified By User (MOD_USER_ID)	Optional Modify User ID to update the corresponding Accounting Template field. Specify a valid User ID as this field is not validated. If left blank, the User ID from the template will be retained.	No Default
Select Block Size (SELECT_BLOCK)	Required performance parameter to control the number of records selected from ACTPLMC. If left blank, 1000 defaults.	1000
Commit Block Size (COMMIT_BLOCK)	Required performance parameter to control the number of records saved in a block to ACTPL. If left blank, 100 defaults.	100
Progression Counter (PROG_CTR_SZ)	Required parameter to specify the interval in number of records the program should process for logging progression messages. If left blank, 5000 defaults.	5000

Major Output

- Accounting Template (ACTPL / R_ACTG_TMPL)

Job Return Code

- This table shows the potential job return codes.

Return Code	Condition
Successful (1)	All of the selected accounting template records are processed successfully.
Warning (4)	No eligible records found. This could be because of the following reasons:

	• ACTPLMC is empty. • ACTPLMC contains no records that match selection criteria.
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)	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

Accounting Template Mass Change records are sorted by Template ID.

Selection Criteria

When supplied, ACTPLMC records that match the Selection Department parameter value(s) are selected.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully and records are selected.	N/A	N/A
Warning (4)	Job ended with a Warning because there is no Accounting Template Mass Change record. **Sample Message:** No Accounting Template Mass Change record found.	Verify whether the selection Department(s) is correct.	Alternatively, the job can be rescheduled with a different set of Department(s).
Non-Fatal Error (8)	N/A	N/A	N/A

Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following two conditions. • Entered parameters are invalid • Encounters any runtime exceptions If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error, and reschedule the job.	N/A
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.	N/A
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of database server or network issues.	Investigate the reason for the System Failure before scheduling a new job.	N/A

2.1.5 Accounting Template Validation

Chain or Job Name	Accounting Template Validation
Recommended Frequency	On demand
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes – An Exception Report

Overview

As Accounting Templates are setup as of a point in time to provide all or some of the necessary chart of accounts (COA) for one or more accounting events, over time a template may no longer contain a partial or complete COA string. The Accounting Template Validation process can be run to identify and report on templates with invalid and missing required COA, it is incumbent upon a user to update each template on the Accounting Template (ACTPL) page using the report produced. As soon as the update is made, the updated template provides different COA. This may cause issues if the old form of the template should be used in the prior year and the new form in the new year. While users can manually account for the differences, interfaces cannot, as the Accounting Template ID does not vary between years. The process of maintaining year templates often requires a copy of the prior year's template to create a new year template with a different ID.

The Accounting Template Validation process performs the following validations on the selected active Accounting Templates:

1. COA elements are valid and active.
2. COA elements adhere to those Requirement Element Tables listed in a batch parameter. Keep in mind that templates may not have enough data to match the keys of a required element rule, nor should every template have a COA in each required field, as they can be manually entered if templates are not 100% complete. These are reported, but are not necessarily problems. Any required elements that cannot be read are not reported.

All accounting templates can be validated in a single run or multiple runs can validate subsets of templates. Subsets are useful when a department is undergoing a re-org or when templates are defined to a particular fiscal year.

The process generates a report indicating the template records that failed validation with errors. The templates that fail validations can be inactivated on the Accounting Template page by setting the Deactivate Invalid Accounting Templates parameter to Yes. Note: In such cases, the inactivated template's Change Management section will continue to reflect the original information.

Process Steps	Messages
1. Parameter Validation	• Before the process begins validating the parameters the following message is logged: "Begin Processing" • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value is displayed in the log and additionally, the following message is logged:

	"Problem with one or more of the batch parameters"
2. Selection & Validation of Records	• After parameter validation, if there are templates that fail validation, the process will initialize the reports and log the following message: "Initialize the Report for errant ACTPL records" The path where reports (PDF/HTML) are stored is displayed. • When Validation year is blank the following message is logged: "Current FY <YYYY> will be used as Validation Year" • Just before selecting Accounting Template records, the following messages are logged: "Begin processing set of R_ACTG_TMPL" "About to query R_ACTG_TMPL" "Begin processing R_ACTG_TMPL returned" • Once validation is complete for the selected templates, the following messages is logged: "End processing R_ACTG_TMPL returned" "Completed processing R_ACTG_TEMPL" • The process then logs count of records, as follows: "Records processed: (count)" "Records that failed edits: (count)" "Records that passed edits: (count)"
3. Create Exception Report	• When the process starts building the report, the following message is displayed in logs: "Rendering report started" • After completing the report, the following message is displayed in logs: "Rendering report completed"

Implementation Considerations

1. While validating COA elements for the upcoming year, if the New Year (NYTI) process has not been scheduled, the COAs can fail. In such cases, ensure the Deactivate Invalid Accounting Template parameter is set to *No,* as there is no way to change back to active with this process.
2. If there are templates defined with a budget fiscal year (BFY), validation with subsequent fiscal years will assist with both BFY 9999 and specific years that remain open for two or more fiscal years.
3. Templates defined with a fiscal year (FY) can be validated with this process, but that will serve little purpose unless that year is included in the Accounting Template ID and there is a subsequent load of templates for the next year with new IDs.

Restartability Information

None

Major Input

- Accounting Template (ACTPL / R_ACTG_TEMPL)
- COA Required Elements such as Department Object-Revenue Requirements (REQELM / REQ_DEPT_OBJ_REV)

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
Selection Year (SEL_YEAR_VL)	Optional year for record selection. Please enter as YYYY.	No Default
Validation Year (VAL_YEAR_VL)	Required fiscal year used for COA validation. Please enter as YYYY. If left blank FY of System Date defaults.	No Default
Selection Department (SEL_DEPT)	Optional selection criteria for one or more departments using the security department of a template and not the department inferred. Please separate multiple codes with commas.	No Default
Required Element Tables (REQD_ELEM_TBL_NMS)	Optional listing of one or more Requirement Element tables used for validation. Enter the database table name, separating multiple values with commas.	No Default
Deactivate Invalid Templates (DEACTV_TEMP_FL)	Required parameter indicating whether or not templates found to be invalid will be automatically deactivated. Valid values are Yes or No. Default value is No if the parameter is left blank.	No
Select Block Size (SELECT_BLOCK)	Required performance parameter to control the number of records committed in a save. If left blank, 1000 defaults.	1000
Commit Block Size (COMMIT_BLOCK)	Required performance parameter to control the number of records selected for a processing round. If left blank, 100 defaults.	100

Progression Counter (PROG_CTR_SZ)	Required parameter to specify the interval in number of records the program should process for logging progression messages. If left blank, 5000 defaults.	5000

Major Output

- Exception Report
- Accounting Template (ACTPL / R_ACTG_TEMPL)

Job Return Code

This table shows the potential job return codes.

Return Code	Condition
Successful (1)	All of the selected accounting template records are processed successfully.
Warning (4)	No eligible records found. This could be because of the following reasons: • No active Accounting Template records exist for selection parameters
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)	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 accounting templates are sorted by Accounting Template ID.

Selection Criteria

Only Active accounting templates are selected in each of the sub selections below:

- When the Selection Year parameter is used, only templates with that year as the Fiscal Year are selected.
- When the Selection Year parameter is blank, only templates with the Fiscal Year field blank are selected.

- When the Selection Department parameter is used, only templates with the 'security' Department (the one found in the General Information section and not the one that defaults to transactions) are selected.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully and records are selected.	N/A	N/A
Warning (4)	Job ended with a Warning because there is no active Accounting Template records. **Sample Message:** No Active Accounting Template table record found:	Either set the Accounting Template record to "Active" or add a new record on the Accounting Template table with the Active status and restart the job.	Alternatively, the job can be rescheduled with a different set of parameters.
Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following two conditions. • Entered parameters are invalid • 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.	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.
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.	N/A
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of database server or network issues.	Investigate the reason for the System Failure before scheduling a new job.	N/A

2.1.6 Activity Override Batch

Chain or Job Name	Activity Override Batch
Recommended Frequency	Daily before sending data to external system
Single Instance Required	No
Can be restarted?	Yes
Reports generated	Yes

Overview

The Activity Override Batch job populates and updates Activity Override (ACTOV) data with valid Chart of Accounts (COA) strings available for use in an external system. Use of this data ensures valid strings of COA so that generated transactions do not reject when a budget line is not found or use invalid COA codes.

There are times when manual intervention on the ACTOV page is required to update for Sub Activity, Program, and Phase. Additionally, after the deactivation of a COA code, this process deactivates the ACTOV records. If that code is re-activated, a user must activate the ACTOV records, as this job only deactivates.

The processing steps in this process are as follows:

1. Parameter Validation
2. Mode Processing - Activate
    - Clear the Budget Tracking table of selected records where the Object Categories were not tracked
    - Update with new budget lines and sub activity codes
    - Activate selected records of the Budget Tracking table
    - Clear the remainder of the Budget Tracking table
3. Mode Processing – Deactivate
    - Clear the COA Tracking table where the fiscal year is not the Activity Override Fiscal Year on Application Parameters
    - Deactivate each corresponding Activity Override record that matches an COA Tracking table record
    - Clear the COA Tracking table
4. Report Creation

The messaging from the process is as follows:

Process Steps	Messages
1. Parameter Validation	• Run Started • Parameters are listed • If a parameter is invalid, a job log message will specify

	what parameter failed.
2. Activate Mode	• ## number of records analyzed in ACTV_OVERRIDE_BUD_TRACK for activation • If an unexpected error is triggered while clearing the table, the log message will indicate an error has occurred. • Display log message with running total of records added to the Activity Override table. • Display log message with the total number of records analyzed for possible activation.
3. Deactivate Mode	• ## number of records analyzed in ACTV_OVERRIDE_COA_TRACK for deactivation • If an unexpected error is triggered while clearing the table, the log message will indicate an error has occurred. • Display log message with running total of records on the Activity Override table that had their Active checkbox unchecked. • Display log message with the total number of records analyzed for possible deactivation. • If an exception is triggered, display log message that indicates a COA error has occurred.
4. Report Creation	• Rendering report started • Rendering report completed

Restartability Information

The Activity Override Batch job supports restartability after any commit point.

When the Activity Batch Mode (ACTV_BATCH_MODE) is set to Activation, the job process will first filter out the records contained in the Activity Override Budget Tracking (ACTV_OVERRIDE_BUD_TRACK) table that will not be processed. Following the end of that process, a database commit is processed and the batch job can restart from this point. Afterwards, the remaining records are processed as Activated, then the records in ACTV_OVERRIDE_BUD_TRACK with a matching Tracking ID are removed. Finally, a commit is processed and completes the batch job. The same alternative process is followed for the Deactivation mode using the Activity Override COA Tracking (ACTV_OVERRIDE_COA_TRACK) table. Thus, if the job is terminated or otherwise ends before the final database commit, the restarted job will either have already filtered out the records not being processed or will have no records to process.

Major Input

- Activity Override Budget Tracking (ACTV_OVERRIDE_BUD_TRACK)
  - Receives records via the submission of an Expense Budget 51 (BGE51) transaction with new line entries added to level 3 of the budget.
- Activity Override COA Tracking (ACTV_OVERRIDE_COA_TRACK)
  - Receives records via the update of existing COA records for the following page codes: DEPT, UNIT, ACTV, SACTV, FUND, SFUND, APPR, and PHPRG

- Only if the Active check box is changed from checked to not checked, when no validation error is triggered, is a new record added to the table.
- Only if the Active check box is changed from not checked to checked, when no validation error is triggered, is an existing record removed from the table.

Application Parameters

Note: The manual method for adding records to the Activity Override page uses the following Application Parameters.

Parameter	Description	Default Value
Object Categories (ACTOVER_ALLOW_OCAT)	A required parameter defining the object categories for payroll costs tracked on the Activity Override page. Separate multiple values with a comma.	(No Default)
Cost Accounting Departments (ACTOVER_DEPT_PROGPH)	An optional parameter listing departments that require values for Program and Phase on the Activity Override page. Separate multiple values with a comma.	(No Default)
Activity Override Enabled (ACTOVER_ENABLED)	The parameter that turns this unique feature on/off with values of *true* and *false,* respectively. When enabled, updates to Fund, Sub Fund, Department, Unit, Activity, Sub Activity, and Program Phase.	(No Default)
Excluded Departments (ACTOVER_EXC_DEPT)	An optional parameter defining the individual department codes that are not part of the Activity Override feature. The process does not track Budget lines created on BQ51LV3 or department reference data listed here. Separate multiple values with a comma.	(No Default)
Excluded Funds (ACTVOVER_EXC_FUND)	An optional parameter defining the individual fund codes that are not part of the Activity Override feature. The process does not track Budget lines created on BQ51LV3 or fund reference data with a fund listed here. Separate multiple values with a comma.	(No Default)
FY to be used for Activity Override validation (ACTOVER_FY)	A required parameter listing the Fiscal Year used in the validation of Activity Override records. Please enter as YYYY.	(No Default)

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
Batch Mode (ACTV_BATCH_MOD	Required mode for processing: Activation (A), Deactivation (D), or Both (A,D)	A,D

E)		
Client Name (CLIENT_NM)	Optional name to appear in the title of reports produced.	(No Default)
Commit Block Size (COMMIT_BLOCK)	Required performance parameter to control the number of records committed in a save. If left blank, 1000 defaults.	1000
Select Block Size (SELECT_BLOCK)	Required performance parameter to control the number of records selected for a processing round. If left blank, 1000 defaults.	1000
Progression Counter (PROG_CTR_SIZE)	Program Counter Size Optional. Enter the interval in number of records the program should process for logging progression messages. If left blank, 100 defaults.	1000

Major Output

- Activity Override (ACTOV / ACTV_OVERRIDE)
- Activity Override Report

Job Return Code

This table shows the potential job return codes.

Return Code	Condition
Successful (1)	A selection of identified records for Activation or Deactivation are successfully processed.
Warning (4)	N/A
Non-Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Unexpected error detected while filtering out records prior to Activation or Deactivation • Unexpected error detected while processing records for Activation or Deactivation • Unexpected error detected while clearing records after Activation or Deactivation
Terminated (16)	The job ends with this return code when a user terminates the job.
System Failure (20)	The job ends with this return code when the job terminates from the database server, application server, or network issues.

Sort Criteria

- Activation:

- 1 – Join Tables: BFY / DEPT_CD
- 2 – Check Activity: SACTV_CD

- Deactivation: (1 – Initial Selection) DEPT_CD

Selection Criteria

Selection is different based on the mode: Activation or Deactivation

Activation

1. Select and delete ACTV_OVERRIDE_BUD_TRACK records based on the following parameters: ACTOVER_ALLOW_OCAT.
2. Join remaining records with BUD_STRU_51_LVL_3 records that meet the following criteria:
   - CURR_BUD_AM is greater than 0
   - CURR_BUD_AM is not equal to PEND_INCR_1
   - ACT_FL is true
   - BFY equals the ACTOVER_FY of APPCTRL
3. Use the remaining records to select their matching R_SACTV records containing the same DEPT and ACTV values with fiscal year equal to the ACTOVER_FY.
4. Add the record to ACTV_OVERRIDE only if that record does not already exist in the table.
5. Remove the matching record from ACTV_OVERRIDE_BUD_TRACK.

Deactivation

1. Select and delete ACTV_OVERRIDE_COA_TRACK records not equal to the ACTOVER_FY.
2. Change the ACT_FL value to false for all matching ACTV_OVERRIDE records that match with an ACTV_OVERRIDE_COA_TRACK record.
3. Remove the matching record from ACTV_OVERRIDE_COA_TRACK.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the identified records for Activation or Deactivation are processed.	N/A	N/A
Warning (4)	This Return Code is not used.	N/A	N/A
Non-Fatal Error (8)	This Return Code is not used.	N/A	N/A

Failed (12)	Job failed due to a Fatal condition.	In this step, the job can fail under the following four conditions: Entered parameters are invalid. Unexpected error detected while filtering out records prior to Activation or Deactivation Unexpected error detected while processing records for Activation or Deactivation Unexpected error detected while clearing records after Activation or Deactivation Check the job log for error details to diagnose.	When possible, restart; otherwise, submit a new instance of the job. Do not restart if another instance has already run.
Terminated (16)	Job ends with a return code of Terminated if a user terminates the job.	Investigate the reason for the termination before scheduling a new job.	N/A
System Failure (20)	Job ends with a return code of System Failure when the job terminates because of database server or network issues.	Investigate the reason for the System Failure before scheduling a new job.	N/A

2.1.7 Annual Close Chain

Chain Name	Annual Close Chain
Recommended Frequency	Annually, after all adjusting entries have been completed and all accounting periods in the closing year have been closed except 99 is the most common. If closing by Fund or Department, the frequency is still annually but in multiple runs.
Single Instance Required	Yes

Overview

The purpose of Annual Close Chain is to facilitate the movement of nominal and real account balances from an accounting fiscal year that is being closed to the subsequent accounting fiscal year. The nominal account balances (for example, cash expenditures, accrued expenditures, collected revenue, and billed revenue) are closed to fund balance, retained earnings or agency due to as specified for the associated fund. These closings into equity accounts are offset by two 'balancing' accounts, which summarize all nominal spending and nominal revenue account activity. Such a balance is necessary to keep the original nominal activity in the ledgers.

Real account balances are rolled forward from the accounting fiscal year that is being closed to the subsequent accounting fiscal year. Closing entries into the old year, use the same chart of accounts (COA) as recorded in that closed year, but use a special closing posting code so that real account activity will remain in the ledgers under the primary posting codes. New year activity is recorded using the posting code and balance sheet account originally used in the closed year. These account balances roll to the new accounting fiscal year using the level of detail found on the input ledger into annual close. If department is required for real account reporting, then the input ledger should summarize real accounts down to that level of detail. No more detail than Fiscal Year, Posting Code, Fund, Sub Fund, BSA, Sub BSA, and Department should exist in the input ledger.

Certain accounts used as offsets in the normal course of business are closed into a special equity account known as Net Assets. Such accounts are offsets for debt, pre-paid assets, and fixed asset expenditures. This new equity account contains many of the accounts that make up the difference between modified and full accrual accounting. If modified accounting reports are being created, then the Net Assets account is not included. However, for full accrual accounting, a fund's Net Assets account is combined with fund balance or retained earnings accounts to arrive at a complete picture of equity.

To accommodate the movement of account balances, two annual closing related fiscal periods have been created. Period 99 has been created for the accounting fiscal year that is being closed. Additionally, period 0 has been created for the subsequent accounting fiscal year into which account balances are being rolled. These fiscal periods have been reserved for the annual closing activity only to allow for the segregation of annual close transaction activity. It is the Allowed Accounting Periods for Transaction Code (AAPDC) page that controls what transaction codes can hit what periods based on a time restriction of accounting period for the transaction code on Transaction Control (DCTRL).

The Annual Closing Process facilitates the movement of account balances though the creation of one or more Journal Voucher Annual Close (JVAC) transactions. The process creates a minimum of two JVAC transactions for each unique combination of fund code and sub fund code, one for the closed year and one for the new year. Transaction header descriptions on these transactions specify the fund, sub fund and the year for easy identification. More than two transactions will be created for a combination when the Number of Accounting Lines control on the System Options – General Tab is exceeded. In such a case, a line is generated automatically

with the Balance Posting Code. On the final transaction for that combination and year, another balancing line is generated that offsets the first.

When reporting with trial balance reports before an actual close is desired, the first job in the chain provides the data necessary to see what the annual closing entries will be. An offline table, called the TEMP_FYDAD, is populated by that job with what would become the journal voucher lines if a closing were done. This report mode allows the entry of one or more funds as job parameters so that trial balance reporting can be done for a single fund or group of funds quickly without having to develop closing entries for all funds. If the job is run in report mode with just one fund, it is not possible to run it again in update mode. Annual closing is not permitted to be done one fund at a time. If the first job in the chain is run in report mode, it is advised that the remaining jobs in the chain be deactivated. Although there is no harm in not deactivating them, the second job will end in a failed status if it is not deactivated. This causes the remaining two jobs to go inactive.

This chain has the following jobs: (each of the jobs listed below is described in the subsequent sections):

- **Annual Close:** Reads the input ledger and updates temporary table with journal voucher lines. When run in other than Report mode, this step creates the journal voucher XML file.
- **Load JVAC:** Loads the XML file as draft transactions.
- **Submit JVAC:** Submits the draft transactions.
- **Closing AFY**: When all journal vouchers are submitted successfully, this step updates the respective closing indication(s).

Advanced Run Methods

Although a single run to close all accounts for a year is the most common, the process is capable of incremental closes. This run method comes with an implied condition of now allowing updates to a closed account from an incremental run unless as part of that update, annual close entries are simulated in APD 99 of the closing year and APD 0 of the new year. Whereas a single run updates the Closing Process Run indication for the closed year on the Fiscal Year page to stop further activity, when closing incrementally by department, the Fiscal Year by Department page is updated as closed, and when closing incrementally by Fund, the Fiscal Year by Fund page is updated as closed. These two closings are 'soft' closings as compared to the other 'hard close'. Be careful while restricting update access to these pages and granting an override level sufficient to get around the soft closes.

- To close by fund, use the Fund selection parameter with one or more values.
- To close by department, use the Department selection parameter with one or more values.
- When a multi-tenant environment where a set of funds is assigned to a department (no sharing funds), there is an added method of running where the Department selection parameter is used in conjunction with a COA Combination table parameter for the system to obtain the list of funds used with the department. This is essential to get balance sheet activity not recorded to department. Closing by department *requires* the use of the COA Combination table unless all ledger records contain department.

Preprocessing Instructions

- Check for units of ledger posting work that are marked as failed on Journal Log. These should be posted by running the Ledger Engine set to reprocess failed work.
- Check that there are no records in the Posting Line Catalog for transactions in the year to be closed that are Final or Historical Final that have a Journal Posting Status value of 2-

Ready to Post, or 1-Not Ready to Post. These transactions should be posted to the Accounting Journal before closing.

- Run the Journal Engine and Ledger Engines in normal modes to ensure all accounting activity is posted to the input ledger for annual close.
- Verify that all Posting Code values are assigned to the desired Posting Code Closing Classification.
- Verify that all Posting Code Closing Classifications use the desired Posting Codes and have the proper Closing Action.
- Ensure the Fund Balance, Retained Earnings, Agency Due, Net Assets, Master Closing Object, Master Closing Revenue, and Annual Close Offset on the Miscellaneous tab of Special Accounts.
- A100 to A105 should point to either the Master Closing Object or Master Closing Revenue, depending on the posting codes A100 - A105.
- A200 - A203 should point to the proper equity or liability accounts.
- Annual Close does not use the Default BSA field for A204, Annual Close Offset, but rather pulls directly from the Annual Closing Offset field on the Miscellaneous SPEC table. Be sure that the type of BSA in that field matches the Account Type setting of A204 found on the Posting Code table.
- Set the Number of Accounting Lines limit on the General tab of Systems Options to a value that is desired for Annual Closing Journal Voucher size limits that will work with system capabilities on Transaction Component Requirements.
- Ensure all Fund codes for the year to be closed have the correct Close Fund into Account value.
- Ensure that Automatic Transaction Numbering contains an entry for the year to be closed and the year balances are rolled into that uses a valid Department Code in both years and has a prefix of ACLS.
- Run the Close Accounting Period job to close all accounting periods in the year to be closed except 99.

Major Input

- Accounting Fiscal Year Detailed ledger (LDGR_FYDAD) or other input ledger
- Special Accounts – Miscellaneous (SPEC / R_MS_SPEC)
- System Options – General (SOPT / R_GEN_SOPT)
- Posting Code Closing Classification (PSCDCL / R_PSCD_CLOSE)
- Posting Code (PSCD / R_PSCD)
- Accounting Period and Fiscal Year (APD / R_APD & R_FY)
- Fiscal Year (FY / R_FY)
- Fund (FUND / R_FUND)
- Fiscal Year by Fund (FYFD / R_FY_FUND)
- Fiscal Year by Department (FYDEPT / R_FY_DEPT)

Major Output

- Temporary Accounting Fiscal Year Detailed ledger (TEMP_FYDAD) to be used by annual close process and trial annual close reports.
- JVAC transactions (if run update mode or the update and report mode)
- Exception reports
- Fiscal Year (FY / R_FY)
- Fund (FUND / R_FUND)
- Fiscal Year by Fund (FYFD / R_FY_FUND)
- Fiscal Year by Department (FYDEPT / R_FY_DEPT)

Chain Return Codes

When an active job step in the chain ends with a Return Code of anything other than successful, any remaining job steps have a Run Status of *Inactive.* Refer to documentation on individual job steps for possible Return Codes for those steps, as *Warning* and *Non-Fatal Error* are not always possible outcomes.

Problem Resolution

If any of the jobs in the chain failed due to application errors, restart the job after correcting the errors instead of rescheduling the job.

Annual Close Chain: Annual Close Job

Job Name	Annual Close
Recommended Frequency	N/A
Single Instance Required	N/A
Can be restarted?	No
Reports generated	Exception report

Overview

This job does the following tasks depending on the Mode:

- All three modes validate batch parameters.
- Report and the Report & Update modes validate several setup conditions on APD, SOPT, SPEC, and PSCD. Items found to be invalid are written out to the exception report. These two modes also select and summarize input ledger records into the temporary table as journal voucher lines. Both modes will clear what is present in the TEMP_FYDAD table.
- Report & Update as well as Update modes take information from the temporary table to create the journal voucher XML 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 Parameter validation completed

2. Configuration Validation	• Reports output folder mapped • HTML report mapping • PDF report mapping • If there are configuration issues: Stopped because of parameter errors. • Run ended (Report Mode only)
3. XML File Creation	• No messages in this step.

Input

- Accounting Fiscal Year Detailed ledger (LDGR_FYDAD) or other input ledger
- Special Accounts – Miscellaneous (SPEC / R_MS_SPEC)
- System Options – General (SOPT / R_GEN_SOPT)
- Posting Code Closing Classification (PSCDCL / R_PSCD_CLOSE)
- Posting Code (PSCD / R_PSCD)
- Accounting Period (APD / R_APD)
- Fiscal Year (FY / R_FY)
- Automatic Transaction Numbering (ADNT / AUTO_DOC_NO)

Parameters:

Parameter	Description	Default Values
Export Directory (AMSEXPORT)	The required file location for the XML file created.	$$AMSROOT$$/ExportImport
Log Directory (AMSLOGS)	The required log file location.	$$AMSROOT$$/Logs
Parameter File Directory (AMSPARM)	The required file location for the parameter files created in the first job that is passed to later jobs.	$$AMSROOT$$/Parms
Transaction Balancing Posting Code (BS Posting Code)	The required posting code used for journal voucher balancing of debits and credits when the number of lines for a fund/sub fund exceeds the Accounting Line Limit from System Options.	A204

Client Name (CLIENT_NM)	An optional name for the header of the report.	No default
COA Combination Table (COA_COM)	The optional database table name when closing by department to obtain the list of fund codes used independently by each department. CVIN_FUND_DEPT would be the most likely baseline table.	No default
Closing AFY	The required fiscal year to be closed as YYYY.	No default
Transaction Code (DOC_CD)	The required journal voucher transaction code used in the XML file.	JVAC
Transaction Prefix (DOC_PFX)	The required transaction prefix used to acquire the Automatic Transaction Numbering parameter. It must he ACLS.	ACLS
Transaction Department (Department)	The required department code used in the creation of each journal voucher.	No default
Document Record Date	An optional date, specified as mm/dd/yyyy, for the journal voucher header records instead of the application date that would default at the time of processing.	No default
XML File Name (EXP_FILE_NM)	The required file name used in the creation of the XML file.	JVACDocuments.xml
Selection Fund (Fund)	An optional parameter used to specify one or more fund codes, separated by commas, for reporting or closing.	No default
Load Parameter File Name (LOAD_PARM_FILE)	The required file name for the file created to direct the Load JVAC job step.	ACLoadParams.txt
Input Ledger (Ledger Name)	The required Data Object Name of input ledger.	LDGR_FYDAD
Opening AFY	The required fiscal year to be rolled into as YYYY.	No default
Run Mode	The required run mode of the job. Refer to the Overview section for details: (1) Report Only, (2) Update, or (3) Update and Report.	1
Selection Department (SEL_DEPT)	An optional parameter used to specify one or more department codes, separated by commas, for	No default

	reporting or closing. This parameter cannot be used in conjunction with Selection Fund.	
Selection Parameter File Name (SEL_PARM_FILE)	A required file name for the file created to direct the Closing AFY job step to know which Fiscal Year Fund or Fiscal Year Department records to close.	LIMITEDAC.txt
Submit Parameter File Name (SUBMIT_PARM_FILE)	A required file name for the file created to direct the Submit JVAC job step.	ACSubmitParams.txt
Transaction Unit (Unit)	Optional field for a Unit code to be used as the Transaction Unit value on JVAC transactions to facilitate security at a level below department.	No default

Major Output

- TEMP_FYDAD
- Parameter Files (3)
- XML File

Sort Criteria

Fund, Sub Fund, Posting Code

Selection Criteria

Records are selected from LDGR_FYDAD based on the following parameters entered by the user: Fund (if entered, all Fund if left blank), Department (if entered), and Closing Fiscal Year and the amount on LDGR_FYDAD <> 0.

Problem Resolution

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

Return Code	Condition
Successful (1)	The job ends as successful when all parameters and configurations are valid and there are matching ledger records to selection criteria.
Warning (4)	The job does not end with this return code.
Non-Fatal Error (8)	The job does not end with this return code.
Failed (12)	The job fails under the following conditions: • Parameters are invalid. • Required configurations are not in place. • Run time exceptions for unexpected situations. When this job fails, subsequent jobs in the chain are set to *Inactive*.

Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to *Inactive*.
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to *Inactive*.

If an exception report is produced, run the process again in Report Only mode after fixing the configuration errors as the progression through the verifications stops when one is encountered. Therefore, a second run may reveal a new issue.

Annual Close Chain: Load JVAC job

Job Name	Load JVAC
Recommended Frequency	N/A
Single Instance Required	N/A
Can be restarted?	No
Reports generated	None

Overview

This job executes a system maintenance utility to import the XML file.

Input

- JVACDocuments.xml
- ACLoadParams.txt

Parameters:

Parameter	Description	Default Values
Load Parameter File Name (LOAD_PARM_FILE)	The required file name created in the first job step with instructions on loading the XML file.	ACLoadParams.txt

Output

- Transaction Catalogs: (DOC_HDR, JV_DOC_HDR, JV_DOC_LNGRP, DOC_ACTG, and JV_DOC_ACTG)

Sort Criteria

None

Selection Criteria

None

Problem Resolution

Return Code	Condition
Successful (1)	Parameter is valid. XML file found and all records imported successfully.
Warning (4)	The job does not end with this return code.
Non-Fatal Error (8)	A non-fatal error is issued when not all transactions are loaded. Typically, the result of the Accounting Line Limit on System Options being less than the MAX_LINE_LIMIT for JV_DOC_ACTG on Transaction Component Requirements.
Failed (12)	The job fails under the following conditions: • Parameters are invalid. • Run time exceptions for unexpected situations. When this job fails, subsequent jobs in the chain are set to *Inactive*.
Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to Inactive.
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to *Inactive*.

Annual Close Chain: Submit JVAC Job

Job Name	Submit JVAC
Recommended Frequency	N/A
Single Instance Required	N/A
Can be restarted?	No
Reports generated	None

Overview

This job executes a system maintenance utility to submit the journal vouchers loaded for the annual close.

Input

- JVACDocuments.xml
- ACLoadParams.txt

Parameters:

Parameter	Description	Default Values
Submit Parameter File Name (SUBMIT_PARM_FILE)	The required file name created in the first job step with instructions on loading the XML file.	ACSubmitParams.txt

Output

- Transaction Catalogs: (DOC_HDR, JV_DOC_HDR, JV_DOC_LNGRP, DOC_ACTG, and JV_DOC_ACTG)

Sort Criteria

None

Selection Criteria

None

Problem Resolution

Return Code	Condition
Successful (1)	Parameter is valid. XML file found and all records imported. successfully.
Warning (4)	The job does not end with this return code.
Non-Fatal Error (8)	A non-fatal error is issued when all transactions are not loaded. Typically, the result of the Accounting Line Limit on System Options being less than the MAX_LINE_LIMIT for JV_DOC_ACTG on Transaction Component Requirements.
Failed (12)	The job fails under the following conditions: • Parameters are invalid. • Run time exceptions for unexpected situations. When this job fails, subsequent jobs in the chain are set to *Inactive*.
Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to *Inactive*.
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to *Inactive*.

Annual Close Chain: Closing AFY Job

Job Name	Closing AFY
Recommended Frequency	N/A
Single Instance Required	N/A
Can be restarted?	No
Reports generated	None

Overview

This job verifies that all journal vouchers created have submitted after parameter validation is done. If that is the case, it proceeds with closings. If the Selection Fund and Selection Department parameters were not used in the 1st job, this last job should hard close (Closing Process Run = *true)* for the Closing AFY on the Fiscal Year page. The soft close (Closed = *true)* is done on the Accounting Period page for APD 99 of the Closing AFT and APD 0 of the Opening AFY. If the Selection Fund parameter was used, the Closed field is set to *true* for the fund(s) selected on Fiscal Year Fund. If the Selection Department parameter was used, the Closed field is set to *true* for the department(s) selected on the Fiscal Year Department page. When selecting by fund or department, when all have been closed, the Final Close Parameter should be set to 1 so this job step can close on the Fiscal Year and Accounting Period pages.

In the event, not all journal vouchers submitted successfully, errors on those should be addressed and they should be submitted to final. Then another instance of the chain submitted with only this job step run.

Input

- LIMITEDAC.txt
- ACSubmitParams.txt

Parameters:

This job step is unusual in that it requires some parameter updates and does not pull them all from the first job step.

Parameter	Description	Default Values
Parameter File Directory (AMSPARM)	The required file location for the parameter files created in the first job that are passed down to this job step.	$$AMSROOT$$/Parms
Client Name (CLIENT_NM)	An optional name for the header of the report.	No default
Closing AFY	The required fiscal year to be closed as YYYY.	No default

Final Close Parameter (FIN_CLOSE_PARM)	A required parameter indicating whether or not FY and APD should be updated only (normal run for all activity), or whether only FYFD or FYDEPT is to be updated (incremental run that is not the final incremental run), or FY and APD is to be updated and either FYFD or FYDEPT is also updated (final incremental run). 1 ensures FY and APD are updated.	1
Opening AFY	The required fiscal year to be rolled into as YYYY.	No default
Selection Parameter File (SEL_PARM_FILE)	A required file name for the file created to direct the Closing AFY job step to know which Fiscal Year Fund or Fiscal Year Department records to close.	LIMITEDAC.txt
Submit Parameter File Name (SUBMIT_PARM_FILE)	The required file name created in the first job step with instructions on loading the XML file.	ACSubmitParams.txt

Output

- Fiscal Year (FY / R_FY)
- Accounting Period (APD / R_APD)
- Fiscal Year Fund (FYFD / R_FY_FUND)
- Fiscal Year Department (FYDETP / R_FY_DEPT)

Sort Criteria

None

Selection Criteria

None

Problem Resolution

Return Code	Condition
Successful (1)	Parameters are valid. All journal vouchers submitted successfully.
Warning (4)	The job does not end with this return code.
Non-Fatal Error (8)	Not all journal vouchers submitted to final.
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. When this job is terminated, subsequent jobs in the chain are set to Inactive.
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to Inactive.

2.1.8 Annual Financial Reporting Data Load

Chain or Job Name	Annual Financial Reporting Data Load
Recommended Frequency	The Cost Allocation Process can be run on an annual basis.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

All Public Sector Organizations, as part of their fiduciary and legal responsibilities, must submit an Annual Comprehensive Financial Report, commonly referred to as the ACFR. The objective of the ACFR is to give constituents of the organization (citizens, businesses, other jurisdictions, bond holders, financial advisors, potential investors, and senior management) a broader view and understanding of the organization's financial operations. The reported financial statements in the ACFR represent the financial position and results of operations of the various funds and component units of the organization. The financial statements included in the ACFR must conform to the generally accepted accounting principles as promulgated by the Governmental Accounting Standards Board (GASB). Responsibility for the accuracy of the data and the completeness and fairness of the presentation, including all disclosures, rests with the organization's management.

The Published ACFR is a very comprehensive transaction that presents the financial position and results of operations through the:

- Financial Statements
- Disclosure (notes)
- Required Supplementary Information

The ACFR module includes the batch job process which summarizes the source data based on the rules established into the ACFR Summary table and the Crosswalk table.

The CGI Advantage ACFR Architecture includes:

- Source Data
- Configuration Rules
- ACFR Batch Process

Steps for Running this Job:

This process is one that can be run at any time without restricting user activity. The chances are very low that setup for a report will change during a run. If that occurs, then run the process again.

- If using a ledger as an input source, the Ledger Engine should not be run during this process as one cannot assume the latest version of a ledger record was selected, which will complicate report verification.

- If using a journal as an input source and all transaction activity to the reported year cannot be restricted during the run, it would be best to run this process when there is no user activity. This situation can be avoided by ensuring a soft close with very restricted override capability in the prior reporting periods to just those making ACFR adjustments.

Processing Logic

The process performs the following steps:

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 will be displayed in the log. It will be followed by the message “Batch Parameter validation failed.” • Batch Parameter validation completed.
2. Selection of Records	• Records for each ACFR statement will be selected from Advantage Financial based upon the setup in the ACFR configuration tables. • Based on the configuration done for each of the financial statements, the Statement Summary table and Statement Crosswalk table are populated. • The Statement Summary table contains objects and measures that are necessary to create an ACFR statement. Statement Row and Statement Column and their respective details are utilized to generate a matrix. • The Statement Crosswalk table contains objects and measures required for the reconciliation between the Statement Summary table and its application data source.

Restartability Information

The job cannot be re-started.

Major Input

1. Source Data

The Annual Financial Reporting Data Load batch job can utilize multiple sources of data and is configurable per installation via the ACFR configuration pages. The Day 0 delivery of the ACFR configuration pages identifies the Advantage Financial application as the primary source of data. The following Advantage Financial tables are included as part of Day 0:

- Advantage Financial Accounting Journal (JRNL_ACTG)
- Advantage Financial Budget Journal (JRNL_BUD)
- Advantage Financial System Assurance Ledger (LDGR_SA_BUD)
- Advantage Financial Statement External Data (STMT_EXT_DATA)

2. Configuration Rules

   The ACFR module includes configuration pages that help define the statement, including rows and columns, which need to be prepared. It allows establishing rules at the individual row and column level. These rules are read by the Annual Financial Reporting Data Load batch job to summarize the source data into the ACFR Summary table. This batch job can be configured to populate the Crosswalk table.

The following pages need to be configured, at a minimum, for generating the statement:

- Statement Definition (STMTDEF)
- Statement Rows (STMTROW)
- Statement Columns (STMTCOL)
- Statement Cell (STMTCELL)
- Statement Rules & Conditions (STMTRULE)

Other configurable pages include:

- Statement Condition Group (STMTCGRP)
- Statement External Data (STMTEXT)
- Statement Crosswalk Definition (STMTXWLK)

Batch Parameters

Parameter	Description	Default Value
REFRESH_MODE	(Required) 1 denotes Full Refresh 2 denotes On Demand Refresh More than one record can also be selected with these two modes. They also decide which ACFR statement needs to be executed and loaded to the STMT_SMRY table. "On Demand" is to run specific ACFR statements. The statements	No Default

Parameter	Description	Default Value
	to be run are marked using the "Run Now" flag check box on the Statement Definition page. "Full Refresh" ignores the flag and picks up all active statements.	
MAX_BATCH_SIZE	(Required) Sets the maximum size of the batch that is sent to a single running instance.	4
STMT_THREAD_LIMIT	(Required) The maximum number of statements to process simultaneously. If left blank, the default value is considered. Based on this value the thread pool is created for refreshing statements.	4
CELL_THREAD_LIMIT	(Required) The maximum number of cells to process simultaneously. If left blank, the default value is considered. Based on this value the thread pool is created for refreshing cells.	8
SQL_AGGREGATION_VALUE	(Required) This flag indicates whether to use Aggregation in the rules' SQL statements. If left blank, the default value is used.	True
USE_PARALLEL	Degree of parallelism to be used while running ACFR Statement queries. Default is blank.	No Default

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

Major Output

- Statement Summary table – contains objects and measures necessary to create an ACFR statement. Statement Row and Statement Column and their respective details are utilized to generate a matrix.
- Statement Crosswalk table – contains objects and measures required for reconciliation between the Statement Summary table and its application data source.

Job Return code

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

Return Code	Condition

Successful (1)	All selected records are processed successfully.
Warning (4)	This job does not use this return code.
Non-Fatal Error (8)	This job does not use this return code.
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

- None

Selection Criteria

Based on the source data that is present in the tables, JRNL_ACTG, JRNL_BUD, LDGR_SA_BUD, STMT_EXT_DATA, and the configuration settings that are done in the Configuration tables, the statements to be processed are selected.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters are validated successfully.	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	The job failed due to fatal conditions: • Encounters any runtime exceptions • Invalid parameters	If the job failed due to parameter edits, correct the parameter and run a new job. If the job failed because of runtime exceptions, investigate the exception reported by the process, resolve the error and run a new job.	Only a new job should be scheduled.
Terminated (16)	The job is terminated manually by the user.	Reason for the System Failure needs to be	Only a new job should be

		investigated. Once the issue is resolved, a new job can be scheduled.	scheduled.
System Failure (20)	The job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Once the issue is resolved, a new job can be scheduled.	Only a new job should be scheduled.

2.1.9 Automated Accrual Process

Chain or Job Name	Automated Accrual Process
Recommended Frequency	Nightly starting at the beginning of a new fiscal year and ending when payments for prior year goods and services stops.
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	Yes - Exception Report

Overview

The Automated Accrual Process chain job selects records from the Accounting Journal to generate Automatic Accrual (ACCA) transactions. The records selected are those accrued expenditure posting codes from the Payment Request (PR) and Accounting Based Spending (ABS) transaction types. Further selection will be based on the current fiscal year (as determined from the application date) and where the Service From Date is before the start of the current fiscal year.

With this process, users are not required to associate any or all of the expenditure to the prior or current budget (BFY) and accounting fiscal year (FY). Users only have to enter the payment request in the current fiscal year and the application will associate all or a portion of the expenditure into the prior FY/BFY based on the service dates. This selection will be for new, modifications, and cancellations of those payment requests up through the end of the accounts payable time frame (a batch parameter signals the end). After that the chain is either not run or run to select only decrease modifications and cancellations. If not run, then accruals have to be done manually with a journal voucher or manual accrual (ACC) transaction. After the accounts payable period ends, there should only be new payment requests for the current year, so the process has the ability if run after the end of that period to only process modifications and cancellations to payment requests that were previously accrued.

As this chain is taking current year activity and accruing all or a portion into the prior FY/BFY, there can be the need to crosswalk COA as part of that accrual. This process reads the Chart of Account Crosswalk (COAX) page for Process ID of 'ACC' to find any crosswalk rules. Unlike the Open Activity Roll that goes from the prior year to current year, the Accrual process goes from current year to prior year. For this reason, the COAX page has to be completed with the current year in the Source Fiscal Year field and the prior year in the Target Fiscal Year.

The chain includes three job steps that perform the following functions:

1. Automated Accrual Preprocess – This job performs parameter validation, record selection, XML file creation, and the creation of files used by later job steps.
2. Accrual Upload Job – This is a multi process load and submit job that would split the XML file as created by the preprocess step to have the transactions load and submit through multiple SMU while creating an exception file for the transactions that failed to submit successfully.
3. Accrual Exception Job – This job step takes the exception file(s) created by the SMU submit job(s) and compiles them into a single exception report.

Major Input

- Journal Log (JRNL_LOG)

- Accounting Journal (JRNL_ACTG)
- Automated Accrual / Accrual Clearing Exclusions (R_ACRL_CLR_EXCL)
- Chart of Account Crosswalk (R_COA_CROSSWALK)

Major Output

- ACCA Transaction XML
- Exception Report

Chain Job Return code

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

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

Problem Resolution

If any of the jobs in the chain failed due to application errors it is advisable to restart the job after correcting the errors instead of rescheduling a new job. Restarting the job will reduce the processing time since the job will resume from where it has last committed and select only the unprocessed records. Please refer to the individual jobs for details regarding the specific job processes and problem resolution.

Automated Accrual PreProcess Chain: Automated Accrual Preprocess Job

Job Name	Automated Accrual Preprocess
Recommended Frequency	See chain
Single Instance Required	Yes
Can be restarted?	No
Reports Generated	No

Overview

This job step consists of following basic steps:

1. **Parameter Validation:** If the parameter validation is successful, the job will go to the next step. Otherwise the job will end with a return code of 'Failed'.
2. **Selection of Records:** Records from JRNL_ACTG are selected based batch parameters, starting at the JRNL_LOG record recorded from any previous run.
3. **Processing of Records:** Selected JRNL_ACTG records are processed and ACCA transactions are created with a Current Year and Previous Year Accounting line.
4. **Generate XML:** An XML file is generated with ACCA transactions.
5. **Files for Later Job Steps:** Files are created for latter load and submit steps.

Process Steps	Messages
4. Parameter Validation	• Run Started • Each parameter is listed with value • Any parameter errors are listed
5. Selection of Records	• No journal records found matching accrual criteria (if no journal records selected) • If records are selected, no messages are issued for this step but for the next
6. Processing of Records	• Journal records processed: ## (where ## is the count of journal records found matching accrual criteria)
7. Generate XML	• No messages issued for step
8. Files for Later Job Steps	• No messages issued for step • Run Ended

Restartability Information

N/A

Major Input

- Accounting Journal (JRNL_ACTG)
- Automated Accrual / Accrual Clearing Exclusions (R_ACRL_CLR_EXCL)
- COA Crosswalk (R_COA_CROSSWALK)
- Application Parameters (IN_APP_CTRL) for the ACCRUAL_PY_APD parameter

Batch Parameters

Listed below are the delivered Default Values. Each implementation is encouraged to set any delivered (blank) to a constant value on the BATSETUP page. Also, there are other parameters that are updated annually.

Parameter	Description	Default Value
AMSEXPORT	(Required) AMS Export Import Directory	$$AMSROOT$$/ExportImport
AMSLOGS	(Required) The location of the log file.	$$AMSLOGS$$
AMSPARM	(Required) Parameter Location for Upload Job	$$AMSPARM$$
AP_END_DT	(Required) The Accounts Payable Period End Date that will be used to signify the selection of only modifications and cancellations of previously accrued lines. Parameter must be changed yearly.	(blank)
CHK_PT_SIZE	(Required) Check Point Size is a performance parameter that controls the check-pointing for job restartability.	5000
COMMIT_SIZE	(Optional) Commit Block Size is a performance parameter that controls the number of records committed to the XML file.	1000
DOC_CD	(Required) Accrual Transaction Code	ACCA
DOC_DEPT	(Optional) Populate with a single department to be used for all generated transactions. Leave blank to use the transaction department from the selected journal record. If left blank, ensure the Transaction Unit Code Required for the generated transaction is not checked on DCTRL if that same flag is not checked for all referenced transaction codes. If at least one referenced transaction code is unchecked, it must also be unchecked for the generated transaction code.	(blank)
DOC_UNIT	(Optional) Accrual Transaction Unit Code used for security purposes.	(blank)
PREFIX	(Optional) Accrual Transaction ID Prefix used for ADNT record retrieval.	(blank)
PROG_CTR_SIZE	(Optional) The Progression Counter defines the number of records processed that will trigger a progression message in the job log.	100
PSTNG_CD	(Required) Posting Codes that are Accrued Expenditures for selection.	D011
SEL_BLK_SIZE	(Required) Select Block Size is a performance parameter that gives the number of journal records to be selected at once.	1000

SUBMIT_FILE_NM	(Required) File Name for Submit job with ACCA Transaction Information	ACCASubmit.txt
SEL_DT_APD	(Required) Selection Date used for modifications and cancellations after the Accounts Payable Period (1 = Transaction Record Date, 2 = Run Time Date). Using Run Time Date does not allow for users to back date or forward date a transaction for a desired accrual result.	1
AFTR_APD_PROC	(Required) After Accounts Payable Processing option for exceptions (1 = Normal processing for records not matching Fund/Debt lists, 2 = Non-matching records are omitted from processing).	1
SP_DEPT_APD_LST	(Optional) Special Departments list for After Accounts Payable Period processing.	No default
SP_DEPT_APD_TYP	(Required) Include or Exclude the Special Departments list (1 = Include, 2 = Exclude).	1
SP_FUND_APD_LST	(Optional) Special Funds list for After Accounts Payable Period processing.	No default
SP_FUND_APD_TYP	(Required) Include or Exclude the Special Funds list (1 = Include, 2 = Exclude).	1

Major Output

- AutoAccruals.xml (XML file of ACCA transactions)
- ACCASubmit.txt (text file containing the transaction information for submit)

Job Return Codes

The following table shows the potential job return codes for the Automated Accrual Preprocess job.

Return Code	Condition
Successful (1)	Parameters were valid and at least 1 record was selected and processed successfully.
Warning (4)	No eligible records found.
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 • Run time exceptions for unexpected situations. 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 Sequence

- None

Selection Criteria

- All records on the Accounting Journal (JRNL_ACTG) having Transaction Type (DOC_TYP) as either ABS or PR, Line Function Code (LN_FUNC_CD) as standard (1), a posting code equal to PSTNG_CD parameter, and a BFY equal to 9999 or the current BFY will be selected.
- The records having the Run Time Date Stamp (RUN_TMDT) on JRNL_ACTG greater than the Run Time (PROC_RUN_DT)" on Journal Log (JRNL_LOG) for the Process ID (PROC_ID) of ACCR will be selected.
- JRNL_ACTG records with a Service From Date (SVC_FRM_DT) less than or equal to the last date of the prior FY with a BFY of either the current BFY or 9999 will be selected.
- Additionally, modification or cancellation transactions will be selected from JRNL_ACTG whose Transaction Record Date (DOC_REC_DT) or Run Time Date Stamp (RUN_TMDT) is greater than the input parameter Accounts Payable Period End Date (AP_END_DT).
    - If the Selection Date after Accounts Payable Period (SEL_DT_APD) parameter is set to "1", then the AP_END_DT will be compared to DOC_REC_DT. If the SEL_DT_APD parameter value is set to "2", then the AP_END_DT will be compared with RUN_TMDT.
    - If either or both of the Special Departments After Accounting Payable Period List (SP_DEPT_APD_LST) and Special Funds After Accounting Payable Period List (SP_FUND_APD_LST) parameters are populated and the AFTR_APD_PROC parameter is set to "1", then journal records that match either or both will be processed with no requirement for previously being accrued and increases (debits) will also be processed. For all other Fund and Department records, existing special After Accounts Payable Period End Date processing will continue to process only decreases (DRCR_IND must be "C" for the selected JRNL_ACTG records) for previously accrued payment requests. Note: "previously accrued" means that 'PY Accrual/Clearing %' field PY_ACRL_CLR_PC) on ABS / PR Accounting lines is greater than zero for the journal records being processed.
    - If either or both of the Special Departments After Accounting Payable Period List (SP_DEPT_APD_LST) and Special Funds After Accounting Payable Period List (SP_FUND_APD_LST) parameters are populated and the AFTR_APD_PROC parameter is set to "2", then journal records that match either or both will be processed with no requirement for previously being accrued and decreases (credits) will only be processed. All other records will not be processed. If the

SP_DEPT_APD_LST/ SP_FUND_APD_LST parameters are left empty, then all values will be excluded or included.

- The Automated Accrual / Accrual Clearing Exclusions (R_ACRL_CLR_EXCL) table will contain the list of records with various combinations of Transaction Code, Event Category, Event Type, Posting Pair, Fund, Department, Appropriation and Object. Any JRNL_ACTG records matching an exclusion record will be excluded from selection.

Problem Resolution

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

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	This step doesn't issue this return code.	N/A	N/A
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	Required Parameters are not entered **Sample Message:** The DOC_CD Parameter cannot be empty	Correct parameter error in a subsequent run of the chain.	N/A
	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before another instance of the chain is started.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the termination needs to be investigated before another instance of the chain is started.	N/A

Step 2: Selection of records

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Records selected are processed successfully	N/A	N/A
Warning (4)	No Accounting Journal records selected for processing	Verify selection criteria are correct and if not, then submit another instance. This may be a valid outcome if selection criteria were correct.	If selection criteria were wrong the JRNL_LOG progress tracking record will have to be reset if incremented so selection can read that range of journal records again.
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	In this step, the job can fail under the following conditions. 1) Encounters any runtime exceptions	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Step 3: Processing of records

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All records are successfully processed	N/A	N/A
Warning (4)	This step doesn't issue this return code.	N/A	N/A
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A

Failed (12)	In this step, the job can fail under the following condition. 1) Encounters any runtime exceptions	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Step 4: Generate XML

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	XML file was successfully generated	N/A	N/A
Warning (4)	This step doesn't issue this return code.	N/A	N/A
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	In this step, the job can fail under the following conditions. 1) Issue creating XML file, for example, Import/Export file location not found. 2) Encounters any runtime exceptions	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the failure needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Step 5: Files for Later Job Steps

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Files were successfully generated	N/A	N/A
Warning (4)	This step doesn't issue this return code.	N/A	N/A
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	In this step, the job can fail under the following conditions. 1) Issue creating parameter files, for example, parameter file location not found. 2) Encounters any runtime exceptions	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the failure needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Automated Accrual Process Chain: Accrual Upload Job

Job Name	Accrual Upload
Recommended Frequency	See chain
Single Instance Required	Yes
Can the job be restarted?	Yes
Reports generated	No

Overview

After the generation of the ACCA transaction xml file, the next step of the Automated Accrual Process chain is the Accrual Upload. This step uses the MultiProcessImport job with parameter as ACCA Transaction XML file name.

This MultiProcessImport job will internally spawn multiple SysManUtil jobs by splitting the ACCA Transaction XML file into the parts as specified by the THREAD_COUNT parameter. The ACCA

transactions are selected from the ACCA Transaction XML file and loaded / submitted into the application by using the System Maintenance Utility (SMU).

Major Input

- ACCA Transaction.XML
- ACCASubmit.txt (text file containing the transaction information for submit)

Batch Parameters

Parameter	Description	Default Value
BLOCK_SIZE	(Required) Number of records in each split segment of the input file.	100
COMMIT_BLOCK_SIZE	(Required) Number of records to commit at a time.	100
FILE_INPUT_DIR	(Required) The location of the source files of records.	$$AMSROOT$$/ExportImport
FILE_LIST	(Required) Comma separated list of files to be uploaded with multi-threaded processing.	AutoAccruals.xml
FILE_OUTPUT_DIR	(Required) Output location for the file segments.	$$AMSROOT$$/ExportImport
FILE_PREFIX	(Required) Prefix used on the filenames for the output file segments.	ACCA
I_SMU_APPLY_OVERRIDES	(Required) If overrides are to be applied to transactions as they are loaded, this parameter should be true. A setting of false will result in transactions rejecting for override errors.	True
I_SMU_BYPS_ADNT_FL	(Required) Bypass ADNT Flag	true
I_SMU_DOC_STA_CD	(Required) Transaction Status For Loaded Transactions 1-Held, 2-Ready	2
I_SMU_OVERRIDE_LVL	(Optional) When an override level should be applied other than that of the user id submitting the batch job, that level should be entered in this parameter field. If left blank the override level of the user id submitting the job will be used if the Apply Overrides parameter is true.	(blank)
LOG_STATUS_INTERVAL	(Required) Logging frequency (in seconds) for controller thread	300

Parameter	Description	Default Value
	reporting status of child threads to the system log.	
MODE	(Required) Mode of operation. (1=Import, 2=Import and Submit, 3=Import and Other Action)	2
SLEEP_INTERVAL	(Required) Polling frequency (in seconds) for internal controller thread for checking child processes.	5
SMU_CTLG_ID	(Required) Catalog id of the System Maintenance Utility job which is spawned as the child process.	3
SMU_EXCEP_REP_FILE_NM	(Required) Exception File Name	ExcepACCASubmit.txt
SMU_EXCEP_REP_IND	(Required) Exception Detailing (1-Detailed, 2-Failed Transactions, 3- Processed Transactions, 4-Failed Transaction Lines, 5-Transaction Status)	4
STAGGER_TIME	(Required) The lag time, in seconds, between the spawning of each child process.	1
THREAD_COUNT	(Required) Parameter defines the number of SMU threads to use for processing.	8

Please refer to the SMU Transaction Upload / Submit Job run sheet in the *CGI Advantage Financial – Utilities Run Sheet Guide* for the full list of SMU Transaction upload / submit batch parameters.

Major Output

- ACCA Transactions in draft version

Job Return Code

The following table shows the potential job return codes for the Inferences & Validations job:

Return Code	Condition
Successful (1)	All of the records are loaded into the Transaction Catalog successfully or the input file is empty. All of the transactions loaded were submitted successfully.
Warning (4)	This return code will be issued when some of the records failed to load where all other records were loaded successfully.
Non-Fatal Error (8)	None of the transactions were loaded into the Transaction Catalog.

Failed (12)	• Parameters are invalid • When the input file is not found in the specified directory • Restart failed because another instance of the Accrual Process chain has already been run successfully • Runtime exceptions encountered for any unexpected situations When the job ends with a return code of Failed, subsequent jobs in the chain will be set to Inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain will be set to Inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to Inactive.

Sort Sequence

N/A

Selection Criteria

N/A

Problem resolution

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All transactions were loaded and submitted	N/A	N/A
Warning (4)	At least 1 transaction didn't load	The reason for the failure needs to be investigated before restarting the job. The transaction(s) that didn't load are in the Import/Export Error directory.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.
Non-Fatal Error (8)	No transactions were able to load	The reason for the failure needs to be investigated before restarting the job.	If another instance of the chain has run, then the XML file has to be

			loaded with a separate SMU instance.
Failed (12)	In this step, the job can fail under the following conditions. 1) Issues in spawning jobs 2) Files not found 3) Runtime exceptions 4) Parameter Edits	The reason for the failure needs to be investigated before restarting the job.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.
Terminated (16)	Job is terminated manually by the user.	The reason for the failure needs to be investigated before re-starting the job.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.
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 re-starting the job.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.

Automated Accrual Process Chain: Accrual Exception Job

Job Name	Accrual Exception
Recommended Frequency	See chain
Single Instance Required	Yes
Can be restarted?	No
Report Generated	Yes

Overview

This batch job will generate an exception report for the failed ACCA transactions. The report will be generated by showing accrual transactions, the referenced transaction and all errors issued.

Major Input

- ExcepACCASubmit.txt
- ACCA Transactions

Batch Parameters

Parameter	Description	Default Value
EXCEP_PARM_FILE_NM	(Required) Exception Report File Name that got generated by SMU	$$AMSEXPORT$$/ExcepACCA Submit.txt
RUN_TYP	(Required) Run Type (1 = Accrual, 2 = Accrual Clearing)	1
CLIENT_NM	(Optional) Client Name for the report header	N/A

Major Output

Exception Report

Batch Return codes

The following table shows the potential job return codes for the individual Accrual Exception job.

Return Code	Condition
Successful (1)	Exception report was not generated. All the ACCA transactions were submitted successfully.
Warning (4)	Exception report was created with failed ACCA transactions
Non-Fatal Error (8)	Not Applicable for this job.
Failed (12)	Input SMU log file was not found
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. When this job ends with a return code of System Failure.

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions

Successful (1)	All transactions are final	N/A	N/A
Warning (4)	One or more ACCA transactions rejected	Address the errors and submit the transactions either manually or with an SMU.	N/A
Non-Fatal Error (8)	The job does not end with this return code	N/A	N/A
Failed (12)	In this step, the job can fail under the following conditions. 1) Exception file not found 2) Runtime exceptions 3) Parameter Edits	The reason for the failure needs to be investigated before restarting the job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the failure needs to be investigated before re-starting 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 before re-starting the job.	N/A

2.1.10 Automated Accrual Clearing Process

Chain or Job Name	Automated Accrual Clearing Process Chain Job
Recommended Frequency	Nightly after the start of a new fiscal year and then less frequently after the bulk of disbursements for prior year goods and services
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	Yes. Missing Automated Accrual Transactions and Exception Report.

Overview

The Automated Accrual Clearing chain job selects records from the Accounting Journal to generate Automatic Accrual Clearing (ACLA) transactions. The records selected are those cash expenditure (and accrued expenditures in the case where retainage is left as accrued) posting codes from the Automatic Disbursement (AD), Manual Disbursement (MD), Internal Exchange (IET), and Internal Transaction Agreement (ITA) transaction types. Further selection will be based on the Service From Date being before the start of the current fiscal year. Selection is not Budget Fiscal Year (BFY) dependent as all BFY values including 9999 are selected. This expanded selection beyond what the Automatic Accrual chain uses will ensure disbursements of long held payment requests are cleared.

This chain is the second of two that allows users to enter only service dates on the purchase order and payment request with the system correctly accruing and clearing expenditures. The ACLA generated from this second chain will reverse accrued expenditures in the prior BFY, post cash expenditures in the prior BFY, reverse accrued expenditures in the current BFY, and reduce cash expenditures in the current BFY. All four parts are recorded in the current Fiscal Year (FY).

Automated and Manual Disbursements (AD/EFT/MD) transactions are the main input into the clearing process. However, the chain has an option to select Manual Disbursement (MD) transactions without referenced payment request, Internal Payment Request (PRMI/PRCI), Internal Exchange Transaction (IET) and Internal Transaction Agreement (ITA) transactions that include prior year service dates and should logically be considered for prior year accrual clearing.

As this chain is taking current year activity and accruing all or a portion into the prior FY/BFY, there can be the need to crosswalk COA as part of that accrual. This process reads the Chart of Account Crosswalk (COAX) page for Process ID of 'ACC' to find any crosswalk rules. Unlike the Open Activity Roll that goes from the prior year to current year, the Accrual process goes from current year to prior year. For this reason, the COAX page has to be completed with the current year in the Source Fiscal Year field and the prior year in the Target Fiscal Year.

Within the nightly cycle, the Automated Accrual Clearing process will normally be run at the end of normal processing and before system assurance jobs. The job definitely needs to be run after disbursement processing and journalization is complete.

The chain includes three job steps that perform the following functions:

1. Automated Accrual Clearing Preprocess – This job performs parameter validation, record selection, XML file creation, and the creation of files used by later job steps.

2. Accrual Clr Load – This is a multi process load and submit job that would split the XML file as created by the preprocess step to have the transactions load and submit through multiple SMU while creating an exception file for the transactions that failed to submit successfully.
3. Accrual Clr Exception – This job step takes the exception file(s) created by the SMU submit job(s) and compiles them into a single exception report.

Major Input

- Journal Log (JRNL_LOG)
- Journal Accounting (JRNL_ACTG)
- Accrual Clearing Management (ACRL_CLR_MGMT)
- Automated Accrual / Accrual Clearing Exclusions (R_ACRL_CLR_EXCL)
- Chart of Account Crosswalk (R_COA_CROSSWALK)

Major Output

- ACLA Transaction XML
- Accrual Clearing Management (ACRL_CLR_MGMT)
- Missing Automated Accrual transactions Report
- Exception Report

Chain Job Return code

The following table shows the potential return codes for the Automated Accrual Clearing Preprocess Chain job. Note that the Chain job will end with the highest return code across all of the jobs.

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

Problem Resolution

If any of the jobs in the chain failed due to application errors it is advisable to restart the job after correcting the errors instead of rescheduling the job. Restarting the job will reduce the processing time since the job will resume from where it has last committed and select only the unprocessed records. Please refer to the individual jobs for details regarding the specific job processes and problem resolution.

Automated Accrual Clearing Process Chain: Automated Accrual Clearing Preprocess Job

Job Name	Automated Accrual Clearing Preprocess
Recommended Frequency	See chain
Single Instance Required	Yes
Can be restarted?	No
Reports Generated	Yes

Overview

This process consists of following basic steps:

1. **Parameter Validation:** If the parameter validation is successful, the job will go to the next step. Otherwise the job will end with a return code of 'Failed'.
2. **Selection of Records:** Records from JRNL_ACTG are selected based batch parameters, starting at the JRNL_LOG record recorded from any previous run. Records from the Accrual Clearing Management (ACRL_CLR_MGMT) page are also selected for reprocessing when the Status there is Process.
3. **Processing of Records:** Selected JRNL_ACTG records are processed and ACLA transactions are created with Current Year and Previous Year Clearing Accounting line.
4. **Generate XML:** An XML file is generated with ACLA transactions.
5. **Generation of Missing Accrual Transactions report:** A report listing all the missing Automated Accrual Transactions will be created.
6. **Files for Later Job Steps:** Files are created for latter load and submit steps.

Process Steps	Messages
1. Parameter Validation	• Run Started • Each parameter is listed with value • Any parameter errors are listed
2. Selection of Records	• No journal records found matching accrual criteria (if no journal records selected) • If records are selected, no messages are issued for this step but for the next.
3. Processing of Records	• Journal records processed: ## (where ## is the count of journal records found matching clearing criteria) • No messages if records are not selected
4. Generate XML	• No messages issued for step
5. Files for Later Job Steps	• No messages issued for step • Run Ended

Restartability Information

N/A

Major Input

- Accounting Journal (JRNL_ACTG)
- Accrual Clearing Management (ACRL_CLR_MGMT)
- Automated Accrual / Accrual Clearing Exclusions (R_ACRL_CLR_EXCL)
- COA Crosswalk (R_COA_CROSSWALK)

Batch Parameters

Listed below are the delivered Default Values. Each implementation is encouraged to set any delivered (blank) to a constant value on the BATSETUP page. Also, there are other parameters that are updated annually.

Parameter	Description	Default Value
ACRL_CLR_EVNT_TYP	(Required) Disbursement Accrual Clearing Event Type	AC02
ACRL_DOC_CD	(Required) ACLA Transaction Department	ACLA
AMSEXPORT	(Required) AMS Export Import Directory	$$AMSEXPORT$$
AMSLOGS	(Required) The location of the log file	$$AMSLOGS$$
AMSPARM	(Required) Parameter Location for Upload Job	$$AMSPARM$$
CHK_PT_SIZE	(Required) Check Point Size is a performance parameter that controls check-pointing logic for job restartability.	5000
CLIENT_NM	(Optional) Client Name for Report	(blank)
COMMIT_SIZE	(Optional) Commit Block Size is a performance parameter that controls the number of records committed to the XML file	1000
DEL_ACRL_MGMT	(Required) The records from Accrual Clearing Management are deleted based on this parameter. Change periodically to Y to clear out the table of records that have been verified and marked for deletion.	N

DOC_CD	(Required) Accrual Clearing Transaction Code	ACLA
DOC_DEPT	(Optional) Populate with a single department to be used for all generated transactions. Leave blank to use the transaction department from the selected journal record. If leaving blank, ensure the Transaction Unit Code Required for the generated transaction is not checked on DCTRL if that same flag is not checked for all referenced transaction codes. If at least one referenced transaction code is unchecked, it must also be unchecked for the generated transaction code.	(blank)
DOC_UNIT	(Optional) Accrual Clearing Transaction Unit Code used for security purposes	(blank)
PREFIX	(Optional) Accrual Clearing Transaction ID Prefix used for ADNT record retrieval	(blank)
PROG_CTR_SIZE	(Optional) The Progression Counter defines the number of records processed that will trigger a progression message in the job log	100
PSTNG_CD	(Required) Posting Code(s) for JRNL_ACTG record selection. If retainage is left as an accrued expenditure (D011) on disbursements, then this parameter has to be D014, D011.	D014
SELECT_NON_REF	(Required) Decides whether Non-Referencing Transactions should be selected or not.	Y
SEL_BLK_SIZE	(Optional) Selection Block is a performance parameter that gives the number of journal records to be selected at once.	1000
SUBMIT_FILE_NM	(Required) File name for Submit job with ACLA Transaction Information	ACLASubmit.txt

Major Output

- AccrualClearing.xml (XML file for ACLA transactions)
- Missing Automated Accrual Transactions Report
- ACLASubmit.txt (text file containing the transaction information)
- Accrual Clearing Management (ACRL_CLR_MGMT)

Job Return Codes

The following table shows the potential job return codes for the Automated Accrual Clearing Preprocess.

Return Code	Condition
Successful (1)	Parameters were valid and at least 1 record was selected and processed successfully
Warning (4)	No eligible records found.
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 • Run time exceptions for unexpected situations. 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 Sequence

- None

Selection Criteria

- Accounting Journal Selection:
  - The records having the “Run Time Date Stamp (RUN_TMDT)” on JRNL_ACTG greater than the “Run date of the last Run (PROC_RUN_DT)” on Journal Log (JRNL_LOG) for this Process ID (ACCL) will be selected.
  - All records on the Accounting Journal (JRNL_ACTG) having Transaction Type (DOC_TYP) as either AD or MD with a reference to a PR or ABS transaction, Line Function Code (LN_FUNC_CD) as standard (1) or non-standard (2), a posting code equal to PSTNG_CD parameter will be selected.
  - If the SELECT_NON_REF parameter is Y, then in addition to the above selection, MD transactions without a reference, ITA, and IET transaction types are selected.

- The Automated Accrual / Accrual Clearing Exclusions (R_ACRL_CLR_EXCL) table will contain the list of records with various combinations of Transaction Code, Event Category, Event Type, Posting Pair, Fund, Department, Appropriation and Object. Any JRNL_ACTG records matching an exclusion record will be excluded from selection.

- Accrual Clearing Management Selection:
  - Those records with a Status (REC_STA) equal to Process (1) will be selected for reprocessing.
  - Those records with a Status (REC_STA) equal to Delete (3) will be selected to delete.

Problem Resolution

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

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	This step doesn't issue this return code.	N/A	N/A
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	Required Parameters are not entered **Sample Message:** The DOC_CD Parameter cannot be empty	Correct parameter error in a subsequent run of the chain.	N/A
	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Step 2: Selection of records

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Records selected are processed successfully	N/A	N/A
Warning (4)	No Accounting Journal records selected for processing	Verify selection criteria are correct and if not, then submit another instance. This may be a valid outcome if selection criteria were correct.	If selection criteria were wrong the JRNL_LOG progress tracking record will have to be reset if incremented so selection can read that range of journal records again.
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	In this step, the job can fail under the following conditions. 1) Encounters any runtime exceptions	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Step 3: Processing of records

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All records are successfully processed	N/A	N/A
Warning (4)	This step doesn't issue this return code.	N/A	N/A

Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	In this step, the job can fail under the following condition. 1) Encounters any runtime exceptions	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Step 4: Generate XML

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	XML file was successfully generated	N/A	N/A
Warning (4)	This step doesn't issue this return code.	N/A	N/A
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	In this step, the job can fail under the following conditions. • Issue creating XML file, for example, Import/Export file location not found. • Encounters any runtime exceptions	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the failure needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Step 5: Files for Later Job Steps

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Files were successfully generated	N/A	N/A
Warning (4)	This step doesn't issue this return code.	N/A	N/A
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	In this step, the job can fail under the following conditions. • Issue creating parameter files, for example, parameter file location not found. • Encounters any runtime exceptions	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the failure needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Step 6: Creation of Missing Automated Accrual Transactions Report

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	The Missing Automated Accrual Transactions Report generated successfully.	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non-Fatal Error (8)	N/A	N/A	N/A

Failed (12)	Failed due to issues in creating the report.	Investigate the reason for the report not being created and re-run the chain.	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 and a new job 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 and a new job scheduled.	N/A.

Automated Accrual Clearing Process Chain: Accrual Clr Load

Job Name	Accrual Clr Load
Recommended Frequency	See chain
Single Instance Required	Yes
Can the job be restarted?	Yes
Reports generated	No

Overview

After the generation of the ACLA transaction xml file, the next step of the Automated Accrual Clearing Process chain is the Accrual Clr Load. This step uses the MultiProcessImport job with parameter as ACLA Transaction XML file name.

This MultiProcessImport job will internally spawn multiple SysManUtil jobs by splitting the ACLA Transaction XML file into the parts as specified by the THREAD_COUNT parameter. The ACLA transactions are selected from the ACLA Transaction XML file and loaded / submitted into the application by using the System Maintenance Utility (SMU).

Major Input

- ACLA Transaction XML
- ACLASubmit.txt (text file containing the transaction information for submit)

Batch Parameters

Parameter	Description	Default Value
BLOCK_SIZE	(Required) Number of records in each split segment of the input file.	100

COMMIT_BLOCK_SIZE	(Required) Number of records to commit at a time.	100
FILE_INPUT_DIR	(Required) The location of the source files of records.	$$AMSROOT$$/ExportImport
FILE_LIST	(Required) Comma separated list of files to be uploaded with multi-threaded processing.	AccrualClearing.xml
FILE_OUTPUT_DIR	(Required) Output location for the file segments.	$$AMSROOT$$/ExportImport
FILE_PREFIX	(Required) Prefix used on the filenames for the output file segments.	ACLA
I_SMU_APPLY_OVERRIDES	(Required) If overrides are to be applied to transactions as they are loaded this parameter should be true. A setting of false will result in transactions rejecting for override errors.	true
I_SMU_BYPS_ADNT_FL	(Required) Bypass ADNT Flag	true
I_SMU_DOC_STA_CD	(Required) Transaction Status For Loaded Transactions 1-Held, 2-Ready	2
I_SMU_OVERRIDE_LVL	(Optional) When an override level should be applied other than that of the user id submitting the batch job, that level should be entered in this parameter field. If left blank the override level of the user id submitting the job will be used if the Apply Overrides parameter is true.	(blank)
LOG_STATUS_INTERVAL	(Required) Logging frequency (in seconds) for controller thread reporting status of child threads to the system log	300
MODE	(Required) Mode of operation. (1=Import, 2=Import and Submit, 3=Import and Other Action)	2
SLEEP_INTERVAL	(Required) Polling frequency (in seconds) for internal controller thread for checking child processes.	5
SMU_CTLG_ID	(Required) Catalog id of the System Maintenance Utility job which is spawned as the child process.	3

SMU_EXCEP_REP_FILE_NM	(Required) Exception File Name	ExcepACLASubmit.txt
S_SMU_EXCEP_REP_IND	(Required) Exception Detailing (1- Detailed, 2-Failed Transactions, 3- Processed Transactions, 4-Failed Transaction Lines, 5-Transaction Status)	4
STAGGER_TIME	(Required) The lag time, in seconds, between the spawning of each child process.	1
THREAD_COUNT	(Required) Parameter defines the number of SMU threads to use for processing.	8

Please refer to the SMU Transaction Upload / Submit Job run sheet in the *CGI Advantage Financial – Utilities Run Sheet Guide* for the full list of SMU Transaction upload / submit batch parameters.

Major Output

- ACLA Transactions in draft version

Job Return code

The following table shows the potential job return codes for the Inferences & Validations job:

Return Code	Condition
Successful (1)	All of the records are loaded into the Transaction Catalog successfully or the input file is empty. All of the transactions loaded were submitted successfully
Warning (4)	This return code will be issued when some of the records failed to load whereas all other records were loaded successfully.
Non-Fatal Error (8)	None of the records get loaded into the Transaction Catalog.
Failed (12)	Parameters are invalid When the input file is not found in the specified directory Restart failed because another instance of the Accrual Clear Process chain has already been run successfully Runtime exceptions encountered for any unexpected situations When the job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a

	return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Sequence

N/A

Selection Criteria

N/A

Problem resolution

The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain. For general errors and recommendations, refer to the SMU Transaction Upload run sheet.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All transactions were loaded / submitted	N/A	N/A
Warning (4)	At least 1 transaction didn't load	The reason for the failure needs to be investigated before restarting the job. The transaction(s) that didn't load are in the Import/Export Error directory.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.
Non-Fatal Error (8)	No transactions were able to load	The reason for the failure needs to be investigated before restarting the job.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.
Failed (12)	In this step, the job can fail under the following conditions. • Issues in spawning jobs • Files not found • Runtime exceptions • Parameter Edits	The reason for the failure needs to be investigated before restarting the job.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.
Terminated (16)	Job is terminated manually by the user.	The reason for the failure needs to be investigated before re-starting the job.	If another instance of the chain has run,

			then the XML file has to be loaded with a separate SMU instance.
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 re-starting the job.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.

Automated Accrual Clearing Process Chain: Accrual Clr Exception Job

Job Name	Accrual Clr Exception
Recommended Frequency	See chain
Single Instance Required	Yes
Can be restarted?	No
Report Generated	Yes

Overview

This batch job will generate exception report for the failed ACLA transactions in earlier step. The report will be generated by showing accrual clear transactions, the referenced transaction and all errors issued.

Major Input

- ExcepACLASubmit.txt
- ACLA Transactions

Batch Parameters

Parameter	Description	Default Value
EXCEP_PARM_FILE_NM	(Required) Exception Report File Name that got generated by SMU	$$AMSEXPORT$$/ExcepACLASubmit.txt
RUN_TYP	(Required) Run Type (1 = Accrual, 2 = Accrual Clearing)	2

CLIENT_NM	(Optional) Client Name for the report header	N/A

Major Output

- Exception Report

Batch Return codes

The following table shows the potential job return codes for the individual Accrual Clr Exception job.

Return Code	Condition
Successful (1)	Exception report was not generated. All the ACLA transactions were submitted successfully.
Warning (4)	Exception report was created with failed ACLA transactions
Non-Fatal Error (8)	Not Applicable for this job.
Failed (12)	Input SMU log file was not found
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. When this job ends with a return code of System Failure.

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All transactions are final	N/A	N/A
Warning (4)	One or more ACLA transactions rejected	Address the errors and submit the transactions either manually or with an SMU.	N/A
Non-Fatal Error (8)	The job does not end with this return code	N/A	N/A

Failed (12)	In this step, the job can fail under the following conditions. • Exception file not found • Runtime exceptions • Parameter Edits	The reason for the failure needs to be investigated before restarting the job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the failure needs to be investigated before re-starting 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 before re-starting the job.	N/A

2.1.11 Automated Bank Account Transfer Process

The Automated Bank Account Transfer Process in Advantage Financial is used to facilitate the movement of bank account balances between various bank accounts in both Advantage Financial and in the actual bank account at a bank. This Automated Bank Account Transfer Process facilitates the movement of cash between various bank accounts through the creation of Journal Voucher Bank Transfer transactions to update Advantage Financial bank account balances and through the creation of an Automated Clearing House (ACH) file to move cash to an appropriate bank account at a bank.

Description

The Automatic Bank Account Transfer Process is used to record cash decreases and increases in the 'real' bank accounts, remove the updates to the 'clearing' bank(s), and create wire transfers to move the actual cash between external banks. The Automated Bank Account Transfer Process facilitates the movement of cash between various bank accounts through the creation of Journal Voucher Bank Transfer transactions to update Advantage Financial bank account balances, as well as through the creation of an Automated Clearinghouse (ACH) file to move cash to an appropriate bank account at a bank.

The process involves scheduling of a chain job comprising of following sub-processes:

- Chain Job 1 - Automated Bank Account Transfer Process
- Chain Job 2 – Load and Submit Journal Voucher Bank Transfer transactions
- Chain Job 3 – Generate Exception Report
- Chain Job 4 – Generate Automated Bank Account Cash Transfer Process Report
- Chain Job 5 – Generate ACH(XML) File
- Chain Job 6 – Generate ACH(.dat) File

When to Run

The process can be run on request

Job	Description	Output
Automated Bank Account Transfer Process	Selects records from Cash Journal to be retained, summarizes them, and generates an XML file for them to load as Journal Voucher Bank Transfer transactions into the system.	.xml transaction file (for system use)
Load and Submit JBVK transactions	Loads the generated Journal Voucher Bank Transfer transactions and submits them.	Journal Voucher Bank Transfer transactions loaded to the Transaction Catalog and processed along with their updates to the system.
Generate Exception Report	Generates Exception Report for Journal Voucher Bank Transfer transactions not submitted	Exception Report

	successfully	
Generate Automated Bank Account Cash Transfer Process Report	Generates Automated Bank Account Cash Transfer Report enumerating banks and the amount transferred between them.	Automated Bank Account Cash Transfer Report
Generate ACH(XML) File	Generate the ACH .xml data file which will be picked up by the subsequent sub-process to generate ACH .dat file.	.xml file for ACH generation (for system use)
Generate ACH(.dat) File	Generates the ACH file	ACH .dat file

Parameters

Important Note – There is a common parameter namely PARM_FILE running across all sub-processes. This parameter is not overrideable. The file mentioned in the PARM_FILE parameter is used by the system for passing information across sub-processes. If by any chance the PARM_FILE value for sub-process 1 is changed then same value for PARM_FILE has to be specified for rest of the five subsequent sub-processes.

Job	Parameter	Description	Default Values	For System Use
AUTOMATED BANK ACCOUNT TRANSFER PROCESS	Client Name (CLIENT_NM)	Optional field for the name of client to be appear on report	No default	
	Transaction Code (DOC_CD)	Required Field for a Transaction Code to be used on the generated Journal Voucher Bank Transfer transactions. Used to generate the JV Transaction.	No default	
	Transaction Department Code (DOC_DEPT_CD)	Required field for a Department Code to be used on the generated Journal Voucher Bank Transfer transactions. Used to look up a numbering entry on the Automatic Transaction Numbering table.	.No default	
	Transaction Prefix (DOC_PFX)	Required field for a Prefix to be used on the generated Journal Voucher Bank Transfer transactions. Used to look up a numbering entry on the		

		Automatic Transaction Numbering table.		
	Transaction Record Date (RUN_DT)	Optional Run Date in MM/DD/YYYY format needed to arrive at Effective Entry Date on ACH file, which is computed as Run Date plus Lag Days. If no value is entered, System takes the current date as Run date. (** **Refer to Note**: Pivot Date/Year Validation, while entering the date)	Current System Date	
	Transaction Unit Code (DOC_UNIT_CD)	Optional field for a Transaction Unit Code for the generated Journal Voucher Bank Transfer transactions. Used to facilitate security at a level below department.	No default	
	Excluded Posting Codes (EXC_PSCD_CD)	Required Field for posting codes. The Posting Codes that will be excluded from the records selected for the Automated Bank Account Transfer Process.	No default	
	Transfer Posting Code (XFER_PSCD_CD)	Required Field for posting codes. Posting Code that will be used in order to transfer cash from Bank Account on the Journal to Master Bank Account on the Fund table.	No Default	
	Transfer Event Type (EVNT_TYP_ID)	Required Field for Event type. Used in order to transfer cash from Bank Account on the Journal to Master Bank Account on the Fund table.	GA22	
	Input Source (INPUT_SRC)	Required Field for Journal names. Used to select records for the Automated Bank Account Transfer Process		
	EXCEP_REP_FILE_NM	Exception .txt file enumerating errors, if any, during Journal Voucher Bank Transfer transaction submit	AbatpException.txt	Yes

	EXCEP_REP_IND	Exception Report Indicator describing the detail up to which the exception report is to be generated.	4 Currently exception detailed report up to Failed Transaction lines is supported.	Yes
	SUM_OPT	Summarization Option (3=Summarization for Limited Data Elements - Fund and BSA)	3	
	FILE_NM	Export .xml file name for generated Journal Voucher Bank Transfer transactions	ABATPDoc.xml	Yes
	Lag Days (LAG_DY)	Optional field for the number of Lag Days needed to arrive at Effective Entry Date on ACH file, which is computed as Run Date plus Lag Days.	0	
	ACH_FILE_NM	ACH (.dat) file name	AbatpACH.dat	Yes
	PARM_FILE	Parameter (.txt) file	AbatpParams.txt	Yes
	SUBMIT_FILE	Submit Parameter (.txt) file	AbatpSubmit.txt	Yes
	EXC_DOC_CD	Transaction Codes(s) to be excluded during processing	JVAC, JVBK	
	AMSEXPORT (** **Refer to Note**: Assumptions for SWBP on page no. 6)	Export Location at Automated Bank Account Transfer Process Job	-	Yes
	AMSPARM (** **Refer to Note**: Assumptions for SWBP on page no. 6)	Parameter Location at Automated Bank Account Transfer Process Job	-	Yes
	AMSLOGS (** **Refer to Note**: Assumptions for SWBP on page no. 6)	Logs Location at Automated Bank Account Transfer Process Job	-	Yes
	MAX_PREFETCH_COUNT	Added MAX_PREFETCH_COUNT parameter.		
Load JBVK transaction	PARM_FILE	Parameter (.txt) file	$$AMSPARM$$/ AbatpParams.txt	Yes

s				
Submit JBVK transactions	PARM_FILE	Submit Parameter (.txt) file	$$AMSPARM$$/ AbatpSubmit.txt	Yes
Generate Automated Bank Account Cash Transfer Process Report	PARM_FILE	Parameter (.txt) file	AbatpParams.txt	Yes
	AMSPARM	Parameter Location at Generate ABATP Report Job	-	Yes
Generate Exception Report	PARM_FILE	Parameter (.txt) file	AbatpParams.txt	Yes
	AMSPARM (** **Refer to Note**: Assumptions for SWBP on page no. 6)	Parameter Location at ABATP Exception Report Job	-	Yes
	AMSLOGS (** **Refer to Note**: Assumptions for SWBP on page no. 6)	Logs Location at ABATP Exception Report Job	-	Yes
Generate ACH(XML) File	PARM_FILE	Parameter (.txt) file	AbatpParams.txt	Yes
	AMSEXPORT (** **Refer to Note**: Assumptions for SWBP on page no. 6)	Export Location at Generate ACH XML File Job	-	Yes
	AMSPARM ** **Refer to Note**: Assumptions for SWBP on page no. 6)	Parameter Location at Generate ACH XML File	-	Yes
	SETTLEMENT_DT	Settlement Date. Valid values are: • 1 (sets value to 000), • 2 (sets value to 3 blank spaces), and • alphanumeric string of	1	Yes

		length 3.		
Generate ACH(.dat) File	PARM_FILE	Parameter (.txt) file	AbatpParams.txt	Yes
	AMSEXPORT (** **Refer to Note**: Assumptions for SWBP on page no. 6)	Export Location at Generate ACH DAT File Job	-	Yes
	AMSPARM (** **Refer to Note**: Assumptions for SWBP on page no. 6)	Parameter Location at Generate ACH DAT File Job	-	Yes
	Commit Block Size (Default = 1): (COMMIT_BLK)	Optional Field. Used to set custom commit block size for Finance Charge Process	1	
	Effective Entry Date (CLR_DT)	Effective Entry Date (MM/DD/YYYY format) to be entered on ACH .dat file in Batch Header record (if not entered then defaulted to Current date)	Current System Date	

Major Input

- The Cash Journal (JRNL_CASH)

Peripheral DataObjects

- Posting Code (R_PSCD)
- Bank (R_AP_BANK_ACCT)
- Disbursement Format (R_AP_DISB_FRMT)
- Automatic Transaction Numbering (AUTO_DOC_NO)
- Journal Log (JRNL_LOG)

Output

- Journal Voucher Header (JV_DOC_HDR)
- Journal Voucher Line Group (JV_DOC_LNGRP)
- Journal Voucher Accounting Line (JV_DOC_ACTG)
- Automated Bank Account Cash Transfer Report
- Exception Report
- ACH File

Sort Criteria

- BANK_ACCT_CD

Selection Criteria

Following are the selection criteria for the record selection, which serves as Data input from Cash Journal to the process

- Posting Code on Cash Journal Record
- Select the Cash Journal records having posting codes with Cash Account Flag checked in the Inference and Edit Information section on the Posting Code table.
- Exclude the Cash Journals having posting codes entered for the batch parameter 'EXC_PSCD_CD'
- Transaction Code on Cash Journal Record
- Select the Cash Journal records having transaction codes not equal to the ones entered for the batch parameter 'EXC_DOC_CD'.
    - This is to exclude all the Cash Journal records, which were created either as the result of a prior execution of the Automated Bank Account Transfer Process or the execution of the Annual Close Process.
- Record Number on Cash Journal Record
- Read in the Last Processed Record Number from Journal Log for Process Id is ABATP (Automated Bank Account Transfer Process) and Run Type is Initial.
- Select all Cash Journal records with Record Number greater than the Last Processed Record Number.

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.

If a Journal Voucher Bank Transfer fails, then manual intervention could be necessary.

This batch program produces new Advantage transactions, which are subject to the line limit functionality constraints. Sites should ensure that they run this job with parameters set to ensure that the created transactions are within the line limit controls.

For the Generate ABATP Report job in the ABATP chain, if any of the JVBK transactions for which the report is to be created is NOT in a Phase of Final it will set the return code as Non Fatal. These transactions must be corrected and submitted to Final before the job will complete successfully. After correcting errors and submitting the transactions, the ABATP Report job should be restarted. Upon successful completion, the remaining jobs, which previously went inactive, will be submitted.

For performance reasons, the Identify and Archive Stale Gaps job should be run periodically to remove gaps in journal records being tracked and processed by this program. Please refer to the Identify and Archive Stale Gaps run sheet for more details on scheduling and recommended parameters.

2.1.12 Begin Day Balance Batch

Job Name	Begin Day Balance Batch
Recommended Frequency	Daily as part of the nightly cycle. This job should be run at the end of nightly cycle after all processing for a day is done and before processing any transactions after the Begin Day job runs.
Single Instance Required	Yes
Can be restarted?	No – Run new instance
Reports generated	N/A

Overview

The Begin Day Balance Batch job populates the Begin Day Balance amount on the ITD Balance Sheet Detail (BBALD) and ITD Balance Sheet Summary (BBALS) pages with the value in the Current Balance amount.

If an instance of this job fails to run for one or more days, when an instance is run the information will be correct. There is no need to perform any type of correction within the application. Any such data written to a data warehouse for those days when the job didn't run will be incorrect and should be updated in a manner similar to the logic of this batch job.

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

Process Steps	Messages
Update of Records	All records from R_BBAL_ITD and R_BBAL_ITD_SMRY are selected and updated.

Restartability Information

Not available as a new instance of the job can be run. Please see the Problem Resolution section for more details.

Major Input

- Tables
- ITD Balance Sheet Detail (R_BBAL_ITD)
- ITD Balance Sheet Summary (R_BBAL_ITD_SMRY)
- Parameters
- None

Major Output

- ITD Balance Sheet Detail (R_BBAL_ITD)

- ITD Balance Sheet Summary (R_BBAL_ITD_SMRY)

Job Return Codes

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

Return Code	Condition
Successful (1)	All records processed successfully
Warning (4)	N/A
Non-Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: - 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. When this job ends with a return of code System Failure.

Sort Sequence

None

Selection Criteria

None

Problem Resolution

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

Step 1: Update of Records:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All 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.	The reason for the failure needs to be investigated before running another instance.	n/a
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before running another instance.	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 running another instance.	n/a

2.1.13 Cash and Fund Balance Synchronization Process

Job Name	Cash and Fund Balance Synchronization Process
Recommended Frequency	Once. Please run SA02 in Full Mode before running this process.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Cash/Fund Balance Synchronization Report. If run in Update mode, displays which records are synchronized. If in Report mode, displays records that need to be synchronized.

Overview

- This process synchronizes Balance Table records (R_CBAL / R_FBAL) with SA_NONBUD, Audit Tables (R_CBAL_AUD / R_FBAL_AUD) and BSA Balance Tables (R_BSA_XBAL).
- Similarly, Summary and Pool Tables are synchronized.
- It also synchronizes BSA Balance Table with Inception-to-Date Balance Sheet Balance Table (R_BBAL_ITD) and Audit Tables.

The Job can be run in two modes:

- Update Mode: In this mode, the records are synchronized. Each record synchronized is written to the report.
- Report Mode: In this mode, the out of sync records are reported.

The following are the major steps in the job:

1. Synchronization of Balance table records with SA_NONBUD
2. Inserting missing records into the Balance table from SA_NONBUD
3. Resetting of Orphan records of Balance table.

   *In the above three steps, the necessary adjustments with Audit Tables is also done.
4. Synchronization of BSA Balance Tables with Inception-to-Date Balance Sheet Balance Table.
5. Synchronization of BSA Cash Balance table with BSA Cash Balance Audit Table.
6. Resetting Orphan BSA Cash Balance Audit records.
7. Synchronization of Balance tables with BSA Balance Tables.
8. Synchronization of Balance tables with Audit Tables.
9. Synchronization of Summary tables with Balance Tables.
10. Synchronization of Cash Balance Pool table with Cash Balance Summary Table.

Job Details:

1. Synchronization of Balance table records with SA_NONBUD

This step synchronizes Balance Table records with values of SA_NONBUD. If non-zero Pending amounts exist in the Audit Table for the current Balance Table record, then those amounts are also synchronized.

Find all the records where joining Balance Table with SA_NONBUD finds that either of the two amount fields does not equal the equivalent inside of the other table where the NONBUD_TAB_ID equals the Balance Table being compared. If Fund / SFund is given it is also used in the query.

So, for CBAL we would join R_CBAL with SA_NONBUD

where NONBUD_TAB_ID = R_CBAL and

SUM of NCASH_ACC_DCRS_AM from SA_NONBUD <> NCASH_ACC_DCRS_AM from R_CBAL OR

the SUM of NCASH_ACC_INCR_AM from SA_NONBUD does not equal NCASH_ACC_INCR_AM from R_CBAL.

For FBAL, REV_ACCT_BAL_AM and EXP_ACCT_BAL_AM are considered.

For each record fetched, the amounts from SA_NONBUD is updated on the Balance Table.

If matching record exists on Audit table, the Pending amounts are also synchronized.

For CBAL, the Pending amounts are NCASH_PEND_DCRS_AM and NCASH_PEND_INCR_AM. For FBAL, they are REV_PEND_DCRS_AM and REV_PEND_INCR_AM.

Records are sorted by Fund Code and Sub Fund Code.

2. Inserting missing records into Balance table from SA_NONBUD

This step inserts records into the Balance Table.

Find all records which exist on SA_NONBUD having at least one non-zero amount, but which do not exist on the Balance Table. For each SA_NONBUD record found insert a new record into the Balance Table. Set the values of Fund and Sub Fund Code and of the amounts.

If matching record exists on Audit table, the Pending amounts are also synchronized.

3. Resetting of Orphan records of Balance table.

This step acquires the set of Balance table records not found in SA_NONBUD and resets their amounts to Zero.

If matching record exists on Audit table, the Pending amounts are also synchronized.

4. Synchronization of BSA Balance Tables with Inception-to-Date Balance Sheet Balance Table.

In this step, all records on BSA Balance Table are selected whose CURR_BAL does not match with that on ITD Balance Sheet Detail. These records are then updated with the CURR_BAL value of ITD Balance Sheet Detail.

If BSA Balance Table record exists without a corresponding record on ITD Balance Sheet Detail, then the CURR_BAL on BSA Balance Table is reset to zero.

Records are sorted by Fund Code, Sub Fund Code, BSA Code and SBSA Code.

5. Synchronization of BSA Cash Balance table with BSA Cash Balance Audit Table.

All BSA_CBAL records are selected for which Pending Amounts are not equal to the Sum of BSA_CBAL_AUD records.

CASH_PEND_INCR_AM and CASH_PEND_DCRS_AM are the amounts to be adjusted.

6. Resetting Orphan BSA Cash Balance Audit records.

This step is meant to zero out any BSA_CBAL_AUD records that do not have BSA_CBAL records mapped to them.

7. Synchronization of Balance table with BSA Balance Tables.

In this step, all Balance Table records whose relevant amounts do not equal the Summation of corresponding BSA Balance Table are selected.

For Cash Balance, CURR_BAL, CASH_PEND_INCR_AM, CASH_PEND_DCRS_AM are the relevant amounts that are updated with the value from BSA Balance Table.

For Fund Balance it is CURR_BAL.

Records are sorted by Fund Code and Sub Fund Code.

8. Synchronization of Balance tables with Audit Tables.

Select all the Balance Table records whose buckets do not equal the Sum of Balance Table Audit records.

The buckets for Cash Balance are NCASH_PEND_INCR_AM and NCASH_PEND_DCRS_AM.

For Fund Balance they are: REV_PEND_INCR_AM, REV_PEND_DCRS_AM, EXP_PEND_INCR_AM and EXP_PEND_DCRS_AM.

9. Synchronization of Summary tables with Balance Tables.

The first step inserts missing records into the Summary Table. If any Balance Table records which do not have Summary records exist, then a record is added into the Summary Table.

The next step synchronizes the Summary Table with Balance Table records.

10. Synchronization of Cash Balance Pool table with Cash Balance Summary Table.

This step is used to sync up the Cash Balance Pool Table.

If the CBAL Pool Parameter on the Application Control Table is true then insert missing records and also sync up existing records.

If false, then reset all non-zero buckets on the Pool Table.

Process Steps	Messages
1. Parameter Validation	• Parameter validation started. • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. • Parameter validation complete.
2. SA_NONBUDTable Validation	• Table validation started. • If the selection returns 0 records, then the following message will be issued: “No records found on SA_NONBUD. Please run SA2 in full mode before running this process”. • Parameter validation complete.
3. Synchronization of Balance table records with SA_NONBUD	• Synchronization of Balance Table records with SA_NONBUD started. • The Progress of this step is shown by displaying the message “Number of records processed are:” based on the parameter Progression Counter Size. • Synchronization of Balance Table records with SA_NONBUD completed.
4. Inserting missing records into Balance table from SA_NONBUD	• Inserting missing records into Balance Table started. • The Progress of this step is shown by displaying the message “Number of records processed are:” based on the parameter Progression Counter Size • Inserting missing records into Balance Table completed.
5. Resetting of Orphan records of Balance table.	• Resetting of Orphan records of Balance Table started. • The Progress of this step is shown by displaying the message “Number of records processed are:” based on the parameter Progression Counter Size • Resetting of Orphan records of Balance Table completed.
6. Synchronization of BSA Balance Tables with ITD Balance Sheet Balance Table.	• Synchronization of BSA Balance Table with BBAL_ITD started. • The Progress of this step is shown by displaying the message “Number of records processed are:” based on the parameter Progression Counter Size • Synchronization of BSA Balance Table with BBAL_ITD completed.
7. Synchronization of BSA Cash Balance table with BSA Cash Balance Audit Table.	• Synchronization of BSA Balance Table with BSA_CBAL_AUD started. • The Progress of this step is shown by displaying the message “Number of records processed are:” based on the parameter Progression Counter Size • Synchronization of BSA Balance Table with BSA_CBAL_AUD completed. • No. of BSA CBAL Audit records reset for missing BSA CBAL are:” ”.

8. Synchronization of Balance table with BSA Balance Tables.	• Synchronization of Balance Table with BSA Balance Table started. • The Progress of this step is shown by displaying the message "Number of records processed are:" based on the parameter Progression Counter Size • Synchronization of Balance Table with BSA Balance Table completed.
9. Synchronization of Balance tables with Audit Tables.	• Synchronization of Balance Table with XBAL_AUD started. • The Progress of this step is shown by displaying the message "Number of records processed are:" based on the parameter Progression Counter Size • Synchronization of Balance Table with XBAL_AUD completed.
10. Synchronization of Summary tables with Balance Tables.	• Synchronization of Summary with Balance Table started. • The Progress of this step is shown by displaying the message "Number of records processed are:" based on the parameter Progression Counter Size. • Synchronization of Summary with Balance Table completed.
11. Synchronization of Cash Balance Pool table with Cash Balance Summary Table.	• Synchronization of CBAL_POOL with CBAL_SMRY Table started. • The Progress of this step is shown by displaying the message "Number of records processed are:" based on the parameter Progression Counter Size. • Synchronization of CBAL_POOL with CBAL_SMRY Table completed.

- This job can be run in report mode to find out of sync records. Multiple instances of the job can be run.
- In Update mode only one instance can be run.

Major Input

- SA_NONBUD table
- Balance Table - R_CBAL / R_FBAL
- BSA Balance Table - R_BSA_CBAL / R_BSA_FBAL
- Balance Audit Table - R_CBAL_AUD / R_FBAL_AUD
- ITD Balance Sheet Table - R_BBAL_ITD
- BSA Cash Balance Audit Table – R_BSA_CBAL_AUD
- Balance Summary Table – R_CBAL_SMRY

Batch Parameters

Parameter	Description	Default Value
Table Name TABLE_NAME	The Table for which the Data is to be corrected. Enter (1) Cash Balance (2) Fund Balance	
Report Mode. REPORT_MODE	Enter (1) Update and Report out Of Synch records (2) Report on Out Of Synch records	
Fund Code. FUND_CD	The FUND for which the Data is to be corrected.	
Sub Fund Code. SFUND_CD	The SUB FUND for which the Data is to be corrected.	
BSA Code. BSA_CD	The BSA for which the Data is to be corrected.	
Sub BSA Code. SBSA_CD	The SUB BSA for which the Data is to be corrected.	
Client Name CLIENT_NM	The Client for which the report is to be generated.	
Report ID REPORT_ID	The REPORT ID for current run.	
Commit Block Size COMMIT_BLK	Set the commit block size using this value.	
Select Block Size SELECT_BLK	Set the select block size using this value.	
Program Counter Size PROG_CTR	Set the Program Counter Size using this value.	

Major Output

- Balance Table - R_CBAL / R_FBAL
- Balance Summary Table - R_CBAL_SMRY / R_FBAL_SMRY Table.
- Cash Balance Pool Table - R_CBAL_POOL Table.
- BSA Balance Table - R_BSA_CBAL / R_BSA_FBAL Table.
- BSA Cash Balance Audit – R_BSA_CBAL_AUD
- Summary Tables – R_CBAL_SMRY / R_FBAL_SMRY
- Pool Table – R_CBAL_POOL.
- Cash / Fund Balance Synchronization Report.

Job Return Code

For a stand-alone job, the job return codes should be discussed in detail as shown in the following table. Conditions shown here are only examples applicable for a process job and needs to be changed for each job. For SMU jobs this will change based on the SMU action. Refer to the Job Return Code diagram for more details.

Return Code	Condition
Successful (1)	All of the parameters are valid and the records are processed successfully or no out of sync record is found.
Warning (4)	This return code will be issued when no records are found on the SA_NONBUD Table.
Non-Fatal Error (8)	The job status will be Non-Fatal Error when all records could not be synchronized.
Failed (12)	The job will fail 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 or network issues.

Sort Criteria

See Job Details.

Selection Criteria

See Job Details.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully. Synchronization is successful for all records or No out of sync records found.	N/A	N/A
Warning (4)	No Records are present on the SA_NONBUD Table. If Fund and/or Sub Fund parameters are given, then no	If no records are present on SA_NONBUD, please run SA02 in Full Mode.	N/A

	matching records exist on SA_NONBUD.		
Non-Fatal Error (8)	Job ended with a Non-Fatal Error because some of the records could not be synchronized.	Refer to the logs generated by the job to find out why the records could not be synchronized.	N/A
Failed (12)	All of the required parameters are not entered. **Sample Message:** Table parameter is required. **Recommendation:** Enter the Table for which job is to be run.	If the required parameters are not entered, enter the required parameters and schedule a new job.	
	Entered Parameters are not valid. **Sample Message**: Invalid Table Parameter. It can be 1 or 2. **Recommendation:** Enter a valid value for Table Parameter.	If the parameters are invalid, enter the valid parameter and Schedule a new job.	
	Job failed due to Fatal conditions.	Encounters any runtime exceptions. Investigate the exception reported by the process, resolve the error and restart the job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	

Important Notes:

- This job does not deal with BBAL. There is a job entitled Rebuild BBAL which will deal with BBAL records. So, when Balance Table is mentioned it stands for either CBAL or FBAL based on the report parameter.
- The SA_NONBUD table is considered the source of truth. It is left to the users of this job to evaluate if SA_NONBUD is correct and if they wish to use this job or not.
- When ensuring that BSA Balance Table's CURR_BAL is in synch, BBAL_ITD Table is seen as the source of truth. So, if this is off, then BSA Balance Table will be off Balance Table's.
- When ensuring Balance Table's XBAL_AM field, it will use BSA Balance Table as the source of truth in terms of resetting the XBAL_AM to be the summation of the BSA Balance Table's CURR_BAL.

2.1.14 Clearing Account Maintenance

Chain or Job Name	Clearing Account Maintenance Chain
Recommended Frequency	To be run on demand
Single Instance Required	Yes
Can be restarted?	See individual job steps
Reports generated	Yes, the last job in the chain generates 4 reports.

Overview

Clearing Account Maintenance is a group of jobs that work together to create, load and submit journal vouchers, based on selection criteria established on the Clearing Account Maintenance Parameters (CAMP) page. Clearing accounts are those used for internal accounting (between two parties on the same Advantage Financial application) when cash is not desired as the real account. Due To and Due From are the common names of each clearing account.

Use of clearing accounts depends on site-specific needs. Examples of when they are used include:

- Fund values for each party are different and use different bank accounts. In this instance, if cash were used, then Advantage Financial would have different cash balances than the actual bank accounts because cash was transferred within the application, but not between the actual banks. A manual wire transfer or the Bank Account Transfer chain job would have to be used to bring the actual banks in sync with the account within Advantage.
- The internal activity occurs within an adjustment period after the end of a fiscal year when cash should not be changing in that prior fiscal year.

The primary use of this program is to facilitate clearing Due To and Due From accounts from internal transactions against a prior fiscal year in a subsequent year when cash updates against that prior year have been cut off. These clearing accounts are cleared out through cash in the new fiscal year. It can also be used to clear those accounts throughout the year; given accounting model setup has been performed to ensure accrued nominal amounts are not stranded in the clearing. Please see the section for the Clearing Account Maintenance program in the *General Accounting Users Guide* for more information on this program. Preparation for running this program has to be done before any transaction processing is done for internal accounting.

The chain job has options of both a *report* and an *update* run mode. If the report option is chosen on the CAMP table, only the first job in the chain should be run with the last three being disabled. If not, the second job will end with a return code of warning as it will not be able to find any XML files. The remaining two jobs will end as inactive. The same result also occurs in update mode if selection criteria results in no records being selected. The report mode is very useful in determining the volume and the records that will be selected if run in update mode (with the understanding that any transaction activity between the report run and the update run that meets selection criteria will also be selected).

The Clearing Account Maintenance chain has the following jobs (each of the jobs listed below, is described in subsequent sections):

1. Clearing Account Maintenance
2. Load JV Transactions
3. Submit JV Transactions

4. Create Reports

Major Input

- Journal Log (JLOG): JRNL_LOG is used to record starting and ending journal records for a given selection criteria set so the subsequent run with the same criteria pick up where the previous left off.
- Bank Account (BANK): R_AP_BANK_ACCT is used to supply the BSA/SBSA value(s) to be used on cash lines if the Cash BSA and Cash SBSA parameters are not used.
- Clearing Account Maintenance Parameters (CAMP): CAM_PROC_PARM supplies selection, run mode, output mode, input journal, transaction creation information, and values for generated transactions.
- Input Journal (JINT, JACTG, JFAAJ): JRNL_INT is likely the most efficient input source. However, it cannot be used to clear Due To and Due From accounts generated from fixed asset internal sales. JRNL_FA is the most efficient input source for clearing fixed asset related clearing accounts. JRNL_ACTG is the single input that contains all clearing account activity, but is the largest input data source available.

Major Output

- Clearing Account Maintenance temporary table: R_CLR_ACT_MNT_TMP
- Journal Voucher transactions in XML format
- Journal Log (JLOG): See Major Input section for details.
- Transaction Exception Report: This report lists all journal vouchers created that failed to submit successfully. When no transactions fail to submit, the report lists one line reading: 'All transactions submitted successfully.' More details are provided in the Create Report job step.

```
Report ID:CAM1                                                          PAGE : 1
Run Date: 01-31-2007              Document Exception Report
Run Time: 14:07:26

Code          Dept          ID

JVCAM         JTY4          01310700000000000016

Total Number of Documents: 1
```

- Journal Record Exception Report: This report lists those journal records that were not processed because there were not two records selected for a given accounting line. The program will not clear ½ of an internal transaction unless it came from a fixed asset transaction (accounting lines always have one clearing account per accounting line.) A large number of records on this report are indicative of bad selection criteria that only matched ½ of many internal transactions. If that was not the cause, then records on this report (and the other ½) have to be cleared manually. When no exception records are found, the report lists one line reading: 'No journal record exceptions.' More details are provided in the Create Report job step.

```
Report ID:CAM2                                                                PAGE : 1
Run Date: 01-31-2007              Journal Record Exception Report
Run Time: 14:07:26

Code     Dept     ID                        Vers     VL   CL   AL

IET      JTY1     09200600000000000003      1        1    0    1
IET      JTY4     12200600000000000019      1        1    0    1

Total Number of Records: 2
```

- Assurance/Detail Report: This report, when produced in report mode, lists all records that matched selection criteria. When run in update mode, it lists all records selected and cleared along with the journal voucher line that performed the clearing. When no records are found since the last run or that match selection criteria, the report lists one line reading: 'No records found.' More details are provided in the Create Report job step.

```
Report ID: CAM3                                                               PAGE : 1
Run Date: 02-01-2007                  Assurance/Detail Report
Run Time: 10:20:44

In Sync

Journal Voucher
Code     Dept     ID                        LG       AL                Line Amount        Detail Sum
Details
Code     Dept     ID                        Vers     VL   CL   AL        Amount

JVCAM    JTY4     02010700000000000046      1        1                   $175.00            $175.00
IET      JTY1     09200600000000000009      1        1    0    1         $175.00
```

- Bank Summary Report: This report summarizes total dollar amounts for each bank used on the journal vouchers by Fund, Sub Fund, BSA, and SBSA. When it is found that more than one bank is used, this report can be used to perform the cash transfers necessary to bring the actual banks in sync with Advantage Financial. More details are provided in the Create Report job step.

```
Report ID: CAM4                                                               PAGE : 1
Run Date: 01-31-2007                    Bank Summary Report
Run Time: 14:07:26

Bank - Name
Fund          Sub Fund      BSA           SBSA          Amount

010 - Bank of America

JTF1                        CM10          001           $160.00
JTF2                        CM20          001           ($160.00)
JTF4                        CM10          001           $160.00
JTF5                        CM20          001           ($160.00)

Total for Bank 010                                      $0.00
```

Problem Resolution

If any job defined as re-startable in the chain failed due to application errors it is advisable to restart the job after correcting the errors instead of rescheduling the job. Restarting the job will

reduce the processing time since the job will resume from where it last committed and will select only the unprocessed records. Please see the problem resolution section for the individual job step that failed for more information.

In the case of accounts not cleared, the user should check to make sure that all transactions created were loaded and submitted successfully. Data setup problems preventing the loading of journal vouchers should be corrected and the program run again or the failed transaction XML should be loaded from the Import/Export Error directory. Data setup problems preventing the successful submission of journal vouchers loaded should be addressed and the vouchers submitted manually.

When clearing accounts still have a balance remaining even after a run, the program can be run again if balances were from transaction processing after the first run. Clearing the balances manually can be done, but should use the same transaction code created by the clearing process (JVCAM as delivered). If another transaction code of the JV is used, the next run of the clearing process will pick up both sets of entries and clear them, resulting in a net of zero.

When clearing accounts still have balances that did not match selection criteria, the program can be run with a different parameter ID with selection criteria for these accounts.

Clearing Account Maintenance Chain: Clearing Account Maintenance Job

Job Name	Clearing Account Maintenance
Recommended Frequency	Run as part of the chain
Single Instance Required	Yes
Can be restarted?	No
Reports Generated	Assurance/Detail report if run in report mode

Overview

The Clearing Account Maintenance job uses the selection criteria from the CAMP page to select matching Due To and Due From entries on the specified input journal. If run in *Report* mode, after selection, there is only the creation of the Assurance/Detail report. If the run mode is *Update*, then an XML file is created if there are selected records with information on the JVCAM transactions (delivered transaction code for use with this program), to reverse clearing account entries through Cash. The Clearing Account Maintenance Parameters (CAMP) table defines the many functional parameters needed for the Clearing Account Maintenance process. This table provides easy data entry and editing prior to execution of the Clearing Account Maintenance process.

The first task of this job step is to evaluate batch parameters entered for the job step. No validation of the CAMP table values is performed other than the parameter id specified. Any batch parameter not valid will cause the job step to fail with an error message written to the job log stating the problem.

Once parameter validation is completed, record selection is ready to begin. Journal Log (JLOG) is read for a record with a Process ID of CAM that has matching selection criteria to the current run. If no match is found, the program starts at the beginning of the input journal. If a matching record is found, the program begins at the point in the input journal where the last run left off. As a result of this behavior, care should be taken to use consistent CAMP records. It is important to note here that the Selection FY and APD values are not recorded on JLOG for a run. Therefore, if selection is done for APD 12 records while APD 13 records are actively being created, the program will not be able to go back into the range defined for the APD 12 run. If that range has to be searched for APD 13, then the JLOG record created by the APD 12 run has to be deleted. Running the chain for APD 13 will not result in any APD 12 records being selected a second time.

When possible, running the program on the date one APD is closed and before the next begins is the recommended approach.

When records are selected for clearing, an XML file will be created to load the transaction during the second job of this chain job. The job output mode can be either *Detail* or *Summary*. If the output mode is *Detail* then all of the COA elements are written to the JVCAM transaction XML. If the output mode is *Summary* then the journal records are summarized on the Fund and Sub Fund values so that information such as Department, Object, and Appropriation are dropped.

After all records have been processed; the job will change the JLOG status for all selected work records from *Intended* to *Final*. Then the job step will create the Submit file to submit those transactions during the third job of this chain job.

Detail Processing Steps

a. Validate Input Parameters:

   In this step, the job validates the user entered batch parameters.

b. Get Last Processed Record Number:

   In this step the job will search for the record in the Journal Log table for the entry of Process CAM with description value, concatenated string of values of Due To posting code, Due To BSA, Due To SBSA, Due From posting code, Due From BSA, and Due From SBSA parameters. If the record is found, then it selects records which are next to the selected one.

c. Copy Matching Record to Temporary Table:

   In this step the job will copy all of the matching records from the input journal table to the temporary table (that is, R_CLR_ACT_MNT_TMP). After all of the records are copied, then the records are marked for Exception if there is an exception. Then the method will set the Bank Account Codes in the temporary table records based on the Fund Code of those records and the Current Fiscal Year.

   If run in *Report* mode, then selection and processing stops and the Assurance/Detail report is created with Temporary Table information.

d. Process Temporary Table Records and Create XML:

   After retrieving records from the table, it will check the line amount for each record.

   - If the line amount is positive then create JVCAM entries by using the following logic, “If the original entry is Debit – Debit Cash account and Credit Due To or Due From account for the amount on the selected record.”
   - If the line amount is negative then create JVCAM entries by using the following logic, “If the original entry is Credit – Debit Due To or Due From account and Credit Cash account for the amount on the selected record.”

e. After all records have been processed and the XML created, the job changes the JLOG status for all selected work records from Intended to Final.

   - If the XML creation step fails for any reason, the JLOG records will remain marked as *Intended.* This record must be deleted from that table before a new chain is submitted. Any previous XML file created will be deleted and a new one created.

f. A submit text file is now generated at this point for the later job to submit the transactions.

The following table shows the various steps that the Clearing Account Maintenance Chain Job goes through and the progression messages issued at each step.

Process Steps	Messages
Reading Parameters	• "Reading Batch Parameters" • If failed to read input parameters then the following message is displayed on the log:" Failed in reading Batch Parameters from CAMP table" • If all parameters read successfully from CAMP table then the following message is displayed on the log: "Batch Parameters read successfully from CAMP table."
Parameter Validation	• "Validating Batch Parameters." • If input batch parameters are invalid then the invalid value will be displayed in the log along with the error message • If all parameters pass validation the following message is displayed on the log:" Batch Parameters validation completed successfully."
Get Last Processed Record Number	If a JLOG record is found but the Ending Journal Record Number is not found, then the following message is displayed: "Input parameters may cause the selection of records already selected" and the job fails.
Process Journal Records	At least one Due To and one Due From record should be selected for a single accounting line unless the transaction code is JV or uses event categories FD, FIA, or FIB. If required selection not found then message is displayed on the log: "Selection resulted on only a Due To or a Due From for an accounting line. See Journal Record Exception Report."

Restartability Information

If this first job fails, a new chain should be submitted after removing any JLOG updated made by the job that failed. There is no need to back out any other updates when this job fails since this job updates only the temporary table that gets cleared out when a new chain is submitted.

Major Input

- The following attributes are from the Journal Log (JLOG) table after initial run only:
- PROC_ID
- JRNL_NM
- BGN_JRNL_REC_NO
- END_JRNL_REC_NO
- RUN_TYP
- JRNL_ID
- ST_FL
- DSCR

- Clearing Account Maintenance Parameters (CAMP)
- Input Journal
- Clearing Account Maintenance temporary table (R_CLR_ACT_MNT_TMP)

Batch Parameters

Parameter	Description	Default Value
ACTN_CD	Action Code	BATCH
AMSPARM (** **Refer to Note**: Assumptions for SWBP on page no.6)	Directory to put the JVCAMParams.txt parameter file	$$AMSROOT$$/Parms
APPLY_OVERRIDES	Apply Overrides - Required parameter when the run mode is Update to specify if overrides are to be automatically applied. Valid Values are: 1 for Yes and 2 for No.	1
BYPASS_APPROVALS	Bypass Approvals - Required parameter when the run mode is Update, to specify if approval routing is to be bypassed. Valid Values are: 1 for Yes and 2 for No.	1
CLIENT_NAME	Client Name - Optional parameter for a Client Name to appear on all Reports.	No Default
DOC_XML_FILE	Transaction XML File	CAM_XML_FILE.xml
EXCEP_REP_FILE_NM	Exception Report File Name	CAMDocExpRepFlatFile.txt
IMPORT_PARM_FILE	Import Parameter File(.txt)	ImportJVCAMParm.txt
MAX_ACCT_LN	Maximum Accounting Lines - Optional number of accounting lines for generated JV transactions. The lower of this number or the DCREQ limit for JV_DOC_ACTG will be used.	No Default
PARM_FLAT_FILE	Batch Parameter File to pass parameters to later job steps (.txt)	JVCAMParams.txt
PARM_ID	Clearing Account Maintenance Parameter ID, from the CAMP page that should be used for this run. When running the job, you can click on the Custom Parameter link to open the Clearing Account Maintenance Parameter table to select a parameter record or enter a	No Default

	parameter ID.	
REPORT_ID_1	Required ID used to identify the Transaction Exception report for distribution.	CAM1
REPORT_ID_2	Required ID used to identify the Journal Record Exception report for distribution.	CAM2
REPORT_ID_3	Required ID used to identify the Assurance/Detail report for distribution.	CAM3
REPORT_ID_4	Required ID used to identify the Assurance/Detail report for distribution.	CAM4
SUBMIT_PARM_FILE	Submit Parameter File(.txt)	SubmitJVCAM Parm.txt

Major Output

- Journal vouchers transaction XML file (CAM_XML_FILE.xml)
- Journal Log (JRNL_LOG)
- Clearing Account Maintenance Temporary table (R_CLR_ACT_MNT_TMP)

Job Return Codes

The following table shows the potential job return codes for the Clearing Account Maintenance Chain job.

Return Code	Condition
Successful (1)	No problems with parameters and records selected, all processed.
Warning (4)	1. No records found for selection because no new records exist in the journal since the last run or no records met the selection criteria. 2. Records found, but no clearing account activity matched CAMP selection criteria.
Non-Fatal Error (8)	If both a Due To and Due From record is not selected for an accounting line of any transaction type other than JV or FA, then no records for that accounting line will be selected. The job step shall continue but end with a statue of Non-Fatal Error.
Failed (12)	The job will fail under the following conditions: • Batch parameters are invalid • The JVCAMPParams.txt file does not exist. • JLOG entry found for same Due To PSDC, Due To BSA, Due To SBSA, Due From PSCD, Due From BSA, or Due From SBSA parameter on CAMP table, but with a different journal than current run. Would result in

	duplicate record selection. • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated, subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Sequence

- Transaction Code
- Department Code
- Transaction ID
- Transaction Line Group Line Number
- Transaction Accounting Line Number

Selection Criteria

When Run Mode is *Update*, selection of records from the Input Journal is based on the following criteria:

- Posting Amount (PSTNG_AM) greater than zero.
- Transaction Code is not the JV Transaction Code on the CAMP Parameter ID record.
- If these fields are not empty on the CAMP table, then the following fields are added to the selection criteria: Accounting Period, Fiscal Year, and Budget Fiscal Year. Journal field matches are made on PER_DC, FY_DC, and BFY respectively.
- If Due From BSA or Due To BSA is not empty on the CAMP table then BSA code (BSA_CD) of journal record matched the Due From BSA and Due To BSA parameters.
- If Due From Sub BSA or Due To Sub BSA is not empty on the CAMP table then Sub BSA code (SBSA_CD) of journal record matched the Due From BSA and Due To BSA parameters. If the SBSA parameters are left blank, then records are matched just on the BSA parameters so the resulting selection could be records with no SBSA value and ones with various SBSA values.
- Due From Posting Code ID or Due To Posting Code ID from the CAMP table matches Posting Code (PSTNG_CD_ID) of journal record.
- If output mode is 'update,' then Journal Record Number (JRNL_REC_NO) is greater than the start journal record number on the JLOG for the CAM process.

When Run Mode is *Report,* mode selection of records from the Input Journal is the same as in *Update* mode, but selection of the temporary table records uses the following criteria:

- Posting Code Id is not a *Cash* posting code
- Exception flag (EXCEP_FL) is not selected on R_CLR_ACT_MNT_TMP

Problem Resolution

If no records are selected by this job step, first review the selection criteria on CAMP and the records on JACTG that you anticipated being selected. Make any necessary adjustments to your CAMP record and run again. If selection criteria matched, but no records were selected, verify records are in the input data source provided. (Please see earlier comment in the chain job section about proper input sources.) If no records are still selected, please review JLOG records to see if a previous run encompassed records anticipated for selection in a prior run. (Please see earlier comment in the Overview section of this job step concerning APD selection.

Clearing Account Maintenance Chain: Load JV Transactions Job

Job Name	Load JV Transactions
Recommended Frequency	Run as part of the chain
Parallel processing enabled	No. Parallel processing is not supported for this job.
Can the job be restarted?	Optionally, based on the Save Restart Information parameter.
Exception report produced	No. All of the exceptions are only written to the job log.

Overview

The JV Load job loads the records from the JVCAM XML files to load JVCAM transactions generated by the Clearing Account Maintenance job, into the Transaction Catalog. This job uses the common utility to load the records into the Transaction Catalog. This job first validates the batch parameters. If the parameters are valid, then it loads the records into the Transaction Catalog. If the parameters are not valid, the job issues appropriate messages and ends with a status of Failed. Once the records are loaded into the Transaction Catalog, the summary information is written into the log to display the number of records that were in the input file and the number of records loaded successfully.

Major Input

- JVCAM XML file: (CAM_XML_FILE.xml)
- Import Parameter file: (IMPORTJVCAMParm.txt)

Batch Parameters

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

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

Parameter	Default Value
Action Code (ACTN_CD)	171

Commit Block Size (COMMIT_BLOCK)	1
Transaction Status Code (DOC_STA_CD)	2
Apply Overrides (APPLY_OVERRIDES)	True
By Pass Auto Transaction Number (BYPS_ADNT_FL)	True

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

Major Output

- JV Transaction Header, Line Group, and Accounting Line Catalogs with records in draft transaction phase

Job Return Codes

The following table shows the potential job return codes for the Load JV Transactions Batch job.

Return Code	Condition
Successful (1)	All of the records are loaded into the Transaction Catalog successfully or the input file is empty.
Warning (4)	This return code will be issued when some of the records failed to load whereas all other records were loaded successfully.
Non-Fatal Error (8)	This job does not issue this return code
Failed (12)	• Parameters are invalid • The input file is not found in the specified directory • Restart failed because another instance of the Clearing Maintenance chain has already been run successfully • Runtime exceptions encountered for any unexpected situations When the job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Warning (4)	None or not all of the transactions successfully loaded from the XML file	Corrections to that XML file or within the application should be made so that transactions can be loaded. Then the load job can be restarted or the failed transactions can be loaded with a separate SMU, reading the Import/Export Error file created by the failed load.	Optionally, if none of the transactions loaded, the JLOG entry should be removed and the whole chain run again with CAMP values that will result in successfully loaded transactions.
Failed (12)	Failed while restarting the job since another instance of the job has already been run successfully. **Sample Message**: Cannot restart the job since another instance of this job has already been run successfully. Invalid job parameter	Wait until first instance has completed. Fix invalid job parameter.	

Clearing Maintenance Chain: Submit JV Transactions Job

Job Name	Submit JV Transactions
Recommended Frequency	Run as part of the chain
Single Instance Required	No
Can be restarted?	Yes

Report Generated	No. All of the exceptions are written to the error file.

Overview

This job submits the transactions listed in the input parameter file originally generated by the Load JV Transactions Batch job, and performs a submit action on a listing of the transactions previously loaded.

Major Input

- SMU job parameter file (SubmitJVCAMParm.txt)
- Draft JVCAM Transactions in the transaction catalog

Batch Parameters

Parameter	Description	Default Value
PARM_FILE	Parameter file to read	$$AMSPARM$$/SubmitJVCAMParm.txt

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

Major Output

The Transactions would have been processed to final or rejected.

Batch Return Codes

The following table shows the potential job return codes for the individual Submit JV Transactions job.

Return Code	Condition
Successful (1)	Job was able to find parameter file, and transactions were processed. This does not mean that all were accepted. Those not accepted are listed in the job log as well as in the exception report created by the next job step.
Warning (4)	Not Applicable for this job.
Non-Fatal Error (8)	Not Applicable for this job.
Failed (12)	• Submit parameter file is not found. • Restart failed because another instance of the Clearing Account Maintenance chain has already been run successfully. • Runtime exceptions encountered for any unexpected

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

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	None or not all of the transactions successfully submitted to final	Make corrections to setup tables in the application and resubmit the transactions manually or through a separate SMU job.	
Failed (12)	Submit parameter file not found	Correct where the submit step is looking for the file or the file name. The file could also be renamed or moved to where the submit job is looking	Correct default parameter values on Job Setup to prevent this in the future.
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	

Clearing Maintenance Chain: Create Report Job

Job Name	Create Report
Recommended Frequency	Run as part of the chain

Single Instance Required	Yes
Can be restarted?	No
Report Generated	Yes. The job generates 4 reports in PDF and HTML formats. 1. Transaction Exception Report. 2. Journal Record Exception Report. 3. Assurance/Detail Report. 4. Bank Summary Report.

Overview

Transaction Exception Report

This job in the Clearing Account Maintenance chain generates an exception report that lists all of the errors encountered when the JV transaction was submitted in the earlier step. This report will list those JV transactions created that did not reach the *Final* transaction phase. Details will be the transaction code, transaction department, and transaction ID for each that is not final. Upon determination and a remedy for the error(s) performed, the transactions can be manually submitted or submitted through a new system maintenance utility step. The report can then be produced again so that it reads the same parameter file listing the transaction ID's loaded to determine if any are still not *Final*. Alternatively, the Transaction Catalog can be searched for rejected transactions created from the CAM process.

Journal Record Exception Report

This report will list those journal records selected that did not have a matching Due To or Due From journal record. Items listed in this report, unless inconsistent batch parameters were supplied, should be minimal to none. Any exceptions listed would have to be manually cleared. If batch parameters were the cause, then all lines should be listed here. For example, if A002 was listed for the Due To and F005 was listed for the Due From, then all lines found with A002 would not have a match. In such a case, JLOG records created from that run should be deleted and the process re-run with correct parameters.

Assurance/Detail Report

This report will list each journal voucher accounting line created followed by the journal record (detail mode) or journal records (summary mode) that contributed to the accounting line.

When the chain job is run in *Report Only* mode, this report will still be produced, except there will be no information in the Journal Voucher section. There will just be a listing of all journal records selected and the amounts. Report Only will not populate the Detail Sum column.

Bank Summary Report

Bank Summary Report will list total adjustment amount by bank along with the bank transferred from or to. This report will summarize the debit and credit amounts by individual bank codes. The bank information is derived from the accounting lines of the Journal Voucher transactions that went into final status. If the adjustment amount is positive for a given bank, it will be listed under the To Bank field. Similarly, if the adjustment amount is negative, it will be listed under the From Bank field.

This report is provided to enable a user to transfer monies between banks so that Advantage Financial is in sync with those banks. Alternatively, the Bank Account Transfer chain job could be used to do that automatically, given that the Master Bank information has been setup properly for funds involved.

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

Process Steps	Messages
Creating exception report	• Validating batch parameters • Total Number of records selected to write Exception report: 'n' • Number of transactions processed for report: 'n' where n is the progression counter size • Writing records to the report • Processing completed

Major Input

- Draft JV Transactions in the catalog with a Rejected status
- CAMDocExpRepFlatFile.txt
- JVCAMParams.txt
- SubmitJVCAMParm.txt

Batch Parameters

Parameter	Description	Default Value
ACTN_CD	Action Code	REPORT
AMSEXPORT (** **Refer to Note**: Assumptions for SWBP on page no.6)	AMSEXPORT	No Default
AMSPARM (** **Refer to Note**: Assumptions for SWBP on page no.6)	AMSPARM	No Default
AMSROOT (** **Refer to Note**: Assumptions for SWBP on page no.6)	AMSROOT	No Default
EXCEP_REP_FILE_NM	Exception Report File Name	CAMDocExpRepFlatFile.txt
PARM_FLAT_FILE	Batch Parameter File(.txt)	JVCAMParams.txt
SUBMIT_PARM_FILE	Submit Parameter File(.txt)	SubmitJVCAMParm.txt

Sort Sequence

- Transaction Exception Report: Transaction ID
- Journal Record Exception Report: Transaction Information (Code, Dept, Id, Version, VL, CL, AL)
- Assurance/Detail Report:
- Overall Sorting: Transaction Information (JVCAM Code, Dept, ID, LG, AL)
- Details Sorting: Transaction Information (Code, Dept, ID, Version, VL, CL, AL)
- Bank Summary Report:
- Overall Sorting: Bank
- Detail Sorting: Fund, Sub Fund, BSA, SBSA

Selection Criteria

Journal Record Exception Report

The records are selected from the temporary table based on the criteria that the Exception flag is set to true.

Bank Summary Report

The records are selected from the temporary table based on the criteria that the Exception flag is set to true. The selection criteria for getting the Bank name from the BANK table is: Bank Account code on the BANK table must be the same as the temporary table.

Assurance/Detail Report

Selection if from the CAM Temporary table where the posting code is not the Cash CAMP value and the Exception Flag is not true.

Transaction Exception Report

Selection is done as part of SMU where all transactions without a transaction status of Final are selected that were originally in the submit parameter file.

Troubleshooting

The job cannot be restarted if the job ends with a return code of Failed. If the failure occurred, then schedule a new chain with only this job step activated. If another instance of the full chain has completed, then files and data used for the reports will have changed and the previous run can no longer be reported upon.

2.1.15 Contract Roll Process

Agreements with vendors frequently extend beyond the current budget fiscal year. Recording information for all years is an essential part of the procurement function. Lines recorded against future budget years may not have an available budget. These lines will use a non-accounting event type that will not result in any budget updates. As budgets become available during the life of the agreement, the non-accounting event type must be changed to one that will update and reserve appropriate budget amounts. Therefore, one objective of the Contract Roll program is to change non-accounting event types used to record contract terms in future budget fiscal years to accounting event types.

The second objective of the Contract Roll is to un-reserve any accounting lines not specifically locked for a given BFY. An accounting line marked as Reserved Funding is for a future budget fiscal year and is not available to be automatically liquidated by a payment. As the new year's funding becomes available, the line will be "available" for payment. The Contract Roll program will change the Reserved Funding indicator for an accounting line from Yes to No. Lines that have a Reserved Funding value of Locked will not be changed.

The third objective of the Contract Roll is to increment the fiscal year of out year lines. To validate chart-of-accounts, the current fiscal year is always used on both non-accounting and accounting event types in both current budget fiscal year and future budget fiscal years when the original entry is done. The Contract Roll program will increment the fiscal years of out year lines to reflect the new fiscal year. No postings will result from this change alone, as changes to the Fiscal Year and Accounting Period fields on an accounting line alone will cause any new postings.

To accomplish any one of the three objectives, a modification transaction will be created, loaded and submitted. The creation of modification transactions will be based on the selection criteria and job parameters. If the user elects to accomplish all three things at one time, parameters can be set to allow this to happen. Then one chain job can be run to create, load and submit the modification transactions. The user can also set the parameters so that one or two changes can occur separately, at different times. One modification transaction will be created, loaded and submitted each time the chain job is run. The timing of changing the first two objectives can eliminate the need to change the Fiscal Year. For example, changing the event type after the start of the new BFY will automatically change the Fiscal Year and Accounting Period.

Two reports are generated by a third job step by reading the output from the submit job step to list those accounting lines successfully modified and those that failed. For those that failed, a limited number of error messages are also displayed.

The fourth job step will discard those modifications that did not reach Final status. If users are to be allowed the ability to manually correct failed modifications, then this job step should be disabled.

In an application where partial referencing is allowed between requisition and award transactions for future Budget Fiscal Year lines, when the Contract Roll Chain is run to change event types, the referencing transactions should be rolled before the referenced transactions.

To schedule and run the process

The chain process is comprised of the following batch jobs:

- Job 1 – Create Modifications
- Job 2 – Submit Modifications
- Job 3 – Generate Report
- Job 4 – Discard Modification Drafts

Job No	Job	Description	Output
1	Create Modifications	Accounting lines are selected from the Transaction Catalog based on the selection criteria defined by the user.	Modification Transactions are created with the changed values from the parameter page.
2	Submit Modifications	The modification transactions created in the previous job are submitted.	N/A
3	Generate Report	Reports are generated based on the successful/failed submissions of the modification transactions.	Two sets of reports are generated: PDF/HTML reports of Successfully submitted transactions and PDF/HTML reports of Failed transactions.
4	Discard Modification Drafts	The failed transactions are discarded.	Cancellation transactions are created for each of the failed transactions.

Parameters

For the entire chain process, a user has to enter parameters for Job 1 – Create Round 1 only. The process propagates required parameters down the chain jobs. Some of them are non-overrideable parameters, which will be provided with the Job Setup and does not need to be specified by the user; these are meant for system use.

Job	Parameter	Description	Default Value
1	Accounting Event Type (ACCT_EVNT_TYP)	Required parameter for the event type(s) used to replace the non-accounting event types on selected accounting lines. Multiple values allowed if separated by commas.	(typically PR05)
	Apply Overrides (ACTN_OVERRIDE)	Indicate whether to apply overrides to all transactions chosen. (0 = no, 1 = yes)	0
	Period *APD)	Optional parameter for the Period for modified lines instead of the default value.	
	Budget Fiscal Year (BFY)	Required parameter for line selection.	
	Bypass Approval (BYPASS_APRV)	Indicates whether to bypass approvals on all transactions chosen. (0 = no, 1 = yes)	0
	Select Accounting Line Department Code	An optional selection parameter used to select Department codes from the Document Accounting Catalog. Multiple	

	(SADC)	values allowed if separated by commas.	
	Chain Parameter File Name (CHAIN_PARM_FILE)	Required file name for the parameter file used to pass parameters to subsequent job steps from the first job.	Contract_ Parms.txt
	Change Event Type (CHNG_EVNT_TYP)	Indicates event types will change to record an accounting and budget update. (0 = no, 1 = yes)	0
	Change Reserved Funding (CHNG_RES_FND)	Indicates reserve funding will change from *yes* to *no* so that the accounting line will be copied forward. (0 = no, 1 = yes)	0
	Client Name (CLIENT_NM)	Optional field supplying a name to output reports.	
	Commit Size (COMMIT_SIZE)	A performance parameter to control the number of transactions committed in a block.	1
	Transaction Code (DOC_CD)	An optional selection parameter used to select a subset of transaction codes in the RQ, PO, and ABD transaction types. Multiple values allowed if separated by commas.	
	Exception Report File Name (EXCEP_REP_FILE_ NM)	Required file name for the creation of the exception report.	Excep_R ep.txt
	Fiscal Year (FY)	Optional parameter for Fiscal Year for modified lines instead of the default, used when rolling before the start of a new fiscal year.	
	Increment Fiscal Year (INCR_FY)	Indicates that the Fiscal Year parameter is written to modified lines. (0 = no, 1 = yes)	0
	Maximum Prefetch Count (MAX_PREFETCH_C OUNT)	A performance parameter controlling the number of accounting lines selected for processing in one instance.	1000
	Non-Accounting Event Type (NON_ACCT_EVNT_ TYP)	A required selection parameter indicating one or more event types that need to be changed to the Accounting Event Type parameter. Multiple values allowed if separated by commas.	(typically PR08)
	Modification Reason (REAS_MOD)	A parameter used when rolling transactions in the Purchase Order transaction type.	
	Submit File (SUBMIT_FILE)	Required file name for a listing of transactions for the later submit job step.	Contract_ Submit.tx t
	AMS Parameter File Location	Parameter file location for the job step.	-

	(AMSPARM)		
	Log File Location (AMSLOGS)	Logs Location at Create Modifications Job	-
2	Parameter File (PARM_FILE)	Required file name for a listing of transactions to submit along with the location of that file.	$$AMSPARM$$/Contract_Submit.txt
3	Chain Parameter File (CHAIN_PARM_FILE)	Name of the parameter file created in the first job step.	Contract_Parms.txt
	Discard Transaction Listing (DISCARD_DOCS)	When incrementing the Fiscal Year on reserved lines for COA edits, transactions can reject with errors. Those rejections from the submit action are logged in this file.	Discard_File.txt
	Error File Name (ERROR_FILE_NM)	Required file name to log all transaction errors upon submit for the exception report.	Excep_Rep.txt
	AMS Parameter File Location (AMSPARM)	Parameter file location for the job step.	
	AMS Log File Location (AMSLOG)	Log file location for the job step.	
4	Discard Transaction Listing (PARM_FILE)	Required file name for a listing of transactions to discard along with the location of that file.	$$AMSPARM$$/DiscardFile.txt

Define parameter values for the Contract Roll Chain process by first entering values in the Contract Roll Create Modifications Job under General Accounting. Click on the **Setup Custom Parameters** link on the Batch Parameters page.

Major Input

Transaction Accounting Catalog (DOC_ACTG)

Peripheral DataObjects

Event Type (R_EVNT_TYP)

Fiscal Year (R_FY)

Transaction Control (R_GEN_DOC_CTRL)

Accounting Period (R_APD)

Output

Generates modification transactions and reports

Sort Criteria

- Transaction Department Code (DOC_DEPT_CD)
- Transaction Type (DOC_TYP)
- Transactions Code (DOC_CD)
- Transaction ID (DOC_ID)
- Transaction Version Number (DOC_VERS_NO)
- Transaction Vendor Line Number (DOC_VEND_LN_NO)
- Transaction Commodity Line Number (DOC_COMM_LN_NO)
- Transaction Accounting Line Number (DOC_ACTG_LN_NO)

Selection Criteria

For Changing Event Type Only

- Selection criteria for obtaining Transaction Accounting lines (DOC_ACTG) to be rolled is –where
- Transaction Phase = Final and Transaction Function is not = Cancellation and
- Event type in (Event type batch parameter) and
- Budget Fiscal Year = Budget Fiscal Year as specified in batch parameter
- If used, Department = one of the Selection Accounting Line Departments specified as a batch parameter

For Changing Reserved Funding Only

- Selection criteria for obtaining Transaction Accounting lines (DOC_ACTG) to be rolled is –where
- Transaction Phase = Final and Transaction Function is not = Cancellation and
- Reserved Funding = Yes and
- Budget Fiscal Year = Budget Fiscal Year as specified in batch parameter
- If used, Department = one of the Selection Accounting Line Departments specified as a batch parameter

For Incrementing FY Only

- Selection criteria for obtaining Transaction Accounting lines (DOC_ACTG) to be rolled is –where
- Transaction Phase = Final and Transaction Function is not = Cancellation and
- Budget Fiscal Year = Budget Fiscal Year as specified in batch parameter
- If used, Department = one of the Selection Accounting Line Departments specified as a batch parameter

Troubleshooting

- It is a good practice to look at the log of each job for errors even if the job has run successfully.

- If the job log of the discard step issues the following message, ”Invalid action code read form parameter file: all corresponding parameter lines ignored" it means that there were no transactions created that did not submit for the discard job to act upon.
- This batch program produces new Advantage transactions, which are subject to the line limit functionality constraints. Sites should ensure that they run this job with parameters set to ensure that the created transactions are within the line limit controls.

2.1.16 Department Transaction Listing

Job Name	Department Transaction Listing
Recommended Frequency	On demand or scheduled regularly
Single Instance Required	No
Can be restarted?	No
Reports generated	Yes

Overview

The Department Transaction Listing report produces a detailed listing of specified transactions as they appear in the Journals. By specifying the dates and departments as parameters, you can report on transactions processed in a single day for given department(s). The report generated is consolidated by Transaction Type and Department Code.

The following table lists the processing steps of the report.

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. Report Generation	• Rendering Report Started • Rendering Report Completed • Records Processed

Major Input

The input journal specified as a batch parameter, such as Accounting Journal (JATCG / JRNL_ACTG).

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
Client Name (CLIENT_NM)	An optional name to appear in the report header	(No Default)
Selected Departments (DEPT_CD)	A required selection parameter of one or more department codes. Separate multiple values with a comma.	(No Default)
Input Journal (JRNL_LDGR_ID)	The required input journal, such as 1 for the Accounting Journal.	(No Default)

Progression Counter Size (PROG_CTR_SZ)	A required performance parameter that controls job log messaging. 1000 defaults if left blank.	1000
Report Date (RPT_DT)	A required date for journal record selection to provide an 'as of date' of the report. Must be entered in the mm/dd/yyyy format.	(No Default)
Report ID (RPT_ID)	A required ID for the report used for report routing.	(No Default)
Select Block Size (SELECT_BLOCK)	A required performance parameter that controls journal record selection. 1000 defaults if left blank.	1000
Selected Transaction Types (TRAN_TYP)	An optional selection parameter listing one or more of the available transaction types in the report (RE, PO, PR, RQ), separated by commas. If left blank, all four are selected.	(No Default)

Major Output

Department Transaction Listing

Job Return Codes

If this job does not finish successfully, there is no restarting.

Return Code	Condition
Successful (1)	The job ends as successful when all parameters are valid and at least one journal record was selected.
Warning (4)	A warning results when no journal records were found to match selection criteria.
Non-Fatal Error (8)	The report does not use this return code
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 issued when the job is terminated because of database server or network issues.

Sort Criteria

Transaction Type (DOC_TYP)

Selection Criteria

Transaction Type (DOC_TYP), Department (DEPT_CD), and Record Date (REC_DT)

Problem Resolution

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

Step 1: Parameter Validation

Return Code	Condition	Recommendation	Other Instructions
Successful (1)	Parameters were valid and at least 1 record selected.	N/A	N/A
Failed (12)	The job failed due to an invalid or missing required parameter.	Run again with a valid parameter.	N/A
Terminated (16)	The job was terminated manually by the user.	The reason for the termination needs to be addressed before running another instance of the report.	N/A
System Failure (20)	The job was terminated because of database server or network issues.	The reason for the termination needs to be addressed before running another instance of the report.	N/A

Step 2: Report Generation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Report created successfully.	N/A	N/A
Warning (4)	No records were selected	Review selection criteria against journal data using the online inquiry page.	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 runtime exceptions, investigate the exception reported by the process, resolve the error, and restart the job.	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Terminated (16)	The job was terminated manually by the user.	The reason for the termination needs to be addressed before running another instance of the report.	N/A
System Failure (20)	The job was terminated because of database server or network issues.	The reason for the termination needs to be addressed before running another instance of the report.	N/A

2.1.17 Identify and Archive Stale Gaps

When to Run

This process should be run as needed, to maintain only those Gap records on the Journal Log table that may be filled in by transaction processing.

In order to speed up the processing of these jobs, it is recommended that sites frequently run the "Identify and Archive Stale Gaps" process. For example, once a week. No other processes that update the Journal or use the Journal as input should be running when the "Identify and Archive Stale Gaps" process runs. The parameter for "Elapsed Days for Gap Retention" can be set to zero if nothing else is running, or can be set to a higher number if desired.

There are two approaches for running the process:

- Running the process in mode 3 (Archive & Identify Stale Gaps) will first archive stale gaps (previously marked "stale") and then mark newer gaps as "stale". The stale gaps will remain until the next run in mode 3. While gaps are marked "stale", they are read by processes but skipped.
- A slightly better boost to performance may be achieved by running the process in mode 1 (Identify Stale Gaps) and then again in mode 2 (Archive Stale Gaps). This approach will remove the "stale" gap records from JLOG, resulting in fewer gap records that must be read first before they are skipped.

Background

Many batch jobs track their progress though a journal with the Journal Log (JLOG) table. When progression through a journal finds a Gap in expected journal record numbering, it is logged on JLOG for later investigation by a subsequent run.

Records are recorded in journals with three types of unique identifiers. The first of these is the transaction code, department, ID, version, vendor – commodity – accounting - posting line values assigned during transaction processing. The next are assigned by the Journal Engine, whether called real time or in batch mode. The first is a generic journal record number. The second is a combination of generic transaction unique ID and line unique ID. These last two types of identifiers are controlled by records on the Unique Numbers (UNUM) table.

Gaps occur in these numberings when the block size on the UNUM table records are greater than 1. A block size of 1 is not very efficient for processing, so it is very likely to have a size greater than 1. A gap would be created in this simple scenario: UNUM for Journal Record Number has a block size of 10 and a Journal Voucher (JV) is submitted to final with real time journal posting. The application grabbed the next 10 values for Journal Record Numbers, but only used 2 because the JV had only 1 debit and 1 credit line. The remaining 8 numbers were discarded. The next transaction to write to the journal would select a block of 10 starting at where the 1st block of 10 left off, so that a gap of records 3 to 10 now exists.

When more than one instance of an application is running, the occurrence of these gaps is even more frequent and this is where the probability of a Gap being filled in by later transaction processing is most likely. For example, 3 instances are running. Many transactions are processed in instances 1 and 3 first before the first transaction is processed in instance 2. If a program that logs gaps is run before transactions are processed in instance 2, it will log a Gap. When run later after transactions are processed in instance 2, that Gap will be investigated for completion.

Just as the Gap in the simple example with the JV will never be completed because the record numbers are gone, in the instance of multiple instances of an application, if the instance is bounced, all unused record numbers in a block are gone. In either case, the lost range will never be used again unless the UNUM table is changed (something never recommended in this case).

These issues go into the 'when to run' decision of determining if a Gap will ever be filled in, for to stale or archive Gaps that can still be filled in would cause a batch program tracking progress with the JLOG table to skip those journal records.

Description

This batch process identifies and archives stale gaps from the Journal Log table that match selection criteria supplied as batch input parameters.

This process deals with Gap related entries on the Journal Log (JRNL_LOG) table. In general, the process should not be run concurrently with any other processes that can potentially deal with Gap related entries on the Journal Log table. Gap related entries include the following Status Flag (ST_FL) values: Gap Found (5), Gap Intended (6), Gap Final (7), Gap Stale (8), and Gap Failed (9). The following is a list of processes that deal with Gap related entries on the Journal Log. The list is not inclusive and individual run sheets should be consulted to determine if a batch program uses JLOG with Gap logic.

- Ledger Engine
- Rebuild Ledger
- Systems Assurance 3
- Systems Assurance 7
- Systems Assurance 9
- Program Asset Generation
- Reimbursement

Along with this list and any other processes that can potentially deal with Gap related entries on the Journal Log should not be run simultaneously with this process, otherwise it will lead to indeterminate results.

The program can run in one of two modes:

- **Stale Mode:** This process will look for all of the Journal Gaps where the ST_FL = "Gap Found" on the Journal Log and will calculate the # of days the Gaps have been out on the Journal Log by subtracting the PROC_RUN_DT (currently defined as Date/Time but we are interested in the Date part of it only) from the Current System Date. If the result is > "# of Elapsed Day for Gap Retention", then mark the Journal Gaps as "Gap Stale" (ST_FL).
- **Archive mode:** This process selects all Stale Gaps where ST_FL = "Gap Stale". Based on the BGN_DOC_UNID and END_DOC_UNID, it will try to find Journal record(s) within this range. If there are no Journal record(s) within this range, then it will archive the Journal Gaps to an XML file. If there is any Journal record(s) found within the range, it will archive the range(s) where there are no Journal record(s) to the XML file and create Journal Log records with the range(s) of BGN_DOC_UNID and END_DOC_UNID where Journal record(s) exists and mark the ST_FL = "Gap Found".

For example:

- We process a "Stale Gap" with BGN_DOC_UNID = 1000 and END_DOC_UNID = 2000. Within this range, we found Journal records 1050, 1051 and 1090. Thus, this process will create the following entries with ranges of BGN_DOC_UNID/END_DOC_UNID 1000 – 1049, 1052 – 1089 and 1091 – 2000 in the XML file. It will also create new Journal Log records with ranges of BGN_DOC_UNID/END_DOC_UNID 1050 – 1051 and 1090 –

1090 with ST_FL = “Gap Found”. It then deletes the original Stale Gap record from the Journal Log.

- We process a “Stale Gap” with BGN_DOC_UNID = 100 and END_DOC_UNID = 150. Within this range, we did not find any Journal records. Thus, this process will archive this record to the XML file.

- **Archive and Stale mode:** This process will do the Archive logic first before it does the Stale logic.

Major Input

Journal Log (JRNL_LOG)

Other Inputs

Journal Gaps are verified against the JRNL_XXXX tables, where XXXX is a descriptive identifier (for example, JRNL_ACTG is the Accounting Journal and JRNL_CA is the Cost Accounting Journal).

1. Accounting Journal (JRNL_ACTG)
2. Cost Accounting Journal (JRNL_CA)
3. Cash Journal (JRNL_CASH)
4. 1099 Journal (JRNL_1099)
5. Internal Accounting Journal (JRNL_INT)
6. Cross Year Transaction Journal (JRNL_BFYNOTFY)
7. Commodity Journal (JRNL_COMM)
8. Fixed Asset Accounting Journal (JRNL_FA)
9. Fixed Asset Component Journal (JRNL_COMP)
10. Budget Journal (JRNL_BUD)

Output

- Stale Gap Mode **-** Updates the ST_FL flag to “Stale Gap” in the JRNL_LOG table.
- Archive Gap Mode - Stale Gap records in JRNL_LOG table are archived and updated to an XML File.

Parameters

Batch Parameters

Parameter	Description	Default Value
ELAPSED_DAYS_FOR_GAP_RETENTION	Elapsed Days for Gap Retention	

RUN_MODE	Run Mode: (1) - Identify Stale Gaps, (2) - Archive Stale Gaps, (3) - Identify and Archive Stale Gaps	3
XML_FILE_NAME_PREFIX	XML File Name Prefix used to store Archive Stale Gap records	ArchiveStale Gaps

Sort Sequence

- Stale Gap Mode - N/A
- Archive Gap Mode - JRNL_LOG records are selected and sorted by
- Unique Id (UNID)
- Begin Transaction Unique ID (BGD_DOC_UNID)

Selection Criteria

- Stale Gap Mode - Selection criteria to look for all of the Journal Gap records on the JRNL_LOG table is:
- ST_FL flag = GAP_FOUND
- (System Date - PROC_RUN_DT) > Elapsed Days for Gap Retention.

- Archive Gap Mode - Selection criteria for obtaining Stale Gap records from the JRNL_LOG table is:
- ST_FL flag = GAP_STALE

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.
- This batch process has restart ability for the Archive mode. When restarting a job, it uses the same XML file to append the archived gap records.
- It is a good practice to look at the log of the job even if the job has run successfully.

2.1.18 JV Unclaimed Property

Chain Name	JV Unclaimed Property
Recommended Frequency	Periodically or on demand as necessary
Single Instance Required	Yes
Can be restarted?	N/A for chains
Reports generated	N/A for chains

Overview

When using the stale dating feature of the Stale Escheat process and not the Stale Process (that is only a Check Status update), there is often a need that some of the monies of an unclaimed, stale-dated disbursement need to go to an unclaimed property account where others need to go back to the original account used. Which treatment is based on the Unclaimed Property Classification on Fund Group (FGRP) or Fund (FUND). There is a batch parameter that controls which is read. The setting on Fund Group is the recommended approach as it can be set once and then new fund codes will use the setting of the fund group rollup code. However, if the alignment of existing fund groups does not match the needs of unclaimed property, the setting at fund code has to be used. In that case, it is recommended that a Configurable Validation be setup to require the field be either *Unclaimed Property* or *Return to Fund*. The most common type of funds that would be *Return to Fund* would be federal funds or other type of fund that is not local funding which would be *Unclaimed Property.*

While the concept of unclaimed property can exist for an implementation using the disbursement method of *Check* or *Clearing Fund Warrant,* the JV Unclaimed Property process was designed to work with only the *Standard Warrant* disbursement method that posts as follows:

Debit D008 – Stale Warrants Payable
Credit D101 – Stale Payable

A journal voucher is created for each Automatic Disbursement (AD), Electronic Funds Transfer (EFT), or Manual Disbursement (MD) that was referenced by the Disbursement Reclassification (DC) transaction. The Extended Description on the header calls out the disbursement by Transaction ID and each accounting line contains a reference to the disbursement accounting line referenced.

Unclaimed Property*:* When this is the setting for the fund for the selected journal record, the process reverses the Stale Payable DC posting and balances with a Cash posting using all the COA found on that journal record plus the addition of a Cash balance sheet account provided as a batch parameter. A second posting reverses the previous cash posting and updates unclaimed property.

Debit D101 – Stale Payable
Credit A001 – Cash

Debit A001 – Cash with cash accounting template parameter
Credit A015 – Generic Liability with due to accounting template parameter

Return to Fund: When this is the setting for the fund for the selected journal record, the process creates the following postings using a mixture of two accounting template parameters (established with same fund/sub fund codes) and COA from the journal records.

Debit D101 – Stale Payable
Credit D014 – Cash Expenditure with journal record COA*

** When the disbursement was out of a revenue account then instead of D014 the R003 posting code is used. If the disbursement was out of an asset account then A016 is used. If the disbursement was out of an equity account then A014 is used. If the disbursement was out of a liability account, then A015 is used.*

Please note that all posting codes in the section above are baseline codes that are either delivered for the DC transaction or as default values for various parameters in this JV Unclaimed Property process. Furthermore, they reflect postings from a positive disbursement accounting line (not a credit memo). They may be replaced with any customized codes as desired of a similar type.

Also note, this is not the end of the accounting required as the Due To account would have to be settled between the fund of the template and the fund of the disbursement. Also, Stale Payable would also need to be reclassified after the stale disbursement is deemed cancelled.

This chain has the following jobs:

- **JV Unclaimed Batch**: Creates the journal voucher XML file
- **JV Unclaimed Multi-threaded Transaction Loader**: Breaks up the XML into smaller files and processes each with a spawned SMU job

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

Important Run Instructions

1. The JV Unclaimed Process should be run after Stale Escheat has processed outstanding disbursements. Never should the two be run simultaneously.
2. When making balance sheet payments, this process assumes that there will be no object or revenue code supplied.
3. It is imperative that all fund codes used on disbursements have a classification directly or through the rollup. If the process cannot determine a treatment for a fund, the process will fail early in processing and not make any updates and records a message to the job log: FGRP/FUND Code is not found - Doc Code: *, Doc Department Code: *, Doc ID: *, Doc Accounting Line No: *, Doc Posting Ln No: *. This allows one to locate the fund without the classification.
4. If stale warrants have been manually reclassified in the past, before using this automated process, a record has to be inserted into Journal Log (JRNL_LOG) to prevent selecting journal records that have already been manually processed. That record would be as follows:
   - Run Date (PROC_RUN_DT) – Today's Date
   - Process ID (PROC_ID) – UCPOST
   - Journal/Ledger Name (JRNL_ID) – JRNL_ACTG
   - Description (DSCR) – Initial record
   - Run Type (RUN_TYP) – Initial (1)
   - Status (ST_FL) – Final (2)
   - Begin Source Journal Record Number (BGN_JRNL_REC_NO) – 1
   - End Source Journal Record Number (BGN_JRNL_REC_NO) – (last stale journal record accounted for manually)

1. Although not likely in a production environment, if the process is run with invalid selection parameters against a new set of journal records, no records will be selected. The job log will state no records were selected, however the Journal Log record tracking progress through the Accounting Journal will be updated. In order to automatically process that block of records, the Journal Log record should be reset so the End Source Journal Record Number equals the Begin Source Journal Record – 1 and the Begin Source Journal Record is 2 less than it is currently.

Major Input

- Accounting Journal (JACTG / JRNL_ACTG)
- One of the two depending on parameter
  - Fund (FUND / R_FUND)
  - Fund Group (FGRP / R_FGRP)

Major Output

- Journal Vouchers

Chain Return Codes

When an active job step in the chain ends with a Return Code of anything other than successful, any remaining job steps have a Run Status of *Inactive.* Please see documentation on individual job steps for possible Return Codes for those steps, as *Warning* and *Non-Fatal Error* are not always possible outcomes.

Problem Resolution

There are different procedures if either job in the chain fails or one of the spawned SMU jobs fail. Please see later sections in this run sheet for more details.

JV Unclaimed Batch

Job Name	JV Unclaimed Batch
Recommended Frequency	N/A for job steps
Single Instance Required	N/A for job steps
Can be restarted?	No
Reports generated	No

Overview

This job step reads the Accounting Journal to create journal vouchers in an XML output file to record either or both the Unclaimed Property or Return to Fund accounting entries.

Process Steps	Messages
1. Parameter	• Run Started

Validation	• Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log. • JV Unclaimed Batch Process Started • JV Unclaimed Batch Process Stale Dated Warrant Vouchers Started • Found matching record in AUTO_DOC_NO table from values in GEN_DOC_CD/GEN_DOC_DEPT_CD/GEN_DOC_PFX • Found matching record in R_ACTG_TMPL table from value in ACCT_CASH_DOR_TEMP • Found matching record in R_ACTG_TMPL table from value in ACCT_DUE_DOR_TEMP • All parameters are validated
3. Record Selection	• JV Unclaimed Batch Get Last Record ID Processed Started • JV Unclaimed Property Batch started processing records from Record ID: ##### • Case by Case processing of all selected records • Max Record Id from JRNL_ACTG: ##### • JV Export Completed - Last Processed Record: ##### • JV Batch - JVC Process Completed
4. XML File Creation	• JV Unclaimed Batch Process Release Resource Started • JV Unclaimed Batch - JVC Process Completed Successfully • Run Ended

Major Input

- Journal Log (JLOG / JRNL_LOG)
- Accounting Journal (JACTG / JRNL_ACTG)
- One of the following
  - Fund (FUND / R_FUND)
  - Fund Group (FGRP / R_FGRP)

Batch Parameters

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

Parameter	Description	Default Value
Export Location (AMSEXPORT)	The required location where XML file is written.	$$AMSROOT$$/ExportImport
Parameter Location (AMSPARM)	The required location where the parameter file for the next job step is written.	$$AMSROOT$$/Parms

JV XML File (GEN_EXPORT_FILE)	The required file name of the XML file of journal vouchers.	UnclaimedPropertyVouchers.xml
Commit Block Size (COMMIT_BLOCK)	A required performance parameter that determines the number of transactions committed to XML at a time. If left blank or an invalid value is given, the system defaults to 100.	100
Select Block Size (SELECT_BLOCK)	A required performance parameter that determines the number of journal records selected at a time. If left blank or an invalid value is given, the system defaults to 1000.	1000
Progression Counter (PROG_CTR_SZ)	A required performance parameter that determines the number of journal records processed before giving a job log status message indicating progress. If left blank or an invalid value is given, the system defaults to 5000.	5000
Unclaimed Property Classification Source (FUND_SOURCE)	Required indication of what funds should be selected. Valid values are FGRP (Fund Group) or FUND. If blank or invalid, defaults to FGRP.	FGRP
ADNT Fiscal Year (FY)	An optional year used in locating the required Automatic Transaction Numbering record. If left blank, the current fiscal year is used.	No Default
Journal Voucher Transaction Code (GEN_DOC_CD)	A required transaction code used when creating the transfer transactions. The JVC or one similar is recommended if using FES so the funding COA are displayed.	JVC
Journal Voucher Department (GEN_DOC_DEPT_CD)	A required department used when creating journal vouchers.	No Default
Journal Voucher Unit (GEN_DOC_UNIT_CD)	An optional unit used when creating the transfer transactions for security.	No Default
Journal Voucher Prefix (GEN_DOC_PFX)	An optional transaction code prefix to use when creating journal vouchers for identification purposes.	No Default
Stale Payable Posting Code (PSTNG_CD_ID)	The required output posting code for stale payable disbursement account.	D101

Stale Dating Transaction (STALE_DOC_CD)	A required transaction code used to select stale payable records.	DC
Cash Accounting Template (ACCT_CASH_DOR_TEMP)	The required accounting template for the cash debit when Return to Fund.	No Default
Due To Accounting Template (ACCT_DUE_DOR_TEMP)	The required accounting template for due to credit when Return to Fund.	No Default
Stale Payable BSA (STALE_PAYABLE_BSA)	The required BSA for selection and output that is the default account for the D101 Posting Code.	No Default
Warrants Payable BSA Code (EXCLD_WRNT_BSA_CD)	The required BSA used as an exclusion parameter to locate the balance sheet account from a disbursement when Return to Fund.	No Default
Expenditure Posting Code (EXP_POST_CD)	A required output parameter for expenditure accounting lines when disbursements use an object and Return to Fund.	D014
Revenue Posting Code (REVENUE_POST_CD)	A required output parameter for revenue accounting lines when disbursements use a revenue source and Return to Fund.	R003
Asset Posting Code (ASSET_POST_CD)	A required output parameter for asset accounting liens when disbursements use an asset BSA and Return to Fund.	A016
Liability Posting Code (LIABILITY_POST_CD)	A required output parameter for liability accounting lines when disbursements use a liability BSA and Return to Fund.	A015
Equity Posting Code (EQUITY_POST_CD)	A required output parameter for equity accounting lines when disbursements use an equity BSA and Return to Fund.	A014
Cash Posting Code (CASH_POST_CD)	A required output parameter for cash accounting lines for both Unclaimed Property and Return to Fund.	A001
Cash BSA (CASH_BSA)	A required output parameter for cash accounting lines.	No Default

Major Output

- UnclaimedPropertyVouchers.xml

- Journal Log (JLOG / JRNL_LOG)

Job Return Codes

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

Return Code	Condition
Successful (1)	The job ends as successful when parameters are valid and at least one DC journal record was selected and processed. The job also ends as successful when parameters are valid and there are journal records but none match selection criteria as the Stale Escheat process has not been run.
Warning (4)	A warning results when the process does not find any new journal records since the last run or it did but none matched selection criteria.
Non-Fatal Error (8)	This job does not use this return code.
Failed (12)	The job fails under the following conditions: • Parameters are invalid • Unable to find the Unclaimed Property Classification for a fund or fund group • Run time exceptions for unexpected situations. When this job fails, subsequent jobs in the chain are set to *Inactive*.
Terminated (16)	The job is terminated by the user. When this job is terminated, subsequent jobs in the chain are set to *Inactive*.
System Failure (20)	A system failure is issued when the job is terminated because of database server or network issues. When this job encounters a system failure, subsequent jobs in the chain are set to *Inactive*.

Sort Criteria

Journal vouchers are created in the order that disbursement reclassification records are listed in the Accounting Journal.

Selection Criteria

This job step reads the Accounting Journal from here the last run left off by looking for the Process ID (PROC_ID) of *UCPOST* on Journal Log (JRNL_LOG). New records that have a balance sheet account (BSA_CD) that match the Stale Payable BSA parameter with a transaction code (DOC_CD) that match the Stale Transaction Code parameter are selected. For each of those records, the process determines if the fund (FUND_CD) or fund group (R_FGRP) has an Unclaimed Property Classification (UC_FUND_GRP_ST) of Unclaimed Property (1) or Return to Fund (2).

Problem Resolution

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)	Parameters were valid, ANDT record was found, accounting templates were valid	N/A	N/A
Warning (4)	This job step does not end with this return code.	N/A	N/A
Non-Fatal Error (8)	This job step does not end with this return code.	N/A	N/A
Failed (12)	The job failed due to a fatal condition such as invalid parameter, no ADNT match, or invalid accounting templates. **Sample Message:** "XYZ is invalid"	Submit another instance of the chain with correct and valid parameters.	N/A
Terminated (16)	The job was terminated manually by the user.	The reason for the termination needs to be addressed. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.
System Failure (20)	The job was terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.

Record Selection

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	At least one record was found and processed successfully into a journal voucher	N/A	N/A

Warning (4)	Job ended with a Warning because: • No new JRNL_ACTG records for STALE_DOC_CD value of 'DC' were found to be processed.	Only run the process after the Stale Escheat process has run and transactions from it have posted.	N/A
Non-Fatal Error (8)	This job step does not end with this return code.	N/A	N/A
Failed (12)	Job failed due to Fatal conditions: • Unclaimed Property Classification not found for a fund or fund group	Ensure all Fund Group records or Fund records have a classification that were identified by using the DC specified in the job log. Then submit another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be addressed. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.
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. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.

XML File Creation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	XML file created and saved to the Export Location	N/A	N/A
Warning (4)	This job step does not end with this return code.	N/A	N/A
Non-Fatal Error (8)	This job step does not end with this return code.	N/A	N/A

Failed (12)	This job step does not end with this return code.	N/A	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be addressed. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.
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. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.

JV Unclaimed Multi-threaded Transaction Loader

Job Name	JV Unclaimed Multi-threaded Transaction Loader
Recommended Frequency	N/A for job steps
Single Instance Required	N/A for job steps
Can be restarted?	No
Reports generated	No

Overview

This job takes the XML file created in the previous job step, breaks the file into smaller files, and then spawns multiple System Maintenance Utility jobs to load and submit the transactions. For example: If an input file has 31 transactions and the Block Size parameter is 5 and the Thread Count parameter is 3, then 3 split files are created and the first 5 transactions from the input file are written to the first split file, and the next 5 written to the next, etc. When complete, the first file contains 11 transactions. The second and third files contain 10 each. The interim processing was the first five records to file 1, the next five to file 2, the next five to file 3, the next five to file 1 again, etc. The final record was written to file 1 resulting in eleven.

The Multi-threaded Transaction Loader 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 as controlled by the Mode and Transaction Status parameters.

- **Import** - Split and import with a Phase of *Draft* and Status of *Held*.
- **Import & Submit** – Split, import, and submit. This is the intended mode for the Treasurer Interface Process.
- **Import & Other Transaction Sub Action** – This mode is not applicable to the Treasurer Interface Process.
- **Import & Validate** – Split, import, and validate resulting in a Status of *Ready* or *Rejected.*

The run sheet for "Multi-threaded Transaction Loader" in the *Utilities Run Sheets Guide* should be consulted for details including: Processing Steps, Return Codes, and Problem Resolutions. Each of these is common and not unique by use of this system feature with unclaimed property.

Major Input

- UnclaimedPropertyVouchers.xml

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
Block Size (BLOCK_SIZE)	Required number of transactions written to a split file before creating a new file.	1
Commit Block Size (COMMIT_BLOCK_SIZE)	Required block size to control the number of records committed in an instance.	1
File Input Directory (FILE_INPUT_DIR)	Required directory where the XML is located.	$$AMSROOT$$/ExportImport
File List (FILE_LIST)	Required file name or names (comma separated) to be uploaded.	UnclaimedPropertyVouchers.xml
File Output Directory (FILE_OUTPUT_DIR)	Required directory to write the file segments created for processing.	$$AMSROOT$$/ExportImport
File Polling Interval (FILE_POLLING_INTERVAL)	Required polling interval, in seconds, that the system will wait to select an output file.	1
File Polling Maximum Time (FILE_POLLING_MAX_TIME)	Required maximum number of seconds in polling.	2
File Prefix (FILE_PREFIX)	Optional prefix used on the filenames of the output file segments.	
Bypass Approvals (I_SMU_BYPASS_APPROVAL)	Required indication if approvals should be bypassed in the transaction loaded. (TRUE, FALSE)	TRUE
Log Status Interval (LOG_STATUS_INTERVAL)	Required logging frequency (in seconds) for controller thread reporting status of child threads to the system log.	300
Mode (MODE)	Required mode of operation. (1=Import, 2=Import and Submit, 3=Import and Other Action).	2
Other Transaction Action (OTHER_ACTION)	Conditionally required other action supplied to occur before submit. Parameter not applicable to this	(No Default)

	chain.	
Sleep Interval (SLEEP_INTERVAL)	Required sleep interval in seconds for internal controller thread for checking child processes.	5
SMU Catalog ID (SMU_CTLG_ID)	Required ID of the System Maintenance Utility job spawned as the child process.	3
Stagger Time (STAGGER_TIME)	Require lag time, in seconds, between the spawning of each child process.	30
Transaction Status (S_SMU_DOC_STA_CD)	Required status for loaded Transactions (2-Ready, 1-Held).	1
Exception Report File Name (S_SMU_EXCEP_REP_FILE_NM)	Required exception file name for recording submit failures.	(No Default)
Exception Report Detail Level (S_SMU_EXCEP_REP_IND)	Required level of detail for exception reporting (1- Detailed, 2-Failed Transaction, 3- Processed Transaction, 4-Failed Transaction Lines, 5- Transaction Status).	2
Thread Count (THREAD_COUNT)	Required number of threads to use for processing.	8

Major Output

- Journal Vouchers

Sort Criteria

Not Applicable

Selection Criteria

Not Applicable

2.1.19 Journal Engine

Chain or Job Name	Journal Engine
Recommended Frequency	Nightly at a minimum (please see below for more details)
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

The Journal Engine has an online and a batch version which create one or more journal records from a posting line. The online mode happens when a transaction is submitted to Final if the Application Parameter (APPCTRL) of Real Time Journal Posting (RLTM_JRNL_PSTG) is set to True and the Synchronous/Asynchronous Posting setting for a transaction code on Transaction Control (DCTRL) is set to Synchronous. Both versions of the engine determine if a posting line should post to a journal and which ones in the same way. Settings on Posting Code (PSCD) denote posting codes that go to the Accounting, 1099 Reporting, Cash, and Fixed Asset Accounting Journals. Data conditions on a posting line denote if a posting line is to update the BFY Not FY, Internal, and Cost Accounting Journals. The triggers are the BFY/FY values not equaling, an Internal Fund code, and a Program code respectively. If any of these journals is inactivated on Journal/Ledger Control (JLCTRL), the application will not post to that journal. The Commodity, Budget, and Fixed Asset Component Journals are not updated from posting lines, and are thus not updated by the Journal Engine.

Note: The PSCD settings for journal posting are placed into system memory so changes will require a bounce of the application. Changes will likely require a rebuild of the journal in question with the Rebuild Journal job under the Posting/Batch Jobs folder in the Batch Catalog.

Two records on APPCTRL are also used when updating the 1099 Reporting Journal. If the Require Vendor on 1099 Journal Updates (REQ_VEND_1099_JRNL_UPD) parameter is blank, records without a vendor/customer code may post to the 1099 Reporting Journal. If set to True, then only records with a vendor/customer code will post to the 1099 Reporting Journal. If the Require TIN for Miscellaneous Vendors on 1099 Journal Updates (REQ_TIN_1099_JRNL_UPD) parameter is blank, then entries for miscellaneous vendors without a TIN may post to the 1099 Journal. If the REQ_TIN_1099_JRNL_UPD parameter is True, then only records for miscellaneous vendors with a TIN will post to the 1099 Journal. The Journal Posting Engine will evaluate both conditions to determine if the record can be added to the Journal. In order for the record to be added to the Journal both parameter conditions must be satisfied.

Recommended Frequency

- Running of the Journal Engine is recommended nightly before the running of the Ledger Engine. This is true even if real time journal posting is done and no transaction codes are set to *Asynchronous.* Running is recommended nightly to pick up any posting lines that may have not posted real time, which is a very rare situation. If not performing real time posting, running the engine periodically throughout the day is recommended to reduce the workload at night.
- The Journal Engine uses an indication at the posting line that is not visible online to determine eligibility for posting. The JRNL_PSTNG_IND field has three values:

- **Not Ready (1)** – A posting line that is not ready for selection by the Journal Engine. The indication is most common for transactions that have not processed to *Final* yet. The indication is also found for *Final* and *Historical Final* transactions if DCTRL has the *Asynchronous* setting and the Journal Posting Initiator batch job under Accounts Payable/Batch folder in the Batch Catalog has not run. The following SQL will find any such posting lines: Select * from PSTNG_LN_CAT where DOC_PHASE_CD in (3, 5) and JRNL_PSTNG_IND = 1.
- **Ready (2)** – A posting line is ready for selection by the Journal Engine. The indication most commonly found with real time posting off or after a running of the Journal Posting Initiator batch job. The following SQL will find any such posting lines: Select * from PSTNG_LN_CAT where DOC_PHASE_CD in (3, 5) and JRNL_PSTNG_IND = 2.
- **Posted (3)** – A posting line has been written to all applicable journals

Process Steps	Messages
1. Parameter Validation	• Each parameter completed is listed • Any invalid parameters are issued with an error message
2. Processing	• Journalizing: Transaction Dept/Transaction Code/Transaction ID/Transaction Vers No • If no lines are journalized then no transactions are listed

Major Input

- Posting Code (R_PSCD) – Flags for Accounting, 1099 Reporting, Cash, and Fixed Asset Journal posting are used.
- Journal/Ledger Control (R_JRNL_LDGR_CTRL) – Provides a list of active journals and their types of posting line information.
- Posting Line Catalog (PSTNG_LN_CAT) – Provides posting lines from which journal records are generated.
- Application Parameters (IN_APP_CTRL) – Provides parameters for exclusive selection of transaction postings to the 1099 Reporting Journal: REQ_TIN_1099_JRNL_UPD and REQ_VEND_1099_JRNL_UPD.

Major Output

- Journal/Ledger Control (JLCTRL) records specify the names of the source journal tables. These are typically named JRNL_xxxx, where xxxx is a descriptive identifier (for example, JRNL_ACTG is the Accounting Journal and JRNL_CA is the Cost Accounting Journal). Potential output journals (in baseline CGI Advantage Financial) include:
- Accounting (JRNL_ACTG)
- Cost Accounting (JRNL_CA)
- Cash (JRNL_CASH)
- 1099 Reporting (JRNL_1099)
- Internal (JRNL_INT)
- Cross-year (JRNL_BFYNOTFY)

- Fixed Asset Accounting (JRNL_FA)

Batch Parameters

Parameter	Description	Default Values
Transaction Code (DOC CODE)	Optional field used when journal posting is to be done for a single transaction. When specified the Transaction Department, ID, and Version must also be specified.	No default
Transaction Department Code (DOC DEPARTMENT CODE)	Optional field used when journal posting is to be done for a single transaction. When specified the Transaction Code, ID, and Version must also be specified.	No default
Transaction ID (DOC ID)	Optional field used when journal posting is to be done for a single transaction. When specified the Transaction Code, Department, and Version must also be specified.	No default
Transaction Version Number (DOC VERSION NUMBER)	Optional field used when journal posting is to be done for a single transaction. When specified the Transaction Code, Department, and ID must also be specified.	No default
Select Block Size (SELECT_BLOCK_SIZE) Should be a positive integer	Required field. Select Block Size indicates the size of the block of records written to a Journal.	1000

To journalize a single specific transaction, you specify the transaction's keys using the Transaction Code, Transaction Department Code, Transaction ID, and Transaction Version Number parameters. Otherwise, these may be left blank.

Chain / Job Return Code

Return Code	Condition
Successful (1)	All of the selected records are processed successfully or no records were selected for processing.
Warning (4)	• The job does not end in warning. • Note: It does not end in warning if no records are found to process.
Non-Fatal Error (8)	The job does not end in non-fatal error.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid or incomplete transaction key is entered • 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

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

Selection Criteria

The Journal Posting Engine selects all the transactions from posting line catalog with JRNL_PSTNG_IND as 2 (Ready to Post).

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 Error (8)	N/A	N/A	N/A
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following two conditions.	N/A

		1) Encounters any runtime exceptions and 2) Parameters are invalid. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and schedule a new job. If the job fails because of the invalid parameters, verify the logs for invalid parameters, and submit a new job with correct parameters.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Schedule a new job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. Schedule a new job.	N/A

If it is found that a transaction has been partially journalized, meaning that not all posting lines have been marked as *Posted,* something revealed by the System Assurance 6 job, updates to the Posting Line Catalog are going to have to be made because the Journal Engine has logic to post an entire transaction or nothing. A SQL statement to change the DOC_DEPT_CD value on those posting lines posted successfully is required. Then the Journal Engine is run with transaction parameters completed, using the original transaction department code so only the previously un-posted records will be selected. After posting, a SQL statement should be run to set the DOC_DEPT_CD back to what it was.

2.1.20 Journal Rebuild

Job Name	Journal Rebuild
Recommended Frequency	Frequency depends on need. The process can be run for any of the following reasons: • Table-driven options initially set are changed later leading to an incorrect or an incomplete journal. • An inactive journal can be activated and needs to be retro-filled to include past data. • In an event of data corruption on a journal, a part or the entire journal needs to be rebuilt from the posting lines.
Single Instance Required	Yes. A single instance is required for a journal. Multiple instances are allowed if against different journals.
Can be Restarted?	Yes, restarting is available in most instances. Please see the Problem Resolution section.
Reports Generated	None

Overview

The Journal Rebuild process rebuilds part or all of a financial journal from historical details resulting from processed transactions previously posted to the journal. The term 'financial journal' applies to those built from posting lines: Accounting, 1099, Cost Accounting, Internal, BFY Not FY, and Cash. The rebuild process does not address special-purpose journals: Commodity, Budget, and Fixed Asset Component Journals.

Selection and posting logic in the Journal Rebuild process is identical to that of the Journal Engine (real-time or in batch) with only additional selection logic that allows for a partial rebuild based on Budget Fiscal Year (BFY), Fiscal Year (FY), Calendar Year (for 1099 journals only), or a time range defined by Run Time Date. Partial rebuilds can be combined into multiple runs by breaking down the rebuilding work into manageable units of work. Such a series of partial rebuilds will essentially perform a full rebuild when a full rebuild would require more time than available for a single, available time period.

The rebuilding of a journal is a single thread unlike a ledger rebuild where multiple instances of the Ledger Engine can be used to rebuild a ledger. Difficulties with the tracking of activity in the Posting Line Catalog does not allow for multiple instances of the Journal Rebuild to run concurrently. When there are time constraints, the optional selection criteria available for a partial rebuild must be used. Using SQL queries against the Posting Line Catalog with the various selection parameters available to the Journal Rebuild will help in determining which parameter would be best to evenly spread out the work load. The following three sample queries for the Accounting Journal find three different types of posting lines.

1. Select count (*) *2 recount from PSTNG_LN_CAT where DOC_PHASE_CD in (3,5) and JRNL_PSTNG_IND = 3 AND PSTNG_CD_ID IN (SELECT PSCD_ID FROM R_PSCD WHERE ACTG_TYP_JRNL_FL =1) and OPSTNG_CD_ID is not null and PSTNG_AM <> 0
2. Select count (*) recount from PSTNG_LN_CAT where DOC_PHASE_CD in (3,5) and JRNL_PSTNG_IND = 3 AND PSTNG_CD_ID IN (SELECT PSCD_ID FROM R_PSCD WHERE ACTG_TYP_JRNL_FL =1) and OPSTNG_CD_ID is null and PSTNG_AM <> 0

3. Select count (*) recount from PSTNG_LN_CAT where DOC_PHASE_CD in (3,5) and JRNL_PSTNG_IND = 3 AND OPSTNG_CD_ID IN (SELECT PSCD_ID FROM R_PSCD WHERE ACTG_TYP_JRNL_FL =1) and PSTNG_CD_ID is null and PSTNG_AM <> 0

The above can be adjusted to use BFY, FY, Calendar Year (CY, for 1099 journals only) or Run Time Date in selection as well as changed to select for other journals by replacing the (SELECT PSCD_ID FROM R_PSCD WHERE ACTG_TYP_JRNL_FL =1) with the correct criteria:

Cost Accounting Journal: PROG_CD is not null

BFY Not FY Journal: FY_DC <> BFY

Internal Journal: IG_FUND_CD is not null

Cash Journal: (SELECT PSCD_ID FROM R_PSCD WHERE CASH_TYP_JRNL_FL =1)

1099 Journal: (SELECT PSCD_ID FROM R_PSCD WHERE JRNL_TYP_1099_RPT =1)

The concept of rebuilding a journal is one not to be taken lightly in a prototyping environment and to be taken even more seriously in a production environment. Complicating factors in almost any journal rebuild arise from batch programs that use the journal as a source of input. Journals that are only for reporting or on-line viewing do not have these complications.

Foremost among the batch programs that interact with journals is the Ledger Engine. Any journal rebuilt that is the input source to a ledger will require the rebuilding of that ledger as well. Other programs read a journal as input to create transactions from activity recorded in the journal since the last run. Such programs present special challenges that are spelled out in the next section.

Important Run Instructions

Several important facts should be known before scheduling a journal rebuild:

1. Any journal rebuild requires that the RLTM_JRNL_PSTG parameter on the Application Parameters (APPCTRL) page must be set to ‘false’. A setting of ‘true’ is most common so that transactions post real-time to journals. Note: Remember to change it back to ‘true’ when the rebuild is done.
2. When rebuilding a journal that has been previously inactive, after checking the Active flag on the Journal Ledger Control (JLCTRL) page, each VLS must be bounced for the change to be registered.
3. When rebuilding a journal (full or partial), any transactions that have been archived will need to be restored in order to rebuild the journal with the same breadth of data, else the journal will be rebuilt with only un-archived transaction data. Un-archiving is not a likely concern when only rebuilding the current year and possibly a prior year that has not been closed as transaction archiving will only be allowed for older, closed fiscal years.
4. The following procedures are strongly recommended to free up system resources and prevent undesired consequences for full and partial rebuilds.
    a. Bring down every VLS except the one on which the Journal Rebuild job will be scheduled.
    b. Journal Rebuild must be the only batch job running.
    c. There should not be any online user logged into the system.
5. If performing partial rebuilds based on Budget Fiscal Year, do not forget to have a run for BFY = 9999 if that value appears on accounting transactions.
6. If the Accounting or Cost Accounting Journals are rebuilt and infoAdvantage is used, then the data warehouse will be impacted. If either is rebuilt in full, then the infoAdvantage journal containing both CGI Financial journals should be cleared and rebuilt from scratch. When performing partial rebuilds on either CGI Financial journal, since the rebuild journal record numbers will not start over but continue with the previous numbering schedule,

infoAdvantage will pick up these new records in the ETL. The records in the infoAdvantage journal for the fiscal year associated with the rebuilt year must be deleted before the incremental load is run. To bring in the rebuilt records before deleting the older records would present difficulties separating the new from the old.

7. There will likely be at least one interaction with another CGI Financial batch program that uses the rebuilt journal. When a full journal rebuild is done, the application will see that the first new record created in the journal is record #1. A partial rebuild cannot do this because of duplicate record number problems. Whether numbering starts at 1 or continues with the next number that was available at the time of the rebuild, those batch programs that track the progress through a journal with a Journal Log (JLOG) record will be impacted. Note: none of the following applies to a journal that is being activated and brought up-to-date.

    a. The last JLOG tracking record created will no longer contain the appropriate journal record number for the process to start with on the next run.

    b. In the case of a journal rebuilt because of incorrect or changed table driven options, a batch program reading the journal would not have had the chance to select missing records, or would have selected records that should not have been in the journal, or would not have been impacted because the incorrect or changed option did not apply to the batch job. No batch program can go back through the rebuilt journal to pick up records initially missed or create transactions to remove activity previously selected that is no longer in the rebuilt journal. An analysis will have to be done manually and a decision made about manually creating transactions for missed activity. For records selected that should not have been, that too will have to be a manual analysis with a decision made about creating transactions to correct such selected records.

    c. The list of the baseline jobs (there may be custom jobs as well) is given below followed by a description of what needs to be considered and done for the job. The JLOG Process ID is given parenthesis for any JLOG record created for tracking. None of the following will apply to a journal that is being activated and brought up-to-date. These are just the jobs that involve only a journal. Please see the Ledger Rebuild run sheet for those jobs that read a ledger as input. The few that read both a ledger and a journal are listed in the Ledger Rebuild run sheet.

Accounting, Cash, 1099, Internal, and BFY Not FY Journals

Daily Bank Account Balance (ABAL) – This job depends on daily runs to create individual records for each date. If the **Cash** Journal is rebuilt those individual date records cannot be recreated. Existing data on the Bank Account Balance page is incorrect and should be deleted so that a new record for the current date can be built with the corrected Cash Journal. All records on JRNL_LOG with a PROC_ID of ABAL should be deleted as well. The next run of the process will read the entire journal to build a single record. This run will take longer than normal to complete. To recreate any prior records would be a conversion effort outside of the ABAL job.

Clearing Account Maintenance (CAM) – If it is found that the source journal to this process (**Cash** or **Accounting**) has to be rebuilt, the process must be run to finish what is in the current version of the journal before a rebuild starts. The reason for this is that after the rebuild, the ordering of records in the journal will change, preventing any means of starting at the last record processed to only select unprocessed records. As the manual analysis in 7b above will have to be done if the process has run before, the final run on the journal will be part of that analysis. Only if the process has never run can 7b and the following JLOG update be skipped as the process needs to read the entire rebuilt journal.

After the rebuild and before the next run of the Clearing Account Maintenance chain, Journal Log data will need to be manipulated. The latest record where PROC_ID is CAM on JRNL_LOG should be found and the END_JRNL_REC_NO should be set to what is the maximum record number in the journal after the rebuild. It is very important that this

setup be done before any new transactions post to the journal. A JRNL_LOG record sample is shown for reference below.

Tables | Execute SQLs | jrnl_log

Proc Id	Proc Run Dt	Jrnl Nm	Bgn Jrnl Rec No	End Jrnl Rec No
CAM	10/6/2008 15:51:57	JRNL_ACTG	1	191763

Ledger Engine (LDGRPOST) – For the **Accounting** Journal (or if a ledger is built from the **Cash**, **1099**, **Internal**, or the **BFY Not FY** Journal) there will have to be a rebuilding of each ledger summarized from it. Until those are rebuilt, System Assurance 7 and 8 will report the ledger as out of sync. For each ledger an instance of the Rebuild Ledger job has to be run, followed by one or more instances of the Ledger Engine in rebuild mode. Old JRNL_LOG records for the rebuilt journal (JRNL_NM = JRNL_ACTG for example) with a PROC_ID of LDGRPOST and DESC of something other than 'Journal Rebuilt' (that DESC is for the Journal Rebuild batch job only) should be deleted as the BGN and END DOC_UNID values from prior Ledger Engine runs are no longer valid as the Journal Rebuild loaded the journal with all new DOC_UNID values.

Bank Account Transfer (ABATP) – The approach taken for this job is the same as that for the Clearing Account Maintenance job with PROC_ID changing from CAM to ABATP for JRNL_LOG records created for the **Cash** or **Accounting** Journal.

System Assurance 3 (SA3) – When the journal assured by SA3 is rebuilt, all JRNL_LOG records with PROC_ID of SA03 and that journal name in the JRNL_NM field should be deleted, followed by a full run for the journal with all the parameters of a normal full run.

System Assurance 9 (SA9) – This assurance runs mainly from the **Accounting** Journal, with some input from the posting line catalog for final transactions not yet journalized (a situation that can be avoided). If the Accounting Journal is rebuilt, delete all JRNL_LOG records with PROC_ID of SA09. The next run of SA9 should be done in full mode, which will take longer than the normal incremental run.

Cost Accounting Journal

Ledger Engine (LDGRPOST) – There will have to be a rebuilding of each ledger summarized from the Cost Accounting Journal. Until those are rebuilt, System Assurance 7 and 8 will report the ledger as out of sync. For each ledger rebuilt an instance of the Rebuild Ledger job has to be run, followed by one or more instances of the Ledger Engine in rebuild mode. Old JRNL_LOG records for the rebuilt journal (JRNL_NM = JRNL_CA) with a PROC_ID of LDGRPOST and DESC of something other than 'Journal Rebuilt' (that DESC is for the Journal Rebuild only) should be deleted as the BGN and END DOC_UNID values from prior Ledger Engine runs are no longer valid. The Journal Rebuild loaded the journal with all new DOC_UNID values.

Overhead Rate Process (OHRB) – The approach taken for this job is the same as that for the Clearing Account Maintenance job with PROC_ID changing from CAM to OHRB for JRNL_LOG records and the journal used is Cost Accounting instead of Accounting.

Program Asset Generation (UPDFACPER) – The approach taken for this job is the same as that for the Clearing Account Maintenance job with PROC_ID changing from CAM to UPDFACPER for JRNL_LOG records and the journal used is Cost Accounting instead of Accounting

Reimbursement Selection (RSEL) – The approach taken for this job is the same as that for the Clearing Account Maintenance job with PROC_ID changing from CAM to RSEL for JRNL_LOG records and the journal used is Cost Accounting instead of Accounting

Any reporting only batch process such as Budget vs. Actual, Open Items, Transaction Listing, or Trial Balance will not require any special attention. The next run will report on the corrected data. The various 1099 and Backup Withholding jobs, System Assurances 12 and 15, along

with Cost Allocations are programs that read a journal, but these do not track progress through JLOG so they are not directly impacted by a rebuild. The analysis in 7b may be required, but there are not steps to take before the next run.

Processing steps:

1. **Parameter Validation**: The first step of the job is to validate all input parameters.
2. **Journalization**: The second step starts with the creation of a Journal Log record then continues with purging all or a subset of journal records along with any Journal Ledger Cross Reference records that may exist. The step continues with the gathering of Posting Line Catalog records based on selection parameters and the journal being rebuilt. Those records are then sorted and then passed through the same posting logic found in the Journal Engine. When all records are posted, the Journal Log record is updated with a status of 'Final' (2).

   If any of the selection batch parameters (Budget Fiscal Year, Fiscal Year, Beginning Run Date Time and Ending Run Date Time) is provided then a partial rebuild is implied. Existing journal records from the journal satisfying the batch parameter selection criteria are purged. Journal/ledger Cross-reference records for those journal records are also purged.

   If all the selection batch parameters are left blank then full rebuild is implied in which case all records from the journal to be rebuilt are purged. Journal/ledger Cross-reference records for the journal are also purged.

Process Steps	Messages

1. Parameter Validation	• Run Started
	• Parameters are listed with values
	• Validating Batch Parameters
	  ◦ If the parameter is invalid, an error message will be displayed.
	  ◦ If the Commit Block or Select Block Size parameters are not specified, the log will state what value defaulted (100 and 1000 respectively).
	• Batch Parameters validation completed

2. Journalization	• Process Started • Purging records from the Journal Ledger Cross Reference table for Journal ID ## started • Purging records from the Journal Ledger Cross Reference table for Journal ID ## completed • Purging records from Journal ID ## started • Purging records from Journal ID ## completed • Rebuilding Journal records started • Number of Posting Lines processed: 5000 (repeated until done using Progression Size parameter) • Rebuilding Journal records completed • Run Ended • The ## above is replaced in the Job Log with the ID of the journal rebuilt.

Major Input

- Posting Line Catalog (PSTNG_LN_CAT)

Other Input

- Posting Code (R_PSCD)
- Journal/Ledger Control (R_JRNL_LDGR_CTRL)

Parameters

Parameter	Description	Default Values
Journal Id	Required Journal Id of the journal to rebuild. See the Journal/Ledger Control table for valid values. Fixed Asset Component, Commodity, and Budget Journal are not allowed.	(blank)
Delete Block Size	Required. This parameter determines the number of records to be deleted in one block. The value should not be too low as it would degrade performance with multiple deletes of a small number of records. The value should not be too high as that can result in an error when deleting a high volume of records. The parameter value must be a positive integer. If the supplied value is not a positive non-zero integer value, all journal records will be deleted in one transaction as the program will consider such a value as no block size. If the parameter value is left blank, the system will default a value of 150000.	150000
Commit block size	Required parameter for performance tuning. The value will be the number of posting lines to be committed at a time. If left blank the system will default a value of 100.	100
Budget Fiscal Year	Optional selection parameter for a partial rebuild. Enter a Budget Fiscal Year as CCYY. Only one value is allowed and	(blank)

	cannot be combined with Fiscal Year or either Run Date Time parameter. The Budget Fiscal Year parameter value is prohibited with 1099 journals (if the Journal Id indicates a 1099 journal).	
Fiscal Year	Optional selection parameter for a partial rebuild. Enter a Fiscal Year as CCYY. Only one value is allowed and cannot be combined with Budget Fiscal Year or either Run Date Time parameter. Any value provided here is interpreted as a Calendar Year for 1099 journals (if the Journal ID indicates the 1099 journal on Journal Ledger Control).	(blank)
Beginning Run Date Time	Optional parameter for a partial rebuild. Enter as mm/dd/ccyy hh:mm:ss. When entered, selection of posting lines and journal records will be for all records with a Run Time Date equal to and from the parameter. Cannot be combined with Budget Fiscal Year or Fiscal Year.	(blank)
Ending Run Date Time	Optional parameter for a partial rebuild. Enter as mm/dd/ccyy hh:mm:ss. When entered, selection of posting lines and journal records will be for all records with a Run Time Date equal to and before the parameter. Cannot be combined with Budget Fiscal Year or Fiscal Year.	(blank)
Select block size	Required parameter for performance tuning. The value is the number of records that can be fetched as a single block and stored to be processed. If left blank the system will default a value of 1000.	1000
Progression Size Counter	Required parameter to trace the progress of a run with	5000

	progression messaging during journalization. If left blank the system will default a value of 5000.	

Major Output

- Journal (JRNL_XYZ) where XYZ is decided by batch parameter JRNL_ID

Other Output

- Journal Log (JRNL_LOG)
- Journal/Ledger Cross Reference (JRNL_LDGR_XREF)
- Unique Numbering (R_IN_UNID) – conditional based on full rebuild

Job Return Codes

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

Return Code	Condition
Successful (1)	All batch parameters were valid and Posting Line Catalog records were found and successfully journalized.
Warning (4)	The job does not end with this return code.
Non-Fatal Error (8)	This error is issued when no Posting Lines are selected.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations. • APPCTRL – Real-Time Posting is not 'false'. • Update to the Journal Log table fails
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

Posting lines selected are sorted by:

- Transaction Code (DOC_CD)
- Transaction Department (DOC_DEPT_CD)
- Transaction Id (DOC_ID)
- Transaction Version Number (DOC_VERS_NO)
- Transaction Posting Internal Line Number (DOC_PSTNG_NO)

Selection Criteria

- Deleting records from the journal to be rebuilt:
  - Budget Fiscal Year is batch parameter supplied (BFY = CCYY) and
  - If rebuilding a non-1099 journal: Fiscal Year is batch parameter supplied (FY_DC = CCYY) and
  - If rebuilding a 1099 journal: Calendar Year matches the Fiscal Year batch parameter value provided (year portion of Transaction Record Date = CCYY) and
  - Run Date Time is greater than or equal to the batch parameter BGN_TMDT supplied (RUN_TMDT >= mm/dd/ccyy hh:mm:ss) and
  - Run Date Time is less than or equal to the batch parameter END_TMDT supplied (RUN_TMDT <= mm/dd/ccyy hh:mm:ss

- Note: Use of the Budget Fiscal Year, Fiscal Year, and Beginning/Ending Run Time Date parameters are mutually exclusive. Budget Fiscal Year can be specified, or Fiscal Year can be specified, or both Beginning/Ending Run Time can be specified.
  - Deleting records from the Journal Ledger Cross Reference (JRNL_LDGR_XREF):
    - Journal ID (JRNL_ID) equals that of batch parameter and
    - Journal Record Number (JRNL_REC_NO) equals a record number purged from journal before rebuilding
  - Posting lines from the Posting Line Catalog (PSTNG_LN_CAT) for reposting:
    - Transaction phase is *Final* or *Historical Final* (DOC_PHASE_CD in (3,5)) and
    - Journal Posting Indicator is *Posted* (JRNL_PSTNG_IND = 3) and
    - Posting Amount is not $0 (PSTNG_AM <> 0) and
    - Budget Fiscal Year is batch parameter supplied (BFY = CCYY) and
    - If rebuilding a non-1099 journal: Fiscal Year is batch parameter supplied (FY_DC – CCYY)
    - If rebuilding a 1099 journal: Calendar Year (CY) matches the Fiscal Year batch parameter supplied. (CY = CCYY) and
    - Run Date Time is greater than or equal to the batch parameter BGN_TMDT supplied (RUN_TMDT >= mm/dd/ccyy hh:mm:ss) and
    - Run Date Time is less than or equal to the batch parameter END_TMDT supplied (RUN_TMDT <= mm/dd/ccyy hh:mm:ss

  Note: Use of the Budget Fiscal Year, Fiscal Year, and Beginning/Ending Run Time Date parameters are mutually exclusive. Budget Fiscal Year can be specified, or Fiscal Year can be specified, or both Beginning/Ending Run Time can be specified.

Problem Resolution

- It is a good practice to look at the log of the job even if the job has run successfully.
- The process has restartability. If the job fails for any data setup reasons then correct the data setup and restart the job. When the job is restarted it begins processing from the last

unsuccessful transaction on which the failure occurred. Alternatively, a new instance of the job could be submitted.

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	Job does not end with this return code	N/A	N/A
Non-Fatal Error (8)	No Posting Line Catalog records were selected	Check partial rebuild selection parameter(s). If invalid values (not that parameter edits failed but invalid because no activity exists for the parameter), run a new instance of the engine with correct parameters. If valid values, then no further action is required.	N/A
Failed (12)	Beginning Run time is not in proper format. **Sample Message:** Invalid Beginning run time. Enter Beginning run time in mm/dd/yyyy hh:mm:ss format	Run a new instance of the engine with correct parameters.	N/A
	The job fails during the purging of journal records.	Job cannot be restarted in this case. Investigate and resolve the reason for the failure. Then run another instance of the job after deleting the JRNL_LOG record created by the process.	In cases where the application cannot delete all records, then a database tool should be used.
	The job fails during record creation.	Investigate and resolve the reason for failure and then restart the job.	N/A

Terminated (16)	Job is terminated manually by the user	Reason for the termination needs to be investigated. Reschedule a new job after reason for termination is resolved.	N/A
System Failure (20)	Job is terminated because of database server or network issues.	Reason for the failure needs to be investigated. Reschedule a new job after the reason for the failure is resolved.	N/A

2.1.21 Ledger Engine

Job Name	Ledger Engine
Recommended Frequency	Frequency depends on need: 1. Once or more nightly 2. On demand 3. Frequently to populate ledgers on a more real-time basis Details are given later in the "Summarized Descriptions of each Ledger Engine Run Mode" section for when the various modes of the engine should be executed. Batch jobs that read ledgers as the sole input require the engine before running (Annual Close for example). Batch jobs that read ledgers in addition to journals and posting lines perform faster with the engine running first. Certain other jobs should not be run while the engine is running. Please refer to the "Important Run Instructions" section for more details.
Single Instance Required	No, a single instance is not required. Multiple instances can be run concurrently with certain restrictions listed in the "Important Run Instructions" section.
Can be Restarted?	No
Reports Generated	None

Overview

As transactions are processed, journal records are created simultaneously or through the Journal Engine. Some of these journals require that their records are summarized into one or more ledgers. That definition along with the summarization rules are defined in the Journal Ledger Control Detail (JLCTRL) table. The Ledger Engine is the process that takes those journal records and writes them to the appropriate ledgers or what is commonly called 'ledgerizing' the records.

The Ledger Engine tracks progress through the various journals with the JRNL_LOG table that is viewed online with either the Journal Log (JLOG) or Ledgerization Log (JLOGLDGR) pages. Rather than track progress through a journal by the Journal Record Number assigned sequentially to each journal record, the Ledger Engine uses another pair of numbers assigned as journal records are created. The first of these is a Transaction Unique ID (DOC_UNID) assigned to each combination of Transaction Code, Transaction Department, Transaction ID, and Version written from the Posting Line Catalog. The second of these is a Transaction Line Unique ID (DOC_LN_ID) assigned to each journal record from a transaction. The use of these two in combination to track Ledger Engine progress is done to simplify the ledgerization requirement to write all and not part of any transaction to the ledger at one time. More detail is given to this selection and tracking process later in the detailed discussions of each run mode of the Ledger Engine.

Records are written to the Journal Log stating ranges that were either successfully ledgerized, failed to ledgerize, or where a gap of records were found in the journal. Due to multiple Application Servers processing transactions and cases where Sequence Block Count for Journal DOC_UNID on the Unique Numbers (R_IN_UNID) table is set to a value greater than one, some DOC_UNID values can be missing and might be filled in at a later point in time. In such cases, the Ledger Engine will log the missing DOC_UNID values as Gap Found on Journal Log.

There is another table that is updated by the engine other than the ledgers and the log – the Journal Ledger Cross Reference (JLXREF) table. On this table, the engine records each journal record summarized into a specific ledger record for all ledgers. External auditing as well as internal auditing performed by systems assurance jobs use this table.

Parallel Processing

As the Ledger Engine often has to handle a large volume of journal records from multiple journals, multiple instances of the Ledger Engine can be scheduled at the same time to allow parallel processing and thereby increasing the throughput of the process. Multiple instances can be submitted simultaneously or sequentially to select all eligible journal records from all source journals. These instances will proceed through each source journal sequentially until finished. A batch parameter also allows multiple instances to target a specific source journal. When more than one journal is supporting the ledger tables and one of the journals is considerably larger than the other(s), ledger engines targeted to specific journals may provide better overall ledgerization performance time.

When multiple instances of the Ledger Engine are running, each instance has to lock a dummy record on the Journal/Ledger Lock (JRNL_LDGR_LOCK) table. The dummy record selected to be locked is determined by the ledgerization mode and the input journal. The first instance that comes in will lock the dummy record, which will cause subsequent instances to wait. The instance that obtains the lock goes out to find its next unit of work. It marks this unit of work with the status of Intended and releases the lock on the dummy record. At this time the waiting instances can proceed with the next unit of work and so forth. That way it is assured that all instances are working on mutually exclusive, non-overlapping units of work from the Journal Log.

Important Run Instructions

The Ledger Engine should NOT be run concurrently with the following processes:

- The Identify and Archive Stale Gaps batch job should not be running as it will be updating gap records on the Journal Log table while new gap records are being created (*normal* mode) or existing gap records are being evaluated (*process gaps* mode).
- The Rebuild Ledger batch job because it would be concurrently clearing the ledger for the rebuilding of records, the Journal Ledger Cross Reference of records, and reading the Journal Log table to determine the rebuild range.
- Ledger Engine in Continue Rebuilding mode should not be run when:
    1. Another instance of the Ledger Engine is running in Reprocess Failed Work, Reprocess Failed Rebuild Work, or Process Logged Gaps mode.
    2. If any Journal Log records created from a prior run have a Status (ST_FL) of *Intended (value of 1).* Such a situation can arise:
        1. If the prior instance of the Ledger Engine run in any mode ends in a Job Return code other than *Successful.*
        2. If an instance of the Ledger Engine job is killed.
        3. If a prior instance of the engine created a Journal Log record with a Status of *Failed (value of 4)* or *Gap Failed (value of 9)* for a Process Id of LDGRPOST or LDGRRBLD.

When any of these situations occurs (refer to the Job Logs page created for each instance for details on each), there is the potential that some work will remains in the Intended status on the Journal Log table.

Any such Intended entries should have the status changed to Failed so that a Ledger Engine running in Reprocess Failed Work mode or Reprocess Failed Rebuild Work mode will pick them up for processing. Run the following Update SQL when no instance of the Ledger Engine in any

mode is running. Again, it is very important to ensure that no instance of the Ledger Engine is running when these database scripts are being fired. These scripts will change any Intended status to Failed and Gap Intended status to Gap Failed.

```
UPDATE <dbSchemaName>.JRNL_LOG SET ST_FL=4 WHERE
ST_FL=1 AND PROC_ID IN('LDGRPOST','LDGRRBLD') ;
UPDATE <dbSchemaName>.JRNL_LOG SET ST_FL=9 WHERE
ST_FL=6 AND PROC_ID IN('LDGRPOST','LDGRRBLD') ;
```

It is very important these instructions are followed; otherwise, it will lead to indeterminate results.

Summarized Descriptions of Each Ledger Engine Run Mode

The following section presents details on each mode of the Ledger Engine. Please refer to the "Detailed Descriptions of Each Ledger Engine Mode" section for even more details on record selection and Journal Log updates.

- **Normal**: This is the most common mode for the engine where the selection of the next set of unledgerized journal records occurs. Previously logged gaps in transaction unique ID's are not evaluated and previous units of work that failed to ledgerize are not selected. Only new journal records written since the last run of the engine in normal mode are selected. Any gaps found in those unique ID's will also be logged so that another mode of the engine can investigate them later to find out if they have been filled in by subsequent transaction processing.

  All ledgers are updated as part of this process. No specific ledger can be updated with this mode. The mode will post records from all active journals that summarize to ledgers or post only to the ledgers built from a single specified journal specified as an input parameter.

  The engine is run in this mode one or more times a day to provide updated information in the online ledger queries as well as to provide updated information for other batch jobs that read a ledger as input. Multiple instances in this mode are recommended for performance.

- **Process Gaps**: This mode evaluates all logged gaps in transaction unique ID's on the Journal Log table from previous runs in *normal* mode. Each gap is evaluated to determine if the missing range of unique ID's has had one or more ID's completed. If so, those new records will be ledgerized and any remaining gap logged. If not, then the gap will remain logged. Later runs in the *process gaps* mode will re-evaluate those ranges until the Identify and Archive Stale Gaps batch job is run to remove the gaps based on aging criteria.

  All ledgers are updated as part of this process. No specific ledger can be updated with this mode. The mode will post records from all active journals that summarize to ledgers.

  The engine is run in this mode one or more times a day to provide updated information in the online ledger queries as well as to provide updated information for other batch jobs that read a ledger as input. A run in this mode should always follow one in the *normal* mode. Only a single instance of this mode is recommended unless system configuration results in many records being subsequently written to gaps.

  It is recommended that sites run the Ledgerization process in "Process Logged Gaps Mode" on a frequent basis. This will ensure that all journal records are included on the Ledgers. If you use asynchronous posting and run Ledgerization once a night, you may want to run this mode once a week. If Ledgerization is run multiple times during the day, it is a good idea to run this mode once a night.

- **Process Failed Work**: This mode attempts to post all units of work with a failed status on the Journal Log table from previous runs of a mode that is not rebuilding a ledger:

normal, process gaps, or *process failed work*. Failure to post a block of journal records is uncommon and unlikely for normal processing. However, it is important to get all journal records ledgerized so the system logs these failures and issue messages in the job log of the Ledger Engine instance that could not process the block. When the failure occurs, no ledgers are updated so that the situation where one is more current that the others will not be created.

Each failed block of journal records is selected and an attempt is made to ledgerize the work. Within the failed block of journal records it logs intermittent missing transaction unique IDs as Gap Found on the Journal Log. Running a single instance of the engine in this mode should ledgerize the block. If the block cannot be ledgerized, then the block will remain marked as failed on the Journal Log. When this occurs, the reason for the failure of this block to ledgerize should be investigated.

All ledgers built from the journal identified in the failed block are updated as part of this process. No specific ledger can be updated with this mode.

The engine is run in this mode under one of two situations. A manual run is executed when an instance of the engine reports a failed unit of work. An automatic run is scheduled daily (or more frequently) after the engine is run in *normal* and *process gap* modes have finished as an automatic assurance when manual evaluation is not or cannot be performed to determine if there are any failed blocks of work. A single instance of this run is recommended unless there are many blocks of failed work.

- **Continue Rebuilding**: This mode always follows the running of the Rebuild Ledger batch job. The previous job was run for a specific Ledger ID and created an instruction record on the Journal Log table in addition to clearing out records from the ledger and from the Journal Ledger Cross Reference table for that ledger. The *continue rebuilding* mode will read that instruction record, select a block of work from the instruction, update that instruction to reflect the selection, and then ledgerize that block of work to that specific ledger being rebuilt. This mode will continue selecting blocks of work and posting them based on that instruction record until the end of the instruction is reached and the ledger has been rebuilt. While rebuilding the Ledger, it skips processing transaction unique IDs that are marked with *Gap Found*, *Gap Stale* or *Gap Intended* status. This mode does not log any intermittent missing transaction unique IDs as Gap Found.

Any new records written to the source journal since the last execution of the engine in *normal* mode will not be selected as all ledgers must remain in sync with each other in terms of journal record selection. For this reason and others, warnings are given in the "Important Run Instructions" section for rebuilds about what should not be running at the same time a rebuild is progressing.

Only the single ledger is updated as part of this process. The engine is run in this mode only on demand after the Rebuild Ledger job has been run. The specified ledger in both jobs must be the same. Multiple instances in this mode are recommended for performance.

- **Process Failed Rebuild Work**: This mode attempts to process all units of work with a failed status on the Journal Log table from a previous rebuild run. Failure to rebuild a block of journal records is uncommon and unlikely for normal processing. However, it is important to get all journal records ledgerized so the system logs these failures and issues messages in the job log of the Ledger Engine instance that could not process the block. When the failure occurs, the ledger is not updated with any information in the block.

Each failed rebuild block of journal records is selected and an attempt is made to ledgerize the work. Within the failed block of journal records it does not log intermittent missing transaction unique IDs as Gap Found on the Journal Log because the Continue Rebuild ledger mode never dealt with Gap entries. Running a single instance of the engine in this mode should ledgerize the block. If the block cannot be ledgerized, then the block will remain

marked as failed on the Journal Log. When this occurs, the reason for the failure of this block to ledgerize should be investigated.

Only the single ledger is updated as part of this process. The engine is run in this mode only on demand after the Rebuild Ledger job has been run. The specified ledger in both jobs must be the same. A single instance of this run is recommended unless there are many blocks of failed work.

Processing steps: As each mode of the engine progresses, messages will be issued to the Job Log for each step in the engine: parameter validation and ledgerization.

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters • If the parameter is invalid, the invalid value will be displayed in the log along with the error message. • Batch Parameter validation completed

2. Ledgerization	• Processing started in XYZ mode (*where XYZ is the run mode*)
	• Ledgerizing journal started
	  • Instance starting record selection and posting for Journal ID #. (where # is the Journal ID as defined on the Journal Ledger Control Detail table)
	  ▪ When a journal is the source to a ledger: # records processed (where # is a progression block size defined as an input parameter. # is 0 if no records for processing)
	  ▪ When a journal is not the source for any ledger: No Ledger found for summarizing Journal ID 'n' on the Journal/Ledger Control Detail table. (where n = 1, 2, etc)
	  • Instance has completed record selection and posting for Journal ID # (where # is the Journal ID as defined on the Journal Ledger Control Detail table)
	• Ledgerizing journals completed
	• Run ended

There are many other messages issued that will detail problems encountered when an instance finishes the Ledgerization step with a return code other than successful. A few of the more common ones are:

- "Active intended rebuild instructions for given rebuild ledger not found. The rebuild may be complete." Here a rebuild instance found no work.

- If no eligible records found for any source journal in normal mode, then this message will be issued: “No eligible journal records found that match the selection criteria for ledgerizing.” The job will finish with a return code of warning.
- “Journal Log record (ID ##) updated with a status of failed. Submit another instance to process failed work.” This message is issued when any mode cannot successfully ledgerize a unit of work. The instance will end with a return code of non-fatal.

Restartability Information

When parameter validation causes the job to end with a Failed Return Code, restarting the job is not an option. A new instance of the engine should be started with valid parameters.

When record selection or ledgerization results in the job ending with a Failed Return Code, a restart is not required. Running another instance of the engine in normal mode will select any new journal records, an instance in process gaps mode will re-evaluate gaps, an instance in rebuild mode will continue rebuilding from the instruction record, and an instance in process failed work or process failed rebuild work will post remaining records that are marked failed. Please note that one of the SQL statements mentioned earlier will have to be run if there are any Journal Log records with a status of intended.

Major Input

Tables

1. The Journal/Ledger Control Detail (R_JRNL_LDGR_CTRL) table provides summarization rules describing which ledgers are summarized from which journals and how are they summarized. These Journal/Ledger Control records specify the names of the Source Journals. These are typically named *JRNL_xxxx*, where *xxxx* is a descriptive identifier (for example, JRNL_ACTG is the Accounting Journal and JRNL_CA is the Cost Accounting Journal).
2. Potential input journals (in baseline CGI Advantage Financial) include:
   - Accounting Journal (JRNL_ACTG)
   - Cost Accounting Journal (JRNL_CA)
   - Cash Journal (JRNL_CASH)
   - 1099 Journal (JRNL_1099)
   - Internal Journal (JRNL_INT)
   - Cross-Year (Budget FY ≠ FY) Journal (JRNL_BFYNOTFY)
   - Fixed Asset Accounting Journal (JRNL_FA)
3. The Journal Log (JRNL_LOG) table provides information for record selection.
4. The Journal Ledger Lock (JRNL_LDGR_LOCK) table provides the assurance that no two instances of the engine will be creating a block of work at the same time. This table is designed to contain dummy entries that will be locked by instances of the Ledger Engine running in parallel. It is used to ensure that all instances of the Ledger Engine are working on mutually exclusive, non-overlapping units of work from the Journal Log and thus resolves Selection logic conflicts when parallel runs of the Ledger Engine are scheduled. Entries are pre-populated on the table.
   - Entry with PROC_ID=LDGRPOST for every journal

- Entry with PROC_ID=LDGRRBLD entry for every ledger
- Entry with PROC_ID=POSTFAILED entry for every journal
- Entry with PROC_ID=PROCGAPS entry for every journal
- Entry with PROC_ID=RBLDFAILED entry for every ledger

Batch Parameters

Parameter	Description	Default Value
Allowed To Exceed	Allowed to exceed max for last transaction. A required parameter that instructs the engine on whether or not to allow the last transaction selected to exceed the Maximum Journal Record's parameter or the Commit Block Size. If set to N, then that last transaction will not be selected in that instance of the engine if the Maximum Journal Records parameter is used. If the Commit Block Size is exceeded and the parameter is set to N, then that transaction will go into the next block of work.	Y
Commit Block Size	Commit Block Size. This required performance parameter is the number of journal records that will be committed at one time to each applicable ledger. Too low of a value can result in more time spent writing, reading, and updating Journal Log entries. Too high of a value can demand higher memory capacities for application and database servers.	250
Journal ID for Normal Mode	Source Journal ID to Ledgerize in Normal Mode (leave blank for all source journals). This optional parameter is only used when the engine is run in *Normal* mode. When a Journal ID is supplied in this parameter, then the Ledger Engine only ledgerizes records from that particular Journal. Valid values are valid active Journal Identifiers (numbers) from the Journal/Ledger Control Detail (R_JRNL_LDGR_CTRL).	(blank)
Max Journal Records	The maximum number of journal records to ledgerize. This field indicates the maximum number of journal records to ledgerize for a given instance of Ledger Engine. For example, if Max Journal Records = 1000, then at most 1000 records from a given journal will be	(blank)

	ledgerized. The process does not halt exactly after ledgerizing 1000 records, it will halt below that figure if the last transaction selected went over the limit and the Allow to Exceed were set to N. If set to yes, it would halt above that limit at the point all records from that last transaction were selected. If a value is not provided then all journal records selected for the particular mode are ledgerized. Valid values are 1 to 2,147,483,647.	
Process Logged Gaps	Process Logged Gaps A required parameter that when set to Y will instruct the engine to only select those Journal Log LDGRPOST records with a Status (ST_FL) of Gap Found (5). When set to N, those Journal Log records will not be processed. A setting of Y will result in the engine running in *Process Gaps* mode.	N
Rebuild Ledger ID	Rebuild single ledger from scratch. When the engine should run in *Continue Rebuilding* or *Process Failed Rebuild Work* mode, this parameter is completed with the Ledger ID used in a prior Rebuild Ledger job. Otherwise, it should be left blank. Valid values are valid Ledger Identifiers (numbers) from the Journal/Ledger Control Detail (R_JRNL_LDGR_CTRL).	(blank)
Rebuild Start	Start rebuilding a ledger. This is a protected parameter that should always be set to N as it is the Rebuild Ledger program that starts any ledger rebuild process. The Ledger Engine only continues the rebuild process.	N
Reprocess Failed Work	Process JRNL_LOG records marked as Failed. A required parameter that when set to Y will instruct the engine to only select those Journal Log LDGRPOST or LDGRRBLD records with a Status (ST_FL) of Failed (4) or Gap Failed (9) for processing. When set to N, those Journal Log records will not be processed. A setting of Y will result in the engine running in *Process Failed Work* or *Process Failed Rebuild Work* mode.	N
SELECT_DOC_SIZE	Select block Size (Number of transactions in each Select statement). A required performance parameter that is	50

	the number of transactions that will be initially placed into a block of work for an instance of the engine. Too low of a value can result in more time spent writing Journal Log entries and too much time spent waiting for locks on the Journal Ledger Lock table. Too high of a value can demand higher memory capacities for application and database servers.	
PROG_CTR_SZ	A required block size to control progression messages detailing the progress of the engine in terms of journal records.	5000

Note: Values of Y or N are case insensitive.

Performance parameters need to be configured only once based on the processing and memory capabilities of the application and database servers. Once determined and set in Job Setup for the Ledger Engine as the default value, the same value can be used for any subsequent runs without user intervention.

The following combines parameter information for each run mode:

Mode	Allowed To Exceed	Rebuild Ledger ID	Reprocess Failed Work	Process Logged Gaps	Journal ID for Normal Mode
Normal	Y/N	-	N	N	Opt
Process Gaps	Y/N	-	N	Y	-
Continue Rebuilding	Y/N	√	N	N	-
Process Failed Work	Y/N	-	Y	N	-
Process Failed Rebuild Work	Y/N	√	Y	N	-

Opt indicates value is optional, √indicates required, - indicates leave blank

Major Output

Tables

The primary output of the Ledger Engine is one or more of the following ledger tables where new records are added and existing records are updated with new amounts:

- Ledger 001 (LDGR_001)
- Ledger 002 (LDGR_002)
- Ledger 003 (LDGR_003)
- Ledger 004 (LDGR_004)
- Ledger 005 (LDGR_005)

- Ledger 006 (LDGR_006)
- Ledger 007 (LDGR_007)
- Ledger 008 (LDGR_008)
- Ledger 009 (LDGR_009)
- Ledger 010 (LDGR_010)
- Accounting APD Ledger (LDGR_APD_ACTG)
- Cost Accounting APD Ledger (LDGR_APD_CA)
- Accounting FYTD Ledger (LDGR_FYTD_ACTG)
- Cost Accounting FYTD Ledger (LDGR_FYTD_CA)
- Accounting BFYTD Ledger (LDGR_BFYTD_ACTG)
- Cost Accounting BFYTD Ledger (LDGR_BFYTD_CA)
- Accounting ITD Ledger (LDGR_ITD_ACTG)
- Cost Accounting ITD Ledger (LDGR_ITD_CA)
- Ledger SA Budgets (LDGR_SA_BUD)
- Full Detail Accounting Ledger (LDGR_FYDAD)
- Fixed Asset Accounting Ledger (LDGR_FA_ACTG)
- Journal Ledger Cross Reference (JRNL_LDGR_XREF) is updated to list those journal records posted with the ledger records they created or updated.
- Journal Log (JRNL_LOG) is updated listing all blocks of work and the resulting status of each. Those records are identified with the Process ID's of LDGRPOST or LDGRRBLD.

Job Return Codes

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

Return Code	Condition
Successful (1)	Journal records were found and successfully ledgerized.
Warning (4)	This return code will be issued under several conditions: • No new journal records found for *normal* mode • No failed work found for *process failed work* or *process failed rebuild work* modes • No gap records found or no new transaction unique IDs found in gap record(s) for *process gaps* mode. • No LDGRINSTR record exists for the ledger selected in the *rebuild* mode that has any remaining work. • Allowed To Exceed is set to 'N' and Commit Block size has a very low value.
Non-Fatal Error (8)	This return code will be issued for any run mode when a block of work cannot be ledgerized.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid

	• Run time exceptions for unexpected situations. • No LDGRINSTR record is found for ledger to rebuild • Lock to Journal/Ledger Lock table could not be obtained • Any update to the Journal Log table fails When this job ends with a return code of Failed, there may be Journal Log records remaining with a status of *intended* that have to be updated with one of the SQL's provided in the "Important Run Instructions" section.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated, there may be Journal Log records remaining with a status of *intended* that have to be updated with one of the SQL's provided in the "Important Run Instructions" section.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, there may be Journal Log records remaining with a status of *intended* that have to be updated with one of the SQL's provided in the "Important Run Instructions" section.

Sort Sequence

N/A

Selection Criteria

Refer to the "Overview" section for selection criteria of each Ledger Engine run mode

Problem Resolution

For performance reasons, the Identify and Archive Stale Gaps job should be run periodically to remove gaps in journal records being tracked and processed by this program. Please refer to the Identify and Archive Stale Gaps run sheet for more details on scheduling and recommended parameters.

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

Parameter Validation

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

Failed (12)	Required Parameters are not entered. **Sample Message:** Select Block Size required.	Run a new instance of the engine with correct parameters.	

Ledgerizing

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	Normal mode	No action necessary unless the perception is that there are records. Retrieve last Journal Log record for LDGRPOST process ID for a given journal. Determine if there are in fact no journal records with a higher DOC_UNID than that found on the last Journal Log record.	If there are and the records were created after the Ledger Engine was run, then run another instance or wait until the next scheduled run. If there are and they were in the journal before the Ledger Engine and the Maximum Journal Records parameter was not used, then run another instance. If the records are still skipped or logged into a gap, then the records should be investigated before running any subsequent process where the inclusion of those records in the ledger is critical.
Warning (4)	Process Gaps mode	No action necessary	
Warning (4)	Process Failed Work mode	No action necessary unless the perception is that there is at least 1 failed block of records. Search the Journal Log for a ST_FL value of 4 or 9.	If there are not any, then no action is required. If there are, run another instance to select the records and search again when that completes. If the failed records are still there, then the records should be investigated

			before running any subsequent process where the inclusion of those records in the ledger is critical.
Warning (4)	Continue Rebuild mode	No action necessary unless the perception is that there is remaining rebuild work. Search the Journal Log for the LDGRINSTRC and retrieve the ending transaction unique ID. Look for a LDGRRBLD for the ledger that ends with that same ID.	If a LDGRRBLD exists that has completed the LDGRINSRC, then no action is required. If there isn't, run another instance to complete the process and search again when that completes. If the rebuild is still not completed, then the reason for the incompletion should be investigated before running any subsequent process where the inclusion of those records in the ledger is critical.
Warning (4)	Process Failed Rebuild Work mode	No action is necessary unless the perception is that there is at least 1 failed block of records. Search the Journal Log for a ST_FL value of 4 or 9.	If there are not any, then no action is required. If there are, run another instance to select the records and search again when that completes. If the failed records are still there, then the records should be investigated before running any subsequent process where the inclusion of those records in the ledger is critical.
Warning (4)	Error: 'Allowed To Exceed is set to 'N' and Commit Block size has very low value' has been issued.	1 or more transactions were found that contain a greater number of journal records to post than the Commit Block size can handle in one instance. Because the Allowed To Exceed parameter was set to No, the transaction(s) was	To prevent the occurrence again, consider running the engine with the Allowed To Exceed parameter set to Y or with a higher commit Block Size that will not present memory problems. Another alternative is to use the Transaction Component

		not able to post. Run another instance of the engine in process failed work mode with either a Commit Block Size larger than the largest transaction, or run with the Allowed To Exceed parameter set to Yes.	Requirements feature in the Administration application to limit the number of accounting lines for the transaction type that had such a large number of journal records.
Non-Fatal Error (8)	A block of work on the Journal Log was written with a failed status.	Submit an instance of the engine in *process failed work* or *process failed rebuild work* mode (depending on the type of original run).	Review the Journal Log after the instance selecting failed work to determine if there is any more failed work (not the failed work would likely end as Non-Fatal also). If there is not any failed work, then no action is required. If there are, run another instance to select the records and search again when that completes. If the failed records are still there, then the records should be investigated before running any subsequent process where the inclusion of those records in the ledger is critical.
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job or running any process where an up-to-date ledger is required.	
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.	

Detailed Descriptions of Each Ledger Engine Mode

Normal Mode

This mode grabs the next unit of work (unledgerized journal records) from all active journals and ledgerizes them to applicable ledgers. It reads those journals one-by-one and ledgerizes them. Let us assume that an instance of Ledger Engine - L1 is scheduled in Normal mode. Select block Size is 100 and Commit block Size is 200. The following entries exist on the Journal Log before L1 is scheduled:

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Final	1	200
2	LDGRPOST	1	Failed	201	300
3	LDGRPOST	2	Final	1	300

L1 sees that the last attempted ledgerized DOC_UNID for journal 1 is 300. It then selects a dummy record on Journal/Ledger Lock table where PROC_ID=LDGRPOST and JRNL_LDGR_ID=1 and locks it. Meanwhile, if another instance of Ledger Engine, say L2, is scheduled in normal mode and it is also in the process of finding the next unit of work from journal 1 then it sees that the dummy record in the Journal/Ledger Lock table is already locked by L1 and it waits until L1 releases it. L1 determines its unit of work as (Last attempted DOC_UNID + Select block Size = 300 + 100 = 400). It adds UNID 4 as Intended to book its unit of work (DOC_UNIDs 301-400) and releases the lock on the dummy record.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
4	LDGRPOST	1	Intended	301	400

Since L1 has already released the lock on the dummy record, L2 can grab the lock and book its unit of work as (Last attempted DOC_UNID (400) + Select block Size of L2(100)) as shown below UNID 5 and continue further. Note that example of L2 is explained to clarify how parallel runs work in coordination in terms of picking non-overlapping units of work.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
5	LDGRPOST	1	Intended	401	500

Coming back to L1, L1 now attempts to ledgerize DOC_UNIDs 301-400. Suppose DOC_UNIDs 301-350 had journal records that summed up to Commit Block Size 200 then the original unit of work is split. A new JRNL_LOG entry, UNID 6, is logged to record the work that could be finished in the first Ledgerization Loop (DOC_UNIDs 301-350). Then the BGN_DOC_UNID of original Intended entry UNID 4 is updated to DOC_UNID 351 so that UNID 4 is logically marked as pending work that will be processed by L1 in next Ledgerization Loop.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
4	LDGRPOST	1	Intended	351	400
6	LDGRPOST	1	Intended	301	350

L1 ledgerizes DOC_UNIDs 301-350. If successful, it marks its intended entry UNID 6 as Final. If it failed, then it marks UNID 6 as Failed.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
4	LDGRPOST	1	Intended	351	400
6	LDGRPOST	1	Final	301	350

Then L1 invokes second Ledgerization Loop to process the pending Work UNID 4 (DOC_UNIDs 351-400). L1 ledgerizes DOC_UNIDs 351-400 and marks the intended entry UNID 4 as Final. L1 found DOC_UNIDs 370, 371 and 380 were missing during ledgerization, it then logs these DOC_UNIDs as Gap Found.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
4	LDGRPOST	1	Final	351	400
7	LDGRPOST	1	Gap Found	370	371
8	LDGRPOST	1	Gap Found	380	380

At this point L1 had finished ledgerizing its original unit of work (DOC_UNIDs 301-400) and now looks for the next unit of work (DOC_UNIDs > 400 and DOC_UNIDs > last attempted DOC_UNID(500) on journal Log). It then tries to lock the dummy record as explained earlier and continues Ledgerizing. When it has finished Ledgerizing all unledgerized records in journal 1, it proceeds to ledgerize journal 2 in a similar fashion and so forth.

Process Gaps Mode

This mode ledgerizes all DOC_UNIDs that were logged as Gap Found on the Journal Log and were later filled in. It selects Gap Found entries of all active journals, one by one and ledgerizes them into all applicable Ledgers. Let us assume that an instance of Ledger Engine L1 is scheduled in Process Logged Gaps mode. Select Block Size is 200 and Commit Block Size is 400. The following entries exist on the Journal Log before L1 is scheduled:

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Gap Found	100	200
2	LDGRPOST	1	Gap Found	400	450
3	LDGRPOST	1	Gap Found	500	600
4	LDGRPOST	2	Gap Found	900	1000

L1 looks out for its unit of work from Journal Log. It selects a dummy record in the Journal/Ledger Lock table where PROC_ID = PROCGAPS and JRNL_LDGR_ID = 1 and locks it. Meantime, if another instance of Ledger Engine, say L2 is scheduled in process gaps mode and it is also in the process of finding the next unit of work from journal 1 then it sees that the dummy record in the Journal/Ledger Lock table is already locked by L1 and it waits until L1 releases it. L1 goes on selecting Gap Found entries for journal 1 till the selected DOC_UNIDs sum up to Select block Size. L1 selects UNID 1, finds the number of DOC_UNIDs selected is (200 100 + 1 = 101), compares it with Select Block Size, finds (101<200), hence marks UNID 1 as Gap Intended. It then further selects UNID 2, finds the number of accumulated DOC_UNIDs as (101 + (450 400 + 1) = 152), compares it with Select Block Size, finds (152<200), hence marks UNID 2 as Gap Intended. It then further selects UNID 3, finds the number of accumulated DOC_UNIDs as (152 + (600 500 + 1) = 253), compares it with Select Block Size, finds (253>200), finds it has exceeded the Select Block Size, hence does not intend to process UNID 3.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Gap Intended	100	200
2	LDGRPOST	1	Gap Intended	400	450
3	LDGRPOST	1	Gap Found	500	600

UNID 1 and 2 form a single unit of work for L1. L1 then releases the lock on the dummy record on the Journal/Ledger Lock table. Since L1 has already released the lock on the dummy record, L2 can grab the lock and books its unit of work in a similar fashion as explained for L1. L2 can mark UNID 3 as Gap Intended and book its unit of work as shown below and continue further. Note that the example of L2 is explained to clarify how parallel runs work in coordination in terms of picking non-overlapping Units of Work.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
3	LDGRPOST	1	Gap Intended	500	600

Coming back to L1, L1 now attempts to ledgerize DOC_UNIDs specified in UNIDs 1 and 2. L1 invokes a Ledgerization Loop for UNID 1. Suppose DOC_UNIDs 200-250 had journal records that summed up to the Commit Block Size(400) then the original unit of work UNID 1 is split. A new JRNL_LOG entry UNID 5 is logged to record the work that could be finished in the first Ledgerization Loop(DOC_UNIDs 200-250). Then the BGN_DOC_UNID of original Intended entry UNID 1 is updated to DOC_UNID 251 so that UNID 1 is logically marked as pending work that will be processed by L1 in the next Ledgerization Loop.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Gap Intended	251	300
5	LDGRPOST	1	Gap Intended	200	250

L1 ledgerizes DOC_UNIDs 200-250. If it was successful, it marks its intended entry UNID 5 as Gap Final. If it failed, then it marks UNID 5 as Gap Failed.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Gap Intended	251	300
5	LDGRPOST	1	Gap Final	200	250

Then L1 invokes second Ledgerization Loop to process the pending Work UNID 1 (DOC_UNIDs 251-300). L1 ledgerizes DOC_UNIDs 251-300 and marks the intended entry UNID 1 as Gap Final. L1 found DOC_UNIDs 270, 271 and 280 were not filled in during ledgerization, it then logs these DOC_UNIDs as Gap Found.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Gap Final	251	300
6	LDGRPOST	1	Gap Found	270	271
7	LDGRPOST	1	Gap Found	280	280

L1 then invokes a third Ledgerization Loop to process the pending Work UNID 2 (DOC_UNIDs 400-450). L1 ledgerizes DOC_UNIDs 400-450 and marks the intended entry UNID 2 as Gap Final.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
2	LDGRPOST	1	Gap Final	400	450

At this point L1 had finished ledgerizing its original unit of work (DOC_UNIDs 200-300 and DOC_UNIDs 400-450) and now looks for the next unit of work(DOC_UNIDs > 450). It then tries to lock the dummy record as explained earlier and continues Ledgerizing. When it has finished Ledgerizing all Gap Found entries for journal 1, it proceeds to ledgerize Gap Found entries for journal 2 in a similar fashion and so on.

Rebuilding Mode

Single Ledger Rebuild is comprised of two steps:

Step 1:

Run the Rebuild Ledger Process when no instance of Ledger Engine is running. This is a separate process that needs to be run once for every Ledger to be rebuilt. It is scheduled from a different Batch Catalog entry appearing as Rebuild Ledger under node Posting (Batch Jobs).

This will initiate a Ledger rebuild and write an Instruction record into journal Log (JRNL_LOG) defining the range of DOC_UNIDs that should be ledgerized into the single

Ledger being rebuilt. Refer to the runsheet (page help) for the Rebuild Ledger process for help on scheduling the Rebuild Ledger process.

Step 2:

Once the Rebuild Ledger Process is run, schedule multiple instances of the Ledger Engine in Continue Ledger Rebuild mode for each Ledger being rebuilt. These Ledger Engine runs will then read the Instruction entry logged by the Rebuild Ledger Process and ledgerize the DOC_UNIDs lying in that range into the single Ledger being rebuilt.

This section explains the Continue Ledger Rebuild mode (Step 2).

The Continue Ledger Rebuild mode finds the existing rebuild instructions with the status Intended for the given ledger ID (specified with Rebuild Ledger ID parameter) and continues rebuilding the identified range. The rebuild instruction is logged by THE Rebuild Ledger process (Step 1). While rebuilding the Ledger, it skips processing DOC_UNIDs that are marked with Gap Found, Gap Stale or Gap Intended status. **This mode should not be run when there are any other instances of Ledger Engine running in Reprocess Failed Work Mode or Reprocess Failed Rebuild Work Mode or Process Logged Gaps modes.** This mode does not log any intermittent missing DOC_UNIDs as Gap Found.

Let us assume that a Ledger Engine instance (L1) is scheduled in *rebuild* mode for Ledger ID 15. Select block Size is 100 and Commit block Size is 200. Before running this mode, the journal Log has the following records immediately following a Rebuild Ledger process that added the Rebuild Instruction UNID 4:

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Final	100	200
2	LDGRPOST	1	Gap Found	170	180
3	LDGRPOST	1	Final	201	300
4	LDGRINSTRC	15	Intended	100	300

L1 first reads the Rebuild Instructions UNID 4 that has been logged by Rebuild Ledger process. L1 notes that journal 1 is the Source journal for Ledger ID 15 from which it summarizes. L1 then selects a dummy record in journal/Ledger Lock table where PROC_ID= LDGRRBLD and JRNL_LDGR_ID=15 (Ledger ID being rebuilt) and locks it. Meantime, if another instance of Ledger Engine say L2 is scheduled in rebuild mode for same Ledger ID 15 and it is also in the process of finding next unit of work from journal 1 then it sees that the dummy record in journal/Ledger Lock table is already locked by L1 and it waits till L1 releases it. L1 notes that it has to start processing from BGN_DOC_UNID 100 of the Rebuild Instructions. L1 books its unit of work as (100 + Select block Size = 100 + 100 = 200). L1 determines that it has to work on DOC_UNIDs 100 to 200. It checks for any Gap related entries with Status as (Gap Found or Gap Stale or Gap Intended) in this DOC_UNID range of 100-200 and finds UNID 2 that is a Gap Found entry with DOC_UNIDs 170-180. Hence L1 re-defines its Unit as Work as DOC_UNIDs 1-169 because this mode does not process DOC_UNIDs marked with Gap Found or Gap Stale or Gap Intended status. L1 adds UNID 5 to intend its unit of work(DOC_UNIDs 100-169) and releases the lock on the dummy record.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
5	LDGRRBLD	15	Intended	100	169

Since L1 has already released the lock on dummy record, L2 can grab the lock. L2 sees the next unit of work available is UNID 3 (DOC_UNIDs 201-300) and determines its unit of work as (201 + Select block Size of L2(100)) as shown below and continues further. Note that example of L2 is explained to clarify how parallel runs work in coordination in terms of picking non-overlapping Units of Work.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
3	LDGRRBLD	1	Intended	201	300

Coming back to L1, L1 now attempts to ledgerize DOC_UNIDs 100-169. Suppose DOC_UNIDs 100-150 had journal records that summed up to Commit Block Size(200) then the original unit of work is split. A new JRNL_LOG entry UNID 6 is logged to record the work that could be finished in the first Ledgerization Loop(DOC_UNIDs 100-150). Then the BGN_DOC_UNID of original Intended entry UNID 5 is updated to DOC_UNID 151 so that UNID 5 is logically marked as pending work that will be processed by L1 in the next Ledgerization Loop.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
5	LDGRRBLD	15	Intended	151	169
6	LDGRRBLD	15	Intended	100	150

L1 ledgerizes DOC_UNIDs 100-150. If it was successful, it marks its intended entry UNID 6 as Final. If it failed, then it marks UNID 6 as Failed.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
5	LDGRRBLD	15	Intended	151	169
6	LDGRRBLD	15	Final	100	150

Then L1 invokes the second Ledgerization Loop to process the pending Work UNID 5 (DOC_UNIDs 151-169) and after ledgerizing them marks the intended entry UNID 5 as Final.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
5	LDGRRBLD	15	Final	151	169

At this point L1 had finished ledgerizing its original unit of work (DOC_UNIDs 100-170) and now looks for the next unit of work (DOC_UNIDs > 180, skipping Gap entry too). It then tries to lock the dummy record as explained earlier and continues Rebuilding the Ledger. The instance that ledgerizes the last DOC_UNID(300) in the Rebuild instructions, set the Status Flag of Rebuild Instruction entry as Final.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
4	LDGRINSTRC	15	Final	100	300

Process Failed Work Mode

Process Failed Work attempts to reprocess the earliest work marked as Failed that was logged during Normal mode and work marked as Gap Failed during Process Logged Gaps mode. The failed work is for a specific range of journal records, so ledgerization attempts to ledgerize the range of journal records against all applicable active ledgers.

Let us assume that a Ledger Engine instance (L1) is scheduled in *process failed work* mode. Select Block Size is 100 and Commit Block Size is 200. Before running this mode, the journal Log has the following records:

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Failed	100	200
2	LDGRPOST	1	Gap Found	170	180
3	LDGRPOST	1	Failed	201	300

L1 selects a dummy record in the Journal/Ledger Lock table where PROC_ID= POSTFAILED and JRNL_LDGR_ID=1 and locks it. Meanwhile, if another instance of Ledger Engine, say L2 is scheduled in Reprocess Failed Work mode and it is also in the process of finding next unit of work from journal 1 then it sees that the dummy record in the Journal/Ledger Lock table is already locked by L1 and it waits until L1 releases it. L1 picks up the first Failed Work UNID 1 (DOC_UNIDs 100-200). L1 marks UNID 1 as Intended to book its unit of work(DOC_UNIDs 100-200) and releases the lock on the dummy record. If UNID 1 had a Gap Failed status then L1 would have marked it as Gap Intended.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Intended	100	200

Since L1 has already released the lock on the dummy record, L2 can grab the lock. L2 sees the next unit of work available is UNID 3 (DOC_UNIDs 201-300) and books its unit of work as Intended shown below and continues further. Note that the example of L2 is explained to clarify how parallel runs work in coordination in terms of picking non-overlapping Units of Work.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
3	LDGRPOST	1	Intended	201	300

Coming back to L1, it has marked DOC_UNIDs 100-200 as Intended. L1 invokes a Ledgerization Loop. L1 determines its unit of work for its first Ledgerization Loop as (100 + Select block Size = 100 + 100 = 200). L1 determines that it has to work on DOC_UNIDs 100- 200. It checks for any Gap related entries with a Status as (Gap Found or Gap Stale or Gap Intended or Gap Final or Gap Failed) in this DOC_UNID range of 100-200 and finds UNID 2 that is a Gap Found entry with DOC_UNIDs 170-180. Hence L1 re-defines its Unit as Work as DOC_UNIDs 1-169 because this mode does not process DOC_UNIDs marked with Gap Found or Gap Stale or Gap Intended or Gap Final or Gap Failed status. L1 splits up its Original unit of work. A new JRNL_LOG entry

UNID 4 to push the remaining pending Failed Work of DOC_UNIDs 181-200 (Gap entry is skipped). L1 updates the END_DOC_UNID of UNID 1 to 169.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Intended	100	169
4	LDGRPOST	1	Failed	181	200

Failed Work of UNID 4 can be picked up by any concurrent instances of the Ledger Engine running in Reprocess Failed Work mode or by the current instance in the next Ledgerization Loop. L1 continues down its path to work with DOC_UNIDs 100-169. Suppose DOC_UNIDs 100-150 had journal records that summed up to Commit Block Size(200) then the original unit of work is split again. Then the END_DOC_UNID of original Intended entry UNID 1 is updated to DOC_UNID 150 and a new entry UNID 5 is added to push the remaining Failed Work. Failed Work of UNID 5 can be picked up by any concurrent instances of Ledger Engine running in Reprocess Failed Work mode or by the current instance in the next Ledgerization Loop.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Intended	100	150
5	LDGRPOST	1	Failed	151	169

L1 ledgerizes DOC_UNIDs 100-150. If it was successful, it marks its intended entry UNID 1 as Final. If it failed, then it marks UNID 1 as Failed. If UNID 1 had a status of Gap Failed then L1 must have updated it to Gap Final (if successful) or Gap Failed (if Failed again). L1 found DOC_UNIDs 140, 141 and 147 were missing during ledgerization, it then logs these DOC_UNIDs as Gap Found.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Final	100	150
6	LDGRPOST	1	Gap Found	140	141
7	LDGRPOST	1	Gap Found	147	147

Then L1 looks for New Unit of Failed Work (DOC_UNIDs > 150).

Process Failed Rebuild Work Mode

Process Failed Rebuild Work attempts to reprocess the earliest rebuild work (for a given ledger) marked as failed during a rebuild mode. If the JRNL_LOG has the following records before invoking ledgerization, the Reprocess Failed Rebuild Work mode reprocesses the work with UNID 3.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRINSTRC	15	Intended	100	500
2	LDGRRBLD	15	Final	100	200
3	LDGRRBLD	15	Failed	201	300

Then the process follows the same rules as *process failed work*, except that it only processes against the given ledger and it tries to lock the dummy record on the Journal Ledger Lock table with PROC_ID = RBLDFAILED and JRNL_LDGR_ID = Ledger ID being rebuilt. It does not log any intermittent missing DOC_UNIDs as Gap Found.

2.1.22 Open Activity and Budget Roll

Chain Job Name	Open Actv & Budget Roll
Recommended Frequency	To be run as the end of the Budget Fiscal Year approaches, is reached, or has passed for a given amount of time. The timing of when the chain is run depends on client needs, which can even vary by the type of accounting activity so that one type is rolled before another. Three batch programs should be run before this chain. • The New Year Table Initialization (NYTI) program is one that is critical to the success of a roll. Refer to the NYTI run sheet in the *CGI Advantage Financial – Utilities Run Sheets* guide for more information. • Another is the Matching chain job so that any purchase orders that have met matching rules for being received and/or invoiced are closed out with a payment request and not rolled. Refer to the Matching Payment Creation run sheet in the *CGI Advantage Financial – Procurement Run Sheets* guide for more information. • The last is the Systems Assurance 15 program to ensure the open amounts found on accounting lines is accurate. Refer to the System Assurance 15 run sheet in the *CGI Advantage Financial – System Assurance Run Sheets* guide for more information.
Single Instance Required	Yes
Can be restarted?	See individual jobs for more restartability information
Reports generated	Report is generated for transactions matching selection criteria. An exception report if there is error during submitting the rolled transactions.

Overview

When a year comes to a close, there are many transactions still un-liquidated with activity that has not yet reached its final state. The Open Activity Roll chain job takes the activity in the old budget fiscal year forward into the next year through a transaction modification, often referred to as 'rolling'. When activity is rolled, the program generates the necessary transaction modification, changing the existing BFY on selected accounting lines to the 'Target Year' BFY, which is specified in parameter setup on the Parameters for Roll Process (RLPA) page. An optional part of that modification would be COA changes made on selected accounting lines that match records of a specified COA Crosswalk (COAX) ID.

The Open Activity and Budget Roll chain job also rolls the accounting activity but combines that functionality with the rolling of budget authority to cover the accounting activity. When budget authority is rolled along with the accounting activity, budget availability in both the old and new year will not change. This is because an amount equal to the decreases of accounting activity on the old year will be used to decrease Current Budget in the old year. Likewise, an amount equal to the accounting activity recorded in the new year will be used to increase the Current Budget of the new year. The budget amount of Reversions is used to record the amount of budget reduced on the old year. Reversions are then subtracted in the formula for Current Budget. Carry Forward is used to record the amount of budget increased in the new year. Carry Forward is then

added in the formula for Original Budget. Reversions and Carry Forward are the delivered budget amounts specifically for this purpose. A given site may choose to change the buckets used and may choose to change the formulas for Original and Current Budget. However, if the amounts used to increase and decrease budget availability are not included in the formulas or constraints that define budget availability, then the budget roll functionality will only serve to record an amount of accounting activity rolled forward and will not change budget availability.

The Open Activity and Budget Roll chain job can roll several types of activity. What activity is a function of specifying one or more event types on RLPA. The Fund (FUND) page does contain several override controls for use when RLPA selection is not based on fund codes.

Accounting Activity	Examples
Pre-Encumbrances	Event types with a posting code defined to update the Pre-Encumbrances (12) budget amount. PR02, PR03, and ST03 are three such delivered event types.
Encumbrances	Event types with a posting code defined to update the Encumbrances (13) budget amount. PR05, PR06, and ST04 are three such delivered event types.
Receivables	Event types with a posting code defined to update the Billed Earned Revenue (22), Unbilled Earned Revenue (23), or Billed Unearned/Deferred Revenue (25) budget amount. AR01, AR06, and AR10 are three such delivered event types.
Other Items	Event types with a posting code defined to update no budget amount, while not also defined to make a Disbursement Request update. PR01, PR07, AR20, AR50, AR52, and AR54 are six such delivered event types. Event type without posting codes. PR08, ST01, and ST02 are three such delivered event types.
Accrued Expenditures for Vendor Refunds	When rolling receivables, there is an option to roll those for vendor refunds that used the AR30 event type (or similar clone). The AR31 event type falls into the Other Items grouping above. In order to roll the expenditure vendor refunds, the Allowed Roll Budget Amounts (ALW_ROLL_BKTS) record on Application Parameter must be updated with the ID of 14 before event type AR30 can be saved on RLPA.

Note – The system looks only at posting pair A for an event type to see what type of accounting activity is booked.

Such activity recorded on the Requisition (RQ), Purchase Order (PO), Accounting Based Spending (ABS), Receivable (RE), Accrued Receivable (ARE), Stock Requisition (SRQ), and Travel (TRVL) transaction types can be selected. If the Allow Fund Action flag for the type of accounting activity is checked on the RLPA Parameter ID used, then the Close Action on the parameter will be read, which is found on Fund (FUND). If the FUND option is other than *Roll* then no open activity matching RLPA selection criteria will be selected for that fund code.

Further selection is based on the Roll Minimum of the activity. If an accounting line with the open activity equals or exceeds that minimum, it will be selected. The same override is available for the minimum on FUND as well. When the header and not the accounting line should be the point of comparison between the Open Amount and Roll Minimum, the Transaction/Line field for each action should be set to *Transaction.*

Notes:

1. If RLPA selection criteria can select a subset of open accounting lines on a transaction, then the Transaction/Line field should be Line to avoid an incorrect comparison of the header Open Amount to the Roll Minimum.
2. Travel Authorization (TRAUTH) is the only Travel Transaction Sub Type within the Travel (TRVL) Transaction Type that can be rolled.

Each instance of the chain is associated with a single record on RPLA that provides selection criteria as well as some of the date information to be placed on the transaction modifications. Other date information will come from the BACKOUT APD and BACKOUT FY parameters on Application Parameters (APPCTRL).

An additional parameter page is used by the Open Activity and Budget Roll chain. The Parameters for Budget Roll (PBRP) page is read by a job late in the chain to determine what budget structure is to have budget availability rolled. PBRP also supplies information for budget transaction creation, event types to be used, date information to be placed on the transactions, and if any special budget features should be rolled as well, such as allotments.

At a high level, the Open Activity and Budget Roll chain (commonly called OABR) selects an open accounting line to roll belonging to an individual BFY and not multi-year lines with 9999, creates a modification draft with changes to the accounting line to include the BFY and any COA values cross-walked, and submits that modification. The budget line for last BFY is decreased if the rolled activity was budgeted. All accounting impacts from the open amount are removed from the prior fiscal year. If a matching budget line exists in the new year, it will be updated. If a budget line does not exist in the new year, one will be automatically generated by the processing of the modification. All budget control errors will be suppressed at this time because any generated line will have a negative unobligated amount. Then, if the rolled activity is budgetary, the program will revert out an amount equal to that rolled from last BFY line and carry it forward to the new BFY line. After the processing of the budget transactions, all budget availability created from rolling out of the old BFY will be removed and carried forward into the new BFY so availability there equals what it was before the roll.

An alternate method of OABR exists where the chain is run with an RLPA ID that has the Reorg flag checked. This method does a mid-year reorganization of COA according to the required COAX ID specified into the chain. What does not occur is a changing of the BFY value.

The chain process is comprised of the following batch jobs below that are given with very brief summaries. Parameters have to be entered at job #1 and Job #8 only. More details can be found later in the individual job steps.

Job 1 **Roll Update Preprocess**

Batch parameter validation is performed. Selection of records from either the Detail Pre Selection table or directly from the transaction catalogs is next. Finally, the job when run in update mode only, the workload table is loaded with selected records to guide later processing.

Job 2 **Create Round 1**

The workload table is reviewed for activity that belongs in round 1. That activity is grouped according to parameters that limit the size of work to be loaded. These groups are then logged in a facilitator table. An Open Activity Roll Update job is then spawned for each unit of work. When those spawned jobs have finished, the create job will finish with a report of those transaction modifications.

Job 3 **Submit Round 1**

The workload table is reviewed for activity that belongs in round 1. That activity is grouped according to parameters that limit the size of work to be submitted. These groups are then logged in a facilitator table. A System Maintenance Utility job is then spawned for each unit of work. An extra is spawned if there was a transaction that failed

to submit previously as a cleanup step. If there is no round 1 activity, then there is no spawning. When those spawned jobs have finished, the submit job will finish.

Job 4 **Exception Report for Round 1**

A report is compiled to list those transactions that failed to submit from the one or more spawned SMU jobs.

Job 5 **Create Round 2**

The workload table is reviewed for activity that belongs in round 2. That activity is grouped according to parameters that limit the size of work to be loaded. These groups are then logged in a facilitator table. An Open Activity Roll Update job is then spawned for each unit of work. When those spawned jobs have finished, the create job will finish with a report of those transaction modifications.

Job 6 **Submit Round 2**

The workload table is reviewed for activity that belongs in round 2. That activity is grouped according to parameters that limit the size of work to be submitted. These groups are then logged in a facilitator table. A System Maintenance Utility job is then spawned for each unit of work. An extra is spawned if there was a transaction that failed to submit previously as a cleanup step. If there is no round 1 activity, then there is no spawning. When those spawned jobs have finished, the submit job will finish.

Job 7 **Exception Report for Round 2**

A report is compiled to list those transactions that failed to submit from the one or more spawned Open Activity Roll batch jobs.

Job 8 **Budget Roll**

An XML file is created of budget transactions at the lowest required budget level of detail to revert from the old BFY and carry forward into the new BFY. Information for this budget roll is gathered from a table that summarizes all budget updates made by the previous roll modifications.

Job 9 **Load Budget Transactions**

The XML file created from the previous job is loaded.

Job 10 **Perform Smart Budget Rollup & Submit Budget Transactions**

As the loaded budget transactions have only the lowest required level of detail, they are rolled up to complete higher levels and submitted.

Job 11 **Budget Roll Reports**

Reports are produced listing the budget transactions successfully processed and those that failed to submit to final.

Job 12 **Post Processing**

The Required Budget table is restored to what it was prior to the chain job.

If using an RLPA ID with a Run Mode of *Pre Selection*, only job #1 of the chain is needed. All others should be disabled. Please remember to check all of the Disable flags, including the 2 on the next page, and hit save to fully disable all other jobs in the chain. If not, job #2 will fail, but that is really an acceptable outcome as there is nothing to load. If using an RLPA ID with a Run Mode of *Update*, usually the whole chain should be run, but there are additional details on this below in the Advanced Run Instructions. If using an RLPA ID with a Run Mode of *Report*, do not use the chain job. There is a job in the Reports-GA folder of General Accounting called Open Actv Roll

that should be used. If the chain is used, it will fail on the first job step with an error stating that the Run Mode was incorrect for the chain.

Pre-Processing Tasks:

1. The System Application Parameter OABR Running **must** be set to "true" for this chain job to run. When it is set to "true", no other system activity should be allowed other than the processing of this chain. Bounce all Versata Logic Servers (VLS) that are pointing to the production database to enable the change. At the end of the processing: (1) Ensure that Job 12 Post Processing has run successfully. (2) Reset System Application Parameter OABR Running to "false". (3) Bounce all VLS that are pointing to the production database to enable the change.
2. The System Application Parameter OABR Source BFY **must** be set to match the Closing BFY in the Parameters for Roll Process (RLPA) table. This is a necessary step since the program uses this parameter when determining how to auto – generate any required allotment lines in the Target BFY. Bounce all Versata Logic Servers (VLS) that are pointing to the production database to enable the change.
3. Sites **should** ensure that Target Year REQBUD entries are established, if not already, for the budget lines that are to be rolled. This is not an issue for those budget structures that are mandatory, but one that may need to be done for optional budget structures. When Round 1 and/or Round 2 transactions are submitted that have budget updates, the OABR program makes use of “Auto – Generation” for both budget lines and any required allotment lines. If REQBUD entries are not established for the target year budget lines which are being rolled, then Round 1 and/or Round 2 transactions will submit without target year budget lines in place.
4. Sites have the ability to specify the Accounting Period and Fiscal Year to be recorded on activity being backed out of the prior BFY. The APPCTRL parameters for BACKOUT_APD and BACKOUT_FY **must** be set if the roll is to happen after the beginning of the new fiscal year. If the next year has not started and the Transaction Record Date parameter on RLPA is blank or not dated to the next fiscal year, the system will default the current fiscal year and accounting period as normal. Please remember to change the BACKOUT FY each year as the BACKOUT APD will not likely change.
5. For the target year posting lines, the Modification Fiscal Year and Modification Accounting Period on RLPA are used. If not specified, the job will use the Transaction Record Date on RLPA to determine these values. If the Transaction Record Date value is blank, the job will use the current system date to determine these values. If rolling into a fiscal year that has not yet started (rolling on the last day of the prior year), these values **must** be set.
6. Whether rolling before the end of a year or after the new year has started, sites must ensure the proper Transaction Control (DCTRL) settings for each transaction code are set to allow either future or past transaction date, accounting period, and fiscal year.
7. For maximum performance, the COA Crosswalk table **could** be placed in Service Data Object (SDO) cache. This must be done for all Versata Logic Servers (VLS) that are pointing to the production database, followed by a bounce of each. The table must be removed from SDO cache and each VLS bounced after the processing is complete. This action is helpful if many combinations are being cross-walked.
8. For maximum performance when specifying RLPA selection parameters, a non-unique index **could** be added to the DOC_ACTG table.
9. Sites **should** try to select only activity that will be in round 1 or 2. Mixing the two is possible, but can become confusing if the chain job stops processing.
10. If a site is to control rolling the same for all funds, then they **should** ensure that all the override flags on RPLA are unchecked to prevent Fund lookups that will return the same value or worse, different values that are not intended.

11. Sites **should** run the Open Activity Roll report, setting up the Run Mode parameter to report only for the RLPA ID. The report provides an activity count to gauge run times by displaying the open accounting lines that will be rolled. One of the fields on the report also shows if a pending version of any final transactions to be rolled exists. Final transactions with Pending versions are skipped by OABR. The Open Activity Roll report has a setting that will generate a text file (*.txt) of all transactions in a Pending state, which can then be input into a SMU job to reject all pending transactions back to Draft. When this is done before a roll, all transactions will be rolled and the pending ones will not be skipped.
12. For sites where the selection criteria on RLPA is not sufficient, sites **should** run with the Pre selection mode by setting up the Run Mode parameter to Pre selection and run only the first job step of the OABR process. Using these tables, a user can decide which transactions or separate accounting lines to roll. When the OABR process is then run in update mode and the Pre-selection table that was created during the pre-selection mode is to be used as input, sites should remember to set the USE_PRETABLE batch parameter to True or else the transaction catalogs will be used.
13. If the COA Crosswalk (COAX) feature is to be used, sites **should** setup all entries before rolling. Transaction codes being cross-walked should be setup on Transaction Control (DCTRL) with the Change Closed Allowed flag checked.
14. Negative and zero-dollar accounting lines that are open **will not be rolled**. They will have to be dealt with manually.
15. Accounting lines with the multi budget year of 9999 **will not be rolled**.
16. To ensure fewer transaction modification failures, sites **must** ensure the transaction codes being rolled are established on Transaction Control (DCTRL) with the following flags checked: Approval Override Allowed and Inactive COA Codes Allowed.
17. For better performance of the budget roll portion, sites **should** set a value in the Transaction Break field on PBRP that will ensure a balance between too many budget transactions and transactions with too many budget lines.
18. Deactivated transactions **should** be reactivated before rolling, as they will cause the program to fail. If reactivation is not appropriate so that they can be rolled, then they should be reactivated and then cancelled. The reason is that a deactivated transaction cannot be modified.
19. This batch program produces transactions which are subject to the line limit functionality constraints defined on the Transaction Component Requirements (DCREQ) in the Administration application. Sites **must** ensure that limits have not been set to lower limits than existing transactions being rolled contain. Additionally, if there is a limit for budget transaction lines, then the Transaction Break parameter on PBRP should be set to ensure that limit is not exceeded.

Common Run Instructions:

1. Run first in a test environment that is a recent copy of production to shake out setup problems and other system configurations that can cause modifications to fail.
2. Establish parameter ID on the RLPA and PBRP tables. There is the option of setting these up when scheduling the chain job by using the Setup Custom Parameters link found with job steps 1 and 8. However, as sites that schedule the chain and those that determine rolling rules are often different, establishing the table records outside of the chain job is recommended. This step often requires only copying the prior year's record or making changes to that record, updating the Closing BFY, Modification Fiscal Year, Modification Accounting Period, and Transaction Record Date on RLPA. For PBRP, updates to the Source BFY, Target BFY, Transaction Record Date, Source Fiscal Year, Target Fiscal Year, Source Accounting Period, and Target Accounting Period.
3. Double check the controls on FUND are correct for the FY value equal to the Closing BFY on RLPA.

4. Run the Open Activity Roll Report with the RLPA parameter that will be used for the actual roll (making sure to adjust the run mode to Report) as an optional step to ensure activity that is desired is all that well be selected and to get a load estimate.
5. For Pre-Selection mode, disable all the jobs except first one, since only first job is required for this mode.
6. On Application Parameters (APPCTRL), set OABR_RUNNING to true and OABR_SRC_BFY to match the Closing BFY in the Parameters for Roll Process (RLPA) table. Stop and restart all VLS that are connected to the database. Do not forget to set it back to false when done, and start/stop all VLS again.
7. A good practice is to query for any remaining open activity that should have rolled. This can be done from a variety of sources: ledgers, journals, and the BBAL Fiscal Year Details (BBALFY) page. If ledgers are to be used, then ensure the Ledger Engine has run to ledgerize all rolling activity.

Advanced Run Instructions:

If any of the situations below do not apply, submit the chain job with all steps enabled, having entered an RLPA ID and PBRP ID in jobs 1 and 8 respectively.

1. When not all budget lines should have authority rolled forward for the selected accounting lines:
    A. The chain job 1st should be run with steps 8 to 12 disabled. The Jobs 3 and 6 will have populated the Summarized Budget Activity (SBA) table with those budget lines updated by the rolled accounting activity. Only budget lines at the lowest required budget level will be shown and only those in the Target Budget Fiscal Year. All records will be loaded as *Selected*.
    B. Sites should uncheck the selected flag and save each record that should not be rolled.
    C. A 2nd chain will then be submitted that executes only job steps 8 to 12. Only those budget lines marked as *Selected* will be placed on a budget transaction and then rolled up. There should not be any instance of the Open Activity and Budget Roll chain with any other roll parameter (starting from job 1) scheduled between the time spent on the two activities.
2. When budget transactions for more than one budget structure have to be created for the rolled transactions, for example, Budget Structure 31 and 32:
    A. Ensure Required Budget (REQBUD) records exist in the new BFY for structures 31 and 32 if they do not already.
    B. 2 budget roll parameter records are defined on PBRP. Parameter Id A having a Budget Structure value as 31 and Parameter Record B having a Budget Structure value as 32.
    C. Schedule the chain with jobs 1 to 7 enabled and disable jobs 8 - 12. Provide the Budget Roll Parameter as A in the Budget Roll job.
    D. After the 1st chain process is completed, schedule another chain with jobs 8 to 12 enabled and disable the rest. Provide the Budget Roll Parameter as B in the Budget Roll job.
3. When rolling activity on a budget structure that has at least one budget level as Presence Optional (setting on Budget Level Update which is found from the BUDST page) , but that level is used by a group that can be identified by Event Type, Transaction Code, Fund, Department, Appropriation, Program, or Appropriation Type:
    A. Rolling activity for the group of sites that use the presence optional level must be done first by having an RLPA record that will select only open activity for that group.

B. On the Budget Level Update page, that optional budget level or levels must be changed to not be presence optional. A start and stop must be done to all VLS instances so the Budget Flex Server will get the new setting.
C. The chain job is run with that specific RLPA record with job 8 - 12 disabled.
D. On the Budget Level Update page, the budget level or levels are now changed back to presence optional. A start and stop must be done to all VLS instances so the Budget Flex Server will get the new setting.
E. The complete chain job is then run with a 2nd RLPA entry that will select all other applicable accounting activity. Since all activity for that group has been rolled with the 1st chain job, none for that group will be available for selection.

Major Input

- Transaction Accounting Line Catalog (DOC_ACTG)
- Budget Activity Summary (SBA / RL_BUD_SUMM)
- Budget Activity Detail (RL_BUD_DTL_AM)
- OABR Facilitator (OABR_FACILITATOR)
- OABR Workload (OABR_WRKLD)
- Roll/Lapse Pre Selection Detail (RLPSD / RLLP_PRE_DET)
- Budget Roll Parameter (PBRP / BUD_ROLL_PARM)
- Roll Parameter (RLPA / RL_PROC_PARM)

Peripheral Input

- Application Parameters (APPCTRL / IN_APP_CTRL)
- Fund (FUND / R_FUND)
- Posting Code (PSCD / R_PSCD)
- Event Type (ETYP / R_EVNT_TYP)
- Auto Transaction Numbering (ADNT / AUTO_DOC_NO)
- Fiscal Year (FY / R_FY)
- Transaction Control (DCTRL / R_GEN_DOC_CTRL)
- Department (DEPT / R_DEPT)
- Appropriation (APPR / R_APPR)
- Appropriation Type (APTYP / R_APTYP)
- Program (PROG / R_PROG)
- Accounting Period (APD / R_APD)
- Transaction Header (*_DOC_HDR)
- Transaction Vendor (*_DOC_VEND)
- Transaction Commodity (*_DOC_COMM)
- Transaction Accounting (*_DOC_ACTG)

- Budget Structure (BUDST / GN_BUD_STRU)
- Budget Level Options (GN_BUD_LVL)
- Budget Allotment Options (GN_ALOT_OPT)
- Budget Structure Level Inquiry (BQ*LV* / BUD_STRU_*_LVL_*)
- COA Crosswalk (COAX / R_COA_CROSSWALK)

*Items above with a * denote that different values may be substituted in place of the *.*

Major Output

- Budget Activity Summary (SBA / RL_BUD_SUMM)
- Budget Activity Detail (RL_BUD_DTL_AM)
- OABR Facilitator (OABR_FACILITATOR)
- OABR Workload (OABR_WRKLD)
- XML file of Budget Transactions (OABR*.xml)
- Modifications transactions (Common & specific transaction catalogs)
- Updated budget lines (BQ*LV* / BUD_STRU_*_LVL_*)
- Updated allotment lines (A LOT_STRU_*_LVL_*)
- Updated Accounting Journal (JACTG / JRNL_ACTG)
- Updated Budget Journal (JBUD / JRNL_BUD)
- Roll/Lapse Pre Selection Summary (RLPSS / RLLP_PRE_SUM)
- Roll/Lapse Pre Selection Detail (RLPSD / RLLP_PRE_DET)
- Rolled Transaction Report per round
- Rolled Transaction Detailed/Summarized Exception Report per round
- Summarized Budget Activity Lines Not Loaded Report
- Budget Roll Crosswalk of Successful Transactions Report
- Budget Roll Listing of Unsuccessful Transactions Report

Chain / Job Return Code

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

Return Code	Condition
Successful (1)	All of the jobs end successfully.
Warning (4)	One of the jobs in the chain 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 on each job step, but several are presented here at the chain job level because they can occur at multiple points in the chain.

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process was discontinued for some reason (return codes 12, 16, or 20) the user has an option to restart the job and the process will continue from the point it left off. This restart has a condition that no roll process must be scheduled between the time frame the job was discontinued and is being restarted. The Restart option is provided for jobs – "Create Round n" and "Submit Round n" where n is 1, or 2 as well as the spawned jobs for those different Create and Submit rounds.

If Create Round 1 or 2 jobs successfully created drafts but the submit for that round encountered some transaction submission exceptions, then the user can view the log for the Submit for the round or the Roll Transaction Detailed/Summarized Exception Report generated by the Exception Report for the Round for the errors. In this example, the Exception Report for the round returns a Return Code of *Non-Fatal*, and further jobs down the chain are discontinued. If the errors were due to data setup, correct the errors first. Then, submit the rejected transactions manually via a new System Maintenance Utility (SMU) job or with the Automatic Transaction Corrections batch job because it is necessary to change COA values that were not caught with COAX setup. Finally, restart the Exception Report for the round job which will return a Return Code of *Successful*, thus allowing the rest of the chain to continue. It is critical to ensure that no parameters are changed on RLPA and no OABR or OAR process has been run between the actual run and restart run, else any *.txt files marked for system use are overwritten and the restart run will not work correctly.

In the case of a system failure during any Create Round or Submit Round, all spawned Create jobs (Open Actv Roll Update + <chain job id>) or spawned Submit jobs (System Maintenance Utility + <chain job id>) will return a Return Code of *Failed* or *System Failure*. The Submit Round or Create Round job will also return a Return Code of *Failed*, and further jobs down the chain are Inactive. In this case, the user should first restart each spawned job (Open Actv Roll Update or System Maintenance Utility) so that the individual job can complete work on the set of transactions it was assigned. Then after each spawned job completes successfully, the user should restart the Create Round or Submit Round job. If there is further work to be done in that Create or Submit Round, the chain will spawn additional Open Actv Roll Update or SMU jobs to complete that round's work. After all the applicable work is completed, the chain will automatically move on to the next job in the chain - Exception Report Round.

Application errors can occur on a roll as a result of options being changed or data being deleted from reference tables during a year. Such errors will cause modifications to fail. The source of these errors must be addressed. If online options can be changed, data put back, or other methods to make the errors go away, then that is the recommended approach. If not, then the severity of the error message must be turned down on the Message (MESG) table. Extreme caution should be taken when changing the severity of an error message as that may cause data integrity problems.

Open Activity and Budget Roll Chain: Roll Update Pre Processor

Job Name	Roll Update Preprocessor

Recommended Frequency	See chain job. May execute this single job step in a chain just for pre selection or as part of a larger chain with other job steps.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This first job step starts with basic editing of parameters and goes on to perform accounting line selection based on Run Mode;

> If *Pre Selection,* then the job will look to the transaction catalogs for open accounting lines that match RLPA and optionally FUND criteria. Selected accounting lines not already on RLPSD will be added. Open accounting lines already on RLPA will be updated for any information that has changed since the last update. If the RLPSD record has a Selection Date and the accounting line is still open, that date will be cleared. Any RLPSD records found that are no longer open with a blank Selection Date will be removed.
>
> If *Update*, the job uses the 'Use Pre Selection' batch parameter to determine where to look for open accounting lines. If that parameter is 1 – Use Pre Selection, then selection will be from the Roll Lapse Pre Selection Detail (RLPSD) page for records with an Action = *Roll*, a checked Approved flag, and a blank Selection Date. Even after all this selection, if the accounting line is no longer open, it will not be selected. Be aware that when using RLPSD as input, selection criteria on RLPA and FUND are not read, as they were used when RLPSD was loaded. Only the Roll Minimum Amounts are considered when selecting from RLPSD.

Data from RLPSD and RLPSS table will be truncated based on the value in specified in the PURGE_DATA parameter. If the value is *Approve Only*, it will truncate all approved data. If the value is *Approved Closed*, it will truncate data where Approve Flag is true and the open amount is equal to the closed amount. If the value is *All*, then it will truncate the entire table.

When the RLPA parameter ID is *Update*, after the selection of records work is listed on the OARB Workload (OABR_WRKLD) table with information that will be used by later jobs. This workload table contains information to record each accounting line to be rolled along with what round it will be rolled and what the amount being rolled is. Each record is assigned a sequence number (SEQ_NO) that will be used by the next job in the chain when forming units of work for the modification process.

This job step goes on to adjust the Require Budget (GN_BUD_RULE) table. In order to ensure modifications do not fail for the new BFY because a budget line does not already exist in that year, this job step stores off the existing Auto Generate flag (NO_CTRL_BUD_FL) into an OABR_NO_CTRL_BUD_FL field for the Target BFY and sets all records for the Target BFY to Auto Generate. The Post Processor job step at the end of the chain will move the values back.

Finally, the job step creates an OABRParms.txt file to pass necessary parameters to later job steps in the chain.

Process Steps	Messages
1. Parameter validation	• Batch input parameters are listed along with APPCTRL parameters. If any is invalid an error message is issued stating such.

2. Deletion of Roll/Lapse Pre Selection records	• Number of summarized records deleted from workload table: ## • (this is really the pre selection records but the message states workload table)
3. Selection of records	• Messages issued for the number of records selected to load in pre selection table.
4. Loading of Roll/Lapse Pre Selection records	• Number of summarized records deleted from workload table: ##
5. OABR Workload update	• Number of Workload records loaded for Round 1: ## • Number of Workload records loaded for Round 2: ## • Accounting lines to be rolled: ##
6. REQBUD update	• No messages issued for this step
7. Parameter file creation	• No messages issued for this step

Restartability Information

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

Major Input

- Transaction Accounting Line Catalog (DOC_ACTG)
- Roll/Lapse Pre Selection Detail (RLPSD / RLLP_PRE_DET)
- OABR Workload (OABR_WRKLD)
- Roll Parameter (RLPA / RL_PROC_PARM)
- Fund (FUND / R_FUND)
- Posting Code (PSCD / R_PSCD)
- Event Type (ETYP / R_EVNT_TYP)

Peripheral Input

- Application Parameters (APPCTRL / IN_APP_CTRL)
- Auto Transaction Numbering (ADNT / AUTO_DOC_NO)
- Fiscal Year (FY / R_FY)
- Accounting Period (APD / R_APD)
- Transaction Control (DCTRL / R_GEN_DOC_CTRL)
- Department (DEPT / R_DEPT)
- Appropriation (APPR / R_APPR)
- Appropriation Type (APTYP / R_APTYP)

- Program (PROG / R_PROG)

Batch Parameters

For the entire chain process, a user has to enter parameters for Job 1 – Roll Update Preprocess and then on Job 8 - Budget Roll only. The process propagates required parameters down the chained jobs. Some of them are non-overrideable parameters, which will be provided with the Job Setup and need not be specified for each run. Many of these are just meant for system use and should not be changed; those are noted in the Description column.

Parameter Name	Description	Default Value
AMSPARM	Parameter Location for Roll Update Preprocess Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	**$$AMSROOT$$/Parms**
CHAIN_PARM_FILE	Common Chain Parameters File (.txt). • For system use and should not be changed.	**RollParams.txt**
CLIENT_NM	Client name to be used for report headers produced from jobs in the chain.	**(blank)**
COMMIT_SIZE	Required commit block size for the transaction load and submit job steps. This figure determines the number of transactions to be committed at a time for all jobs in the chain that perform transaction load and submit action. It can be used for performance enhancements depending upon the server memory configuration.	**1**
EXCP_RPT_TYP	Exception report type for the exception report generated in the chain. 1 = Detailed & 2 = Summarized. If not entered, then defaults to 1.	**1**
JOB_BLOCK_SIZE	Number of accounting lines per SMU Facilitator Parameter file.	**100**
LOAD_DATA	How to load the Detail Pre Selection table. 1 = load all records with as approved (checked) or 2 = load all records unapproved.	**1**
OACTV_BUD_ROLL	Is class run as part of Open Activity & Budget Roll? For system use to differentiate from the Open Activity Roll. Should not be changed.	**YES**
PARALLEL_JOB_CO	Number of jobs for SMU Facilitator	**1**

UNT	to keep running.	
POLLING_TIME	Number of seconds to wait between polling occurrences.	**30**
PRE_SEL_COMMIT_BLOCK	Commit block size for loading pre selection tables. If not entered then defaulted to 1000. It is the number of records that are loaded into the tables after which a commit is issued. Can be used for performance tuning. Too low of a value can cause frequent database commits, thereby increasing processing time. Too high a value will demand for higher memory. Can be used for a performance enhancement depending upon the server memory configuration.	**1000**
PRE_SELECT_BLOCK	Select block size for loading the pre selection tables. If not entered then defaults to 1000. It is the number of accounting lines that will be fetched as a single block from DOC_ACTG. Too low of a value can cause higher number of database fetches, thereby increasing processing time. Too high of a value will demand higher memory. Can be used for a performance enhancement depending upon the server memory configuration.	**1000**
PURGE_DATA	How to purge the Detail Pre Selection. 1 to purge all records, 2 to purge only approved records and 3 to purge closed records only. Parameter is read only for Pre Selection mode. Use parameter 2 or 3 when 'refreshing' or loading additional data to RLPSD. Use parameter 1 for the first run of a new year.	**1**
RL_PROC_PARM_ID	Parameter ID to identify the RLPA record used. The Setup Custom Parameters link opens the RLPA page for record selection, update, or addition.	**(blank)**
ROLL_CATALOG_ID	Batch Job Catalog ID for the Open Activity Roll Update job. For use when employing customized jobs for transaction creation. For system use and should not be changed.	**1238**

SMU_CATALOG_ID	Batch Job Catalog ID for the System Maintenance Utility job. For use when employing customized jobs for transaction loading, budget rollup, and submitting. For system use and should not be changed.	**3**
SMU_FILE_PREFIX	SMU Facilitator parameter file prefix. For system use and should not be changed.	**OABR**
USE_PRETABLE	Use Detail Pre-Selection Table (1= Use Pre-Selection table as input, 2= do not use Detail Pre-Selection table but use transaction catalogs.") If pre selection is never used, change the default value of this to 2 on BATSETUP as the parameter is often overlooked.	**1**
XWALK_PROC_ID	Process Id to pick COA Crosswalk records.	**(blank)**

If you are running the job in the Financial application, you can click on the Custom Parameter link and select a parameter record. If you are running the job in the Administration application, enter a valid Parameter ID from the Custom Parameter table from the Financial application.

Schedule the Roll Batch process by specifying the Parameter ID as the Batch Parameter ID.

Custom Batch Parameters (RLPA)

Please keep in mind that when using the COA selection fields of Department, Program, Appropriation, and Appropriation Type that these elements may not exist on all event types being selected. If an event type is selected and used without one of these elements used as selection criteria on RLPA, the accounting line will not be selected.

Also, if the selection criteria are not enough to identify what is to be rolled, then the pre-selection mode is available to load the Roll/Lapse Pre Selection Summary and Detail tables for manual selection. Reports can be run to take records from the detail table and join with other information to produce a very detailed listing of potential records for selection.

Field	Description
Parameter ID	A required unique ID, identifying a set of parameters for system processing.
Closing BFY	A required budget fiscal year for accounting line selection. Please enter as CCYY.
Reorg (Reorganization)	An indication that the Budget FY will not change but COA changes indicated by the COAX ID batch parameter will be used. This is not allowed for the Accrual mode.
Mode	A required indication of how the Parameter ID will be used: *Report*

	Only, *Update*, and *Pre-Selection*. • *Report Only* is used by the Open Activity Roll Report only to produce a listing of matching records. • *Pre-Selection* is an optional mode used to populate the Roll/Lapse Pre-Selection Summary and Detail records for user review before rolling. This mode is also used for refreshing that pre-selection data. • *Update* is used to perform a roll whether from the Pre-Selection data approved for rolling or from the current transaction catalogs for all matching records.
Event Type Selection Criteria	A required selection parameter of one or more event types. Commas should separate multiple event types. There are limits on what event types can be rolled. Any listed must meet one of the following by reviewing the posting codes in posting pair A: • Expense Budget ID of 12 or 13 • Revenue Budget ID of 22, 23, or 25 • Both Budget IDs are blank. • If not the IDs mentioned above, the ID is listed in the Application Parameter of ALW_ROLL_BKTS, which is intended to allow rolling of billed vendor refunds that updated Accrued Expenditures (14) only. • Event types that update Disbursement Request are prohibited.
Target Event Type	An optional event type to be used in place of the existing event type on modification transactions. The use of this has to be where the postings are the same between the old and new event types. The intended use is to change an event type such as PR05 to a cloned event type that is only different in terms of ID and name. This allows BFY Staging to work differently to limit to decreases only. Furthermore, it allows reporting to locate reductions to this event type when budget availability was rolled to cover the encumbrance, in the event that availability needs to be reverted.
Transaction Codes	An optional selection parameter for one or more transaction codes. Separate multiple values with commas. Rolling is only supported for the following transaction type codes: ABS, RE, ARE, PO, RQ, SRQ, and TRVL.
Fund Selection Criteria	An optional selection parameter for one or more funds. Separate multiple values with commas.
Department Selection Criteria	An optional selection parameter for one or more departments. Separate multiple values with commas.
Appropriation Selection Criteria	An optional selection parameter for one or more appropriations. Separate multiple values with commas.
Appropriation Type Selection Criteria	An optional selection parameter for one or more appropriation types. Separate multiple values with commas.
Program Selection Criteria	An optional selection parameter for one or more programs. Separate multiple values with commas.
Modification Fiscal Year	An optional fiscal year placed on rolled accounting lines in order not to use the default year of the Record Date parameter (if used) or the

	Application Date. If either date is in the 'old' year, this parameter is used to ensure the 'new' year is used when rolling activity before year end. Please enter as CCYY. The backout posting line will use the Application Parameter record – BACKOUT_FY.
Modification Accounting Period	An optional accounting placed on rolled accounting lines in order not to use the default year of the Record Date parameter (if used) or the Application Date. If either date is in an 'old year' APD, this parameter ensures a 'new' year APD is used when rolling activity before year end. Please enter as CCYY. The backout posting line will use the Application Parameter record – BACKOUT_APD.
Adjustment Reason	A required output parameter when rolling receivables as such an event requires a Reason Code. It is not used when rolling other types of transactions.
Transaction Record Date	An optional date to be used on rolled transactions instead of letting the application date default.
Roll or Accrual Processing	A required type of processing to indicate if performing a *roll* (reorg or not) or an *accrual*. The latter is infrequent and used when it is known that goods or services have been received by year end, but no payment request has been created for reasons such as no invoice. *Accrual* works only with Open Activity Roll and not the larger version of that process that involves budget transactions. This mode decreases the encumbering accounting line to what has been closed, copies the line to create a new line, and sets that line to be the amount reduced from the existing line.
Accrual Event Type	A required output parameter when the Roll or Accrual Processing indication is *Accrual.* The event type should be a clone of AP01 without disbursement request updates and will likely use a different balance sheet posting code from Disbursements Payable (D001). Perhaps one for accrued payables.
Allow Fund Pre-Encumbrance Encumbrance Receivable Other Items Close Actions	When set to Yes, an individual fund may supply a different close action. Using this option can eliminate the need to use the Selected Funds field with long lists.
Pre-Encumbrance Encumbrance Receivable Other Items Roll Minimum	A minimum dollar amount which an open accounting line or transaction must be equal to or greater than for selection. Set to $0.00 if all matching open activity should be rolled. Common use is to roll 'big dollar' Lines leaving 'small dollar' lines for lapsing.
Allow Fund Pre-Encumbrance Encumbrance	When set to *Yes*, an individual fund may supply a different minimum.

Receivable Other Items Roll Minimum	
Pre-Encumbrance Encumbrance Receivable Other Items Roll Minimum Transaction/Line	The setting determines at which level the roll minimum is applied: Transaction or Accounting Line. This gives a larger picture of what is small or large.

Application Parameters (APPCTRL)

Field	Description
Backout Fiscal Year (BACKOUT_FY)	Backout fiscal year on the posting line(s) that remove open amounts from the Source BFY. Sites must change this each year. If left blank, if a Modification FY is used on RLPA that will be the FY used. If Modification FY is left blank and BACKOUT_FY is left blank, then all posting lines will use the default FY. This would only be correct accounting if rolling receivables into the new BFY before the end of the FY.
Backout APD (BACKOUT_APD)	Backout accounting period on the posting line(s) that remove open amounts from the Source BFY. Unlike BACKOUT_FY, this parameter is often set to 12 or 13 and left alone. If left blank, if a Modification APD is used on RLPA that will be the APD used. If Modification APD is left blank and BACKOUT_APD is left blank, then all posting lines will use the default APD. This would only be correct accounting if rolling receivables into the new BFY before the end of the FY.
Open Activity and Budget Roll Running (OABR_RUNNING)	OABR Running is a true/false parameter that must be set to True before running and False after running the Open Activity & Budget Roll chain job. **All VLS must be restarted after this value is changed (whether from false to true or true to false).**
Open Activity and Budget Roll BFY (OABR_SRC_BFY)	The Open Activity & Budget Roll chain job requires that the same value entered in the Source BFY field on the Parameters for Roll Process (RLPA) ID being used to be entered here as well. The value is used by the application when generating allotment lines. Whether allotments are used or not, this parameter has to be completed. Sites must change this each year.
Allowed Roll Budget Amounts (ALW_ROLL_BKTS)	This parameter is used as an additional control to the edits that the Parameters for Roll Process (RLPA) page currently performs for the Event Type Selection Criteria field. It is to expand what is allowed for that selection criteria and not restrict the field. The selection field currently allows event types with a posting code in Posting Pair A that has a Revenue or an

	Expense Budget ID or Revenue Budget Bucket ID that is: 22 (billed earned revenue), 23 (unbilled earned revenue), 25 (billed unearned/deferred revenue), 12 (pre-encumbrance), 13 (encumbrance), or no value in either ID field. If an event type entered has a different ID, the page will look to this parameter to see if that ID is allowed. The intended use of this parameter is to contain 14 (accrued expenses) so that only vendor refunds (event type AR30) can be rolled on receivables so the eventual collection will have a Fiscal Year equal to the Budget Fiscal Year. When these should not roll, the parameter should be left blank. When using this parameter, care should be taken in the setup on RLPA to ensure accrued expenditures that are payables are not rolled.

Major Output

- OABR Workload (OABR_WRKLD)
- Parameter file (OABRParms.txt)

Job Return Code

Return Code	Condition
Successful (1)	If preprocessing runs without any parameter errors. Pre selection ends as successful whether or not records were selected.
Warning (4)	Job does not end with this code.
Non-Fatal Error (8)	No records were selected for updated mode based on FUND and RLPA.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

- Transaction Department Code (DOC_DEPT_CD)
- Transactions Code (DOC_CD)
- Transaction ID (DOC_ID)
- Transaction Version Number (DOC_VERS_NO)
- Transaction Vendor Line Number (DOC_VEND_LN_NO)
- Transaction Commodity Line Number (DOC_COMM_LN_NO)
- Transaction Accounting Line Number (DOC_ACTG_LN_NO)

Selection Criteria

1. Selection criteria for obtaining Transaction Accounting lines (DOC_ACTG) to be rolled is –where the Accounting Line value below matches the RLPA value:
    a. BFY matches Closing BFY
    b. Event Type matches one of Event Type Selection Criteria
    c. Transaction Code matches one of the optional Transaction Codes
    d. Fund matches one of the optional Fund Selection Criteria
    e. Department (accounting line one and not Transaction Department) matches one of the optional Department Selection Criteria
    f. Appropriation matches on of the optional Appropriation Selection Criteria
    g. Appropriation Type matches on of the optional Appropriation Type Selection Criteria
    h. Program matches one of the optional Program Selection Criteria
    i. Open Amount is > $0.00
    j. Transaction Type is not ‘JV’
2. If the respective Fund Override flag is checked, get respective Closing Action and Minimum Roll Amount by lookup to FUND using Closing BFY as FY. Please see the Chain Job overview for more information on matching action to type of open activity.
    a. If that FUND action is *Roll*, then select.
    b. If that Fund action is other than *Roll*, do not select.
3. When using the RLPSD table for selection, if the Approved flag is checked, the Action is *Roll* and the Selection Date is blank, then select.

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If no records are selected, review the selection criteria on the RLPA ID used. Also, review the FUND page to ensure that the appropriate Action field has a value of *Roll*.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	The pre processor did not encounter any severe parameter errors.	If no records were selected, check the job log for messages as certain parameter errors will not cause the job to fail, but will	N/A

	If in pre select mode, then successful whether or not records loaded.	be logged.	
Warning (4)	Job does not end with this return code.	N/A	N/A
Non Fatal Error (8)	No records were selected for rolling	Check RLPA ID, the Use Pre selection batch parameters.	Run in report mode until parameters are correct as it is an easier job to run. Then change RPPA to Update or Pre Selection and run the chain.
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following conditions. 1) Encounters any runtime exceptions and 2) Invalid parameter found 3) text files used as templates could not be found If the job fails because of these, investigate the problem. Start a new chain job once the problem is fixed.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Reschedule a new job after the reason for termination is resolved.	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. Reschedule a new job after the reason for termination is resolved.	N/A

Open Activity and Budget Roll Chain: Create Round 1 Job

Job Name	Create Round 1

Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	Yes – Please see the Problem Resolution section for details
Reports generated	Yes – Roll Transaction Report to list transactions being rolled in round 1

Overview

This job will read the OABRParams txt file and the OABR Workload table to create units of work on the OABR Facilitator table and then spawn one or more Open Actv & Budget Roll jobs that will create the necessary modification drafts.

The Create Round 1 job inserts one or more rows into the OARB Facilitator to define a unit of work from one sequence number to another (numbers defined on the OABR Workload table by the previous job step) based on the JOB_BLOCK_SIZE parameter specified in the first job step. Each row inserted is then assigned to a spawned Open Actv & Budget Update job through the population of the Agent ID with the Job ID of the spawned Open Activity Roll Update job). All facilitator records are tied by a Run Number (RUN_NO), which is the Chain Job ID.

The Create Round 1 job will keep running until all spawned Open Actv & Budget Roll jobs are spawned and completed. When those are done, this job step will end.

Process Steps	Messages
1. Parameter Validation	• Batch input parameters are listed. If any is invalid, an error message is issued stating such.
2. Spawning of Open Actv & Budget Update jobs	• Job with Id:### sequence number:# completed successfully (listed for each spawned job that completed successfully) • Batch job ### failed (listed for each spawned job that failed.
3. Report generation	• Reports output folder mapped (followed by pdf and html report information) • Rendering report started • Rendering report completed

Major Input

- OABR Workload (OABR_WRKLD)
- Parameter file (OABRParams.txt)

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Create Round 1 Job (**** Refer to Note:** Assumptions for	$$AMSROOT$$/Logs

	SWBP on page no. 6)	
AMSPARM	Parameter Location at Create Round 1 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Parms
PARM_FILE	Parameter File (.txt) For system use and should not be changed.	OABRParams.txt
ROUND_NO	Round Number For system use and should not be changed.	1
RUN_TYPE	Run Type For system use and should not be changed.	3 (create)

Major Output

- Open Actv & Budget Update job(s) created
- Roll Transaction Report

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and all spawned jobs ended successfully
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 • One or more spawned jobs failed (ID of that job is specified in job log) • Run time exceptions for unexpected situations When this job ends as failed, subsequent jobs in the chain will become inactive
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process was discontinued for some reason (for example, job was killed) the user has an option to restart the job and the process will continue from the point it left off. This restart has a condition that no roll process must be scheduled between the time frame the job was discontinued and is being restarted. The Restart option is provided for jobs – "Create Round n" and "Submit Round n" where n is 1, or 2 as well as the spawned jobs for those different Create and Submit rounds.

In the case of a system failure during any Create Round, all spawned submit jobs (Open Actv & Budget Update + <chain job id>) will return a Return Code of *Failed* or *System Failure*. The Create Round or will also return a Return Code of *Failed*, and further jobs down the chain are Inactive. In this case, the user should first restart each spawned job (Open Actv & Budget Update) so that the individual job can complete work on the set of transactions it was assigned. Then after each spawned job completes successfully, the user should restart the Create Round job. If there is further work to be done in that Create Round, the chain will spawn additional Open Actv & Budget Update jobs to complete that round's work. After all the applicable work is completed, the chain will automatically move on to the next job in the chain – Submit Round.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and all spawned jobs completed 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 conditions. 1) Encounters any runtime exceptions and 2) Invalid parameter found 3) text files used as templates could not be found 4) Spawned job failed If the job fails because of	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

		these, investigate the problem. This and any spawned job that failed can be restarted after the problem is resolved.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity and Budget Roll Chain: Submit Round 1 Job

Job Name	Submit Round 1
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	Yes – Please see the Problem Resolution section for details
Reports generated	No

Overview

This job will read the OABRParams txt file and OABR Workload table to create units of work on the OABR Facilitator table and then spawn one or more System Maintenance Utility (SMU) jobs that will attempt to submit the modification drafts previously loaded.

The Submit Round 1 job inserts one or more rows into the OARB Facilitator with a txt file created to log a block of transactions for the spawned SMU job to submit. One final SMU submit is also spawned to catch any transactions that failed on the first submit as a cleanup measure to catch any transactions that failed because another was making an update at the same exact moment.

The Submit Round 1 job will keep running until all spawned SMU jobs are spawned and completed. When those are done, this job step will end.

Process Steps	Messages
1. Parameter Validation	• Batch input parameters are listed. If any is invalid an error message is issued stating such
2. Spawning of SMU jobs	• Job with Id:### sequence number:# completed successfully (listed for each spawned job that completed successfully) • Batch job ### failed (listed for each spawned job that failed

Major Input

- OABRParams.txt file
- OABR Workload (OABR_WRKLD)

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Create Round 1 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Logs
AMSPARM	Parameter Location for Roll Update Preprocess Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Parms
PARM_FILE	Parameter File (.txt) For system use and should not be changed.	OABRParams.txt
ROUND_NO	Round Number For system use and should not be changed.	1
RUN_TYPE	Run Type For system use and should not be changed.	2 (submit)
SUBMIT_FILE_NM	Report File for Round 1 (.txt) For system use and should not be changed.	OABRReportRndOne.txt

Major Output

- Spawned SMU job(s)
- Accepted modification transactions (all common and specific transaction components)
- Accounting Journal (JACTG/JRNL_ACTG)

- Report file (OARBRReportRndOne.txt)

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and all spawned jobs ended successfully
Warning (4)	Job does not end with this return code, but spawned SMU jobs may.
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 • Spawned SMU job fails • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process was discontinued for some reason (for example, job was killed) the user has an option to restart the job and the process will continue from the point it left off. This restart has a condition that no roll process must be scheduled between the time frame the job was discontinued and is being restarted. The Restart option is provided for jobs – “Create Round n” and “Submit Round n” where n is 1, or 2 as well as the spawned jobs for those different Create and Submit rounds.

In the case of a system failure during any Submit Round, all spawned submit jobs (System Maintenance Utility + <chain job id>) will return a Return Code of *Failed* or *System Failure*. The Submit Round or will also return a Return Code of *Failed*, and further jobs down the chain are Inactive. In this case, the user should first restart each spawned job (System Maintenance Utility) so that the individual job can complete work on the set of transactions it was assigned. Then

after each spawned job completes successfully, the user should restart the Submit Round job. If there is further work to be done in that Submit Round, the chain will spawn additional SMU jobs to complete that round's work. After all the applicable work is completed, the chain will automatically move on to the next job in the chain - Exception Report Round.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and all spawned jobs completed successfully.	N/A	N/A
Warning (4)	Job does not end with this return code, but spawned SMU jobs may because of optimistic locking errors.	Restart the SMU job	If the Load Round 1 job will not restart after the SMU job completes, then start a new chain at the Round 1 Exception Report. Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
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 conditions. 1) Encounters any runtime exceptions 2) Invalid parameter found 3) text files used as templates could not be found 4) Spawned job failed If the job fails because of these, investigate the problem. This and any spawned	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

		job that failed can be restarted after the problem is resolved.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity and Budget Roll Chain: Exception Report Round 1 Job

Job Name	Except Rpt Rnd 1
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Rolled Transaction Exception Report

Overview

This job will read the Report file (RollReportRndOne.txt) created from the submit step and produce a report listing those modifications that failed to submit along with a limited amount of information on the errors issued for that transaction. If no transactions fail, a report is still produced to state "No Transaction Errors". In the case of no errors, the job ends as *Successful*. If any transaction fails to submit to final, then the job step ends as *Non-Fatal* and the chain stops. Please see the Problem Resolution section for details about what to do in this case.

Process Steps	Messages
1. Parameter validation	• Batch input parameters are listed. If any is invalid, an error message is issued stating such.
2. Report generation	• Reports output folder mapped (followed by pdf and html report information) • No transaction errors or

	• Error records processed ## • Rendering report started • Rendering report completed

Major Input

- RollReportRndOne.txt

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Except Rpt Round 1 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Logs
AMSPARM	Parameter Location at Except Rpt Rnd 1 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Parms
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) For system use and should not be changed.	RollParams.txt
SUBMIT_FILE_NM	Report File for Round 1 (.txt) For system use and should not be changed.	OABRReportRndOne.txt

Major Output

- Exception Report

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and report was generated successfully.
Warning (4)	Job does not end with this return code.
Non-Fatal Error (8)	At least one transaction failed to submit.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent

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

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If an Exception Report lists any failed transactions due to wrong data setup that caused the modifications to fail, then the following steps can be executed to resubmit the failed rolling transactions.

1. Correct any data setup problems that caused transactions to fail. Often it is just a very limited number of control table settings.
2. If the number of failed transactions is low, they can be manually submitted. If the number is beyond manual intervention, then an independent SMU job can be run to submit the modifications.
3. After either method, a new chain can be submitted starting with the Transaction Exception Report to verify all were submitted successfully. It is critical to ensure that no parameters are changed on the Parameter table and no roll process has run in update mode between the actual run and restart run, else any .txt files marked for system use (see parameter list above) are overwritten and the restart run will not work correctly.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and all spawned jobs completed successfully.	N/A	N/A
Warning (4)	Job does not end with this return code.	N/A	N/A
Non-Fatal Error (8)	One or more modifications failed to submit.	Need to complete submit process after the reason for the failure is resolved.	See problem resolution above

Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following conditions. 1) Encounters any runtime exceptions 2) Invalid parameter found 3) text files used as templates could not be found 4) Exception report file could not be found. If the job fails because of these, investigate the problem. Need to schedule a new chain starting at this point after reason for the failure is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Need to schedule a new chain starting at this point after reason for the termination is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. Need to schedule a new chain starting at this point after reason for the failure is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity and Budget Roll Chain: Create Round 2 Job

Job Name	Create Round 2
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	Yes – Please see the Problem Resolution section for details
Reports generated	Yes – Roll Transaction Report to list transactions being rolled in round 2

Overview

This job will read the RollParams.txt file and the OABR Workload table to create units of work on the OABR Facilitator table and then spawn one or more Open Actv & Budget Roll jobs that will create the necessary modification drafts.

The Create Round 2 job inserts one or more rows into the OARB Facilitator to define a unit of work from one sequence number to another (numbers defined on the OABR Workload table by the previous job step) based on the JOB_BLOCK_SIZE parameter specified in the first job step. Each row inserted is then assigned to a spawned Open Actv & Budget Update job through the population of the Agent ID with the Job ID of the spawned Open Activity Roll Update job). All facilitator records are tied by a Run Number (RUN_NO), which is the Chain Job ID.

The Create Round 2 job will keep running until all spawned Open Actv & Budget Roll jobs are spawned and completed. When those are done, this job step will end.

Process Steps	Messages
1. Parameter Validation	• Batch input parameters are listed. If any is invalid, an error message is issued stating such.
2. Spawning of Open Actv & Budget Update jobs	• Job with Id:### sequence number:# completed successfully (listed for each spawned job that completed successfully) • Batch job ### failed (listed for each spawned job that failed.
3. Report generation	• Reports output folder mapped (followed by pdf and html report information) • Rendering report started • Rendering report completed

Major Input

- OABR Workload (OABR_WRKLD)
- Parameter file (OABRParams.txt)

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Create Round 2 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Logs
AMSPARM	Parameter Location at Create Round 2 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Parms
PARM_FILE	Parameter File (.txt)	OABRParams.txt

	For system use and should not be changed.	
ROUND_NO	Round Number For system use and should not be changed.	2
RUN_TYPE	Run Type For system use and should not be changed.	3 (create)

Major Output

- Open Actv & Budget Update job(s) created
- Roll Transaction Report

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and all spawned jobs ended successfully.
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 • One or more spawned jobs failed (ID of that job is specified in job log) • Run time exceptions for unexpected situations When this job ends as failed, subsequent jobs in the chain will become inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process was discontinued for some reason (for example, job was killed) the user has an option to restart the job and the process will continue from the point it left off. This restart has a condition that no roll process must be scheduled between the time frame the job was discontinued and is being restarted. The Restart option is provided for jobs – "Create Round n" and "Submit Round n" where n is 1, or 2 as well as the spawned jobs for those different Create and Submit rounds.

In the case of a system failure during any Create Round, all spawned submit jobs (Open Actv & Budget Update + <chain job id>) will return a Return Code of *Failed* or *System Failure*. The Create Round will also return a Return Code of *Failed*, and further jobs down the chain are Inactive. In this case, the user should first restart each spawned job (Open Actv & Budget Update) so that the individual job can complete work on the set of transactions it was assigned. Then after each spawned job completes successfully, the user should restart the Create Round job. If there is further work to be done in that Create Round, the chain will spawn additional Open Actv & Budget Update jobs to complete that round's work. After all the applicable work is completed, the chain will automatically move on to the next job in the chain – Submit Round.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and all spawned jobs completed 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 conditions. 1) Encounters any runtime exceptions and 2) Invalid parameter found 3) text files used as templates could not be found 4) Spawned job failed If the job fails because of these, investigate the problem. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity and Budget Roll Chain: Submit Round 2 Job

Job Name	Submit Round 2
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	No

Overview

This job will read the OABRParams txt file and OABR Workload table to create units of work on the OABR Facilitator table and then spawn one or more System Maintenance Utility (SMU) jobs that will attempt to submit the modification drafts previously loaded.

The Submit Round 2 job inserts one or more rows into the OARB Facilitator with a txt file created to log a block of transactions for the spawned SMU job to submit. One final SMU submit is also spawned to catch any transactions that failed on the first submit as a cleanup measure to catch any transactions that failed because another was making an update at the same exact moment.

The Submit Round 2 job will keep running until all spawned SMU jobs are spawned and completed. When those are done, this job step will end.

Process Steps	Messages
1. Parameter Validation	• Batch input parameters are listed. If any is invalid an error message is issued stating such
2. Spawning of SMU jobs	• Job with Id:### sequence number:# completed successfully (listed for each spawned job that completed successfully) • Batch job ### failed (listed for each spawned job that failed

Major Input

- OABRParams.txt file
- OABR Workload (OABR_WRKLD)

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Create Round 2 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Logs
AMSPARM	Parameter Location for Roll Update Preprocess Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Parms
PARM_FILE	Parameter File (.txt) For system use and should not be changed.	OABRParams.txt
ROUND_NO	Round Number For system use and should not be changed.	2
RUN_TYPE	Run Type For system use and should not be changed.	2 (submit)
SUBMIT_FILE_NM	Report File for Round 2 (.txt) For system use and should not be changed.	OABRReportRndTwo.txt

Major Output

- Spawned SMU job(s)
- Accepted modification transactions (all common and specific transaction components)
- Accounting Journal (JACTG/JRNL_ACTG)
- Report file (OARBRReportRndOne.txt)

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and all spawned jobs ended successfully.

Warning (4)	Job does not end with this return code, but spawned SMU jobs may.
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 • Spawned SMU job fails • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process was discontinued for some reason (for example, job was killed) the user has an option to restart the job and the process will continue from the point it left off. This restart has a condition that no roll process must be scheduled between the time frame the job was discontinued and is being restarted. The Restart option is provided for jobs – “Create Round n” and “Submit Round n” where n is 1, or 2 as well as the spawned jobs for those different Create and Submit rounds.

In the case of a system failure during any Submit Round, all spawned submit jobs (System Maintenance Utility + <chain job id>) will return a Return Code of *Failed* or *System Failure*. The Submit Round will also return a Return Code of *Failed*, and further jobs down the chain are Inactive. In this case, the user should first restart each spawned job (System Maintenance Utility) so that the individual job can complete work on the set of transactions it was assigned. Then after each spawned job completes successfully, the user should restart the Submit Round job. If there is further work to be done in that Submit Round, the chain will spawn additional SMU jobs to complete that round’s work. After all the applicable work is completed, the chain will automatically move on to the next job in the chain - Exception Report Round.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and all spawned jobs completed successfully.	N/A	N/A
Warning (4)	Job does not end with this return code, but spawned SMU jobs may because of optimistic locking errors.	Restart the SMU job	If the Load Round 1 job will not restart after the SMU job completes, then start a new chain at the Round 1 Exception Report. Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
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 conditions. 1) Encounters any runtime exceptions 2) Invalid parameter found 3) text files used as templates could not be found 4) Spawned job failed If the job fails because of these, investigate the problem. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
Terminated	Job is terminated	Reason for the	Ensure no

(16)	manually by the user.	termination needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity and Budget Roll Chain: Exception Report Round 2 Job

Job Name	Except Rpt Rnd 2
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Rolled Transaction Exception Report

Overview

This job will read the Report file (OARBRReportRndTwo.txt) created from the submit step and produce a report listing those modifications that failed to submit along with a limited amount of information on the errors issued for that transaction. If no transactions fail, a report is still produced to state "No Transaction Errors". In the case of no errors, the job ends as *Successful*. If any transaction fails to submit to final, then the job step ends as *Non-Fatal* and the chain stops. Please see the Problem Resolution section for details about what to do in this case.

Process Steps	Messages
• Parameter validation	• Batch input parameters are listed. If any is invalid, an error message is issued stating such.
• Create Report	• Reports output folder mapped (followed by pdf and html report information) • No transaction errors or • Error records processed ## • Rendering report started • Rendering report completed

Major Input

- OARBRReportRndTwo.txt

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Except Rpt Round 2 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Logs
AMSPARM	Parameter Location at Except Rpt Rnd 2 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Parms
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) For system use and should not be changed.	OABRParams.txt
SUBMIT_FILE_NM	Report File for Round 2 (.txt) For system use and should not be changed.	OABRReportRndTwo.txt

Major Output

- Exception Report

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and report was generated successfully.
Warning (4)	Job does not end with this return code.
Non-Fatal Error (8)	At least one transaction failed to submit.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated

	because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If an Exception Report lists any failed transactions due to wrong data setup that caused the modifications to fail, then following steps can be executed to resubmit the failed rolling transactions.

1. Correct any data setup problems that caused transactions to fail. Often it is just a very limited number of control table settings.
2. If the number of failed transactions is low, they can be manually submitted. If the number is beyond manual intervention, then an independent SMU job can be run to submit the modifications.
3. After either method, a new chain can be submitted starting with the Transaction Exception Report to verify all were submitted successfully. It is critical to ensure that no parameters are changed on the Parameter table and no roll process has run in update mode between the actual run and restart run, else any .txt files marked for system use (see parameter list above) are overwritten and the restart run will not work correctly.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and all spawned jobs completed successfully	N/A	N/A
Warning (4)	Job does not end with this return code.	N/A	N/A
Non-Fatal Error (8)	One or more modifications failed to submit.	Need to complete submit process after the reason for the failure is resolved.	See problem resolution above
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following conditions. 1) Encounters any runtime exceptions	Ensure no OABR chain has started with the Pre Process job step (#1) in

		2) Invalid parameter found 3) text files used as templates could not be found 4) Exception report file could not be found. If the job fails because of these, investigate the problem. Need to schedule a new chain starting at this point after reason for the failure is resolved.	the interim. If so, this chain is effectively over.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Need to schedule a new chain starting at this point after reason for the termination is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. Need to schedule a new chain starting at this point after reason for the failure is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity and Budget Roll Chain: Budget Roll Job

Job Name	Budget Roll
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	Yes – for parameter errors only
Reports generated	No

Overview

- This job step is the first of four to facilitate the 'budget' part of Open Activity & Budget Roll. The Application Parameter (APPCTRL) of OABR_RUNNING set to *true* has resulted in the population of data on a summary table to record and accumulate all budget updates made

from the processing of accounting transaction modifications. The Budget Activity Detail (RL_BUD_DTL_AM) and Summarized Budget Activity (RL_BUD_SUMM) tables now contain all information necessary to generate budget transactions to reduce budgets in the older BFY and move the funds to the new BFY to cover the accounting updates made by the processing of the rolled transactions.

- The Budget Activity Summary table contains an online page for viewing with the page code of SBA. The page also allows for roll feature whereby a user can designate which budget lines should roll and which should not. This feature does require that two different OABR chain jobs are submitted: the first to run through the Exception Report Round 2 job and the second to pick up with the Budget Roll job step and continue to the end. In between the chains, one or more users would set the Selected field as appropriate for different budget lines.

Because the Auto Generate was set for all Budget Structures in the target fiscal year in Job 1 – Roll Update Preprocess, the budgets are initialized with zero amounts in the target fiscal year as necessary during transaction submission. This job step starts by summarizing amounts from the RL_BUD_DTL_AM table and creating an XML file. Finally, it creates a file that lists all transactions created for use by Job 11 – Budget Roll Reports to generate reports.

The Parameters for Budget Roll (PBRP) table supplies information for the selection of Summarized Budget Activity records as well as values used to populate the XML. In the case that more than one budget is to be rolled with open accounting activity, multiple chains are required. The first will contain jobs through Job 11 - Budget Roll Reports. Then as many chains as there are other budget structures will be run containing only Job 8 – Budget Roll through Job 11 – Budget Roll Reports, with the last one having the Post Roll Update job.

Process Steps	Messages
1. Collect Common Chain Parameters	Acquire the parameters for Rolling Budgets.
2. Parameter Validation	Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log.
3. After Parameter Validation	Acquire the PBRP for Rolling Budgets.
4. Verify PBRP is valid	Ensure the PBRP is valid for Rolling Budgets.
5. Before Parameter Validation	Perform additional validations on parameters for Rolling Budgets.
6. Parameter Validation	Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log.
7. Initialize values if restarted	Initialize critical values for Rolling Budgets.
8. Before Record Selection	Begin logic to build Budgets to be rolled.

9. Selection of Records	Transactions created: #
10. XML file creation	No messages.
11. Text file creation	No messages.
12. End of job	End logic to build Budgets to be rolled.

Major Input

- Budget Activity Detail (RL_BUD_DTL_AM) and Budget Activity Summary (SBA / RL_BUD_SUMM) tables
- OABRParams.txt
- Parameters for Budget Roll Process (BPBP / BUD_ROLL_PARM)

Peripheral Input

- Application Parameters (APPCTRL / IN_APP_CTRL)
- Auto Transaction Numbering (ADNT / AUTO_DOC_NO)
- COA Crosswalk (COAX / R_COA_CROSSWALK)
- Budget Allotment Options (GN_ALOT_OPT)

Batch Parameters

For the entire chain process, a user has to enter parameters for Job 1 – Roll Update Preprocess and then on Job 8 - Budget Roll only. The process propagates required parameters down the chained jobs. Some of them are non-overrideable parameters, which will be provided with the Job Setup and need not be specified for each run. Many of these are just meant for system use and should not be changed. Those are so noted in the Description column.

Parameter Name	Description	Default Value
AMSEXPORT	Export Location at Budget Roll Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Export Import
AMSPARM	Parameter Location at Budget Roll Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Parms
BUD_ROLL_PARM_ID	Budget Roll Parameter ID. The **Setup Custom Parameters** link leads to the Parameter page.	(bank)
CHAIN_PARM_FILE	Common Chain Parameters File (.txt)	OABRParams.txt

	This parameter value should be the same as that of the CHAIN_PARM_FILE parameter of Job 1. For system use and should not be changed.	
CHK_PT_SIZE	Checkpoint Block Size (Number of transactions processed after which the checkpoint is to be updated)	100
COMMIT_SIZE	Required commit block size for the transaction load and submit job steps. This figure determines the number of transactions to be committed at a time for all jobs in the chain that perform transaction load and submit action. It can be used for performance enhancements depending upon the server memory configuration.	1
EXP_FILE_NM	Transaction XML file (.xml) for Budget Transactions For system use and should not be changed.	OABRDocs.xml
LOAD_FILE_NM	Load Parameter File (.txt) for Budget Transactions For system use and should not be changed.	OABRLoad.txt
ROLLUP_SUBMIT_FILE_NM	Report file (.txt) for Budget Transactions For system use and should not be changed.	OABRRollupSubmit.txt

Custom Batch Parameters (PBRP)

There are several fields on PBRP that are not allowed or not used with the *Work with Open Activity* mode. Those have been omitted from the table below.

Field	Description	Edits
Parameter ID	Unique Parameter ID	• Error issued if not value is entered
Budget Structure	The budget structure ID that will have budget authority rolled forward with open accounting activity.	• Must be a valid budget structure ID. • Should be one that has a BFY in its definition.
Target BFY	For *Work with Open Activity Mode,* the field should be set to the Closing BFY value for the RLPA parameter ID being used.	• Error issued if the year does not exist on the FY page • Error issued if the field is

		blank • Error issued if multiple years are entered • In each case, the job does not run successfully and an error is issued in the job log
Mode	Can only be *Work with Open Activity Mode* when used with OABR.	• Can only be *Work with Open Activity Mode* when used with OABR.
Source Event Type	Required event type to record the decrease in budget in the Source BFY. Most commonly set to BG05 - Revert An Expense Budget. Must be an event type that uses a posting code that updates a bucket that is subtracted as updates will be made with the Increase setting on the budget transaction.	• Error issued if the event type does not exist on R_EVNT_TYP. • Error issued if the field is null. • In each case, the job does not run successfully and an error is issued in the log.
Target Event Type	Required event type to record the increase in budget in the Target BFY. Most commonly set to BG04 – Carry Forward An Expense Budget.	• Error issued if the event type does not exist on R_EVNT_TYP. • Error issued if the field is null. • In each case, the job does not run successfully and an error is issued in the log.
Transaction Code	Required transaction code of the Budget Structure specified earlier to be used for the budget roll.	• Error issued if the transaction code does not exist on R_GEN_DOC_CTRL.
Transaction Department (Transaction Dept)	Required transaction department code to be used for the budget roll.	• Error issued if the department code does not exist on R_DEPT.
Transaction Unit (Transaction Unit)	Optional transaction unit code to be used for the budget roll. Must be entered if Transaction Control required Unit or it is necessary for security purposes.	• Error issued if the unit code does not exist on R_UNIT.
Transaction Prefix	Optional transaction prefix to be used for the budget roll.	• Error issued if the transaction code, department, and prefix do not exist on AUTO_DOC_NO, using the Target BFY as the FY.

Transaction Break	Required choice of 4 values that will control the size of budget transactions created as well as the consistency of information within each. Choose from *Budget Line to 1 Transaction, All Budget Lines for a Fund to 1 Transaction, All Budget Lines for a Department to 1 Transaction,* and *All Budget Lines for an Appropriation to 1 Transaction.*	• N/A
Include Allotments	Flag to indicate whether budget authority at the allotment line should also be rolled as part of OABR. Feature only works if the lowest required budget level is allotted.	• Cannot be checked if the Budget Structure used does not contain allotments.
Transaction Record Date	Optional date to be placed on the header of budget transactions. Used when the application date should not default or the roll may span the running of the Begin Day job.	• Must be a valid date.
Source Fiscal Year	Optional fiscal year (FY) to be placed on all budget lines for the Source BFY Please enter as CCYY. Leave blank to default from Transaction Record Date (see parameter below) if used or the Application Date when the roll is performed.	• Error issued if the FY does not exist on the Fiscal Year (R_FY) table. • Error issued if multiple periods are entered
Target Fiscal Year	Optional fiscal year (FY) to be placed on all budget lines for the Target BFY Please enter as CCYY. Leave blank to default from Transaction Record Date (see parameter below) if used or the Application Date when the roll is performed.	• Error issued if the FY does not exist on the Fiscal Year (R_FY) table. • Error issued if multiple periods are entered
Source Accounting Period	Optional accounting period (APD) to be placed on all budget lines for the Source BFY Leave blank to default from Transaction Record Date (see parameter below) if used or the Application Date when the roll is performed.	• Error issued if the APD does not exist on the Accounting Period (R_APD) table. • Error issued if multiple periods are entered.
Target Accounting Period	Optional accounting period (APD) to be placed on all budget lines for the Target BFY Leave blank to default from	• Error issued if the APD does not exist on the Accounting Period (R_APD) table.

	Transaction Record Date (see parameter below) if used or the Application Date when the roll is performed.	• Error issued if multiple periods are entered.

Application Parameters (APPCTRL)

Field	Description
OABR_SRC_BFY	The Open Activity & Budget Roll chain job requires that the same value entered in the Source BFY field on the Parameters for Roll Process (RLPA) ID being used to be entered here as well. The value is used by the application when generating allotment lines. Whether allotments are used or not, this parameter has to be completed. Sites must change this each year.

Major Output

- OABRDocs.xml file
- OABR Facilitator
- OARBLoad.txt
- OABRReport.txt.

Job Return Code

Return Code	Condition
Successful (1)	Parameter validation is successful and XML is generated.
Warning (4)	Job does not end with this code
Non-Fatal Error (8)	Job does not end with this code
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the

	chain will be set to inactive.

Sort Criteria

See Transaction Break in Selection Criteria

Selection Criteria

Only those records marked as approved (SEL_FL = 1) are selected that are for the budget structure of the PBRP ID.

The Transaction Break for the Parameter Id record determines if budget transactions will be created grouping all lines for a Fund, Appropriation, or Department. The other choice is one budget line per budget transaction. Sorting will be done prior to transaction creation based on this break value.

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. Restarting has a condition that no Budget Roll chain process should be scheduled between the time frame the job was discontinued and is being restarted and the parameters provided to the chain process should not be changed.

This batch program produces new Advantage transactions, which are subject to the line limit functionality constraints. Sites should ensure that they run this job with parameters set to ensure that the created transactions are within the line limit controls. The Transaction Component Requirement (DCREQ) entry in the Administration application that would be of issue would be one for BG_DOC_LN. The Transaction Break parameter on PBRP should be used to keep below any limit on the budget transaction line component.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Parameter validation is successful and XML is generated.	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 conditions. 1) Encounters any runtime exceptions and 2) Invalid parameter found 3) text files used could not be found If the job fails because of these, investigate the	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

		problem. Start a new chain job once the problem is fixed.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Start a new chain job once the problem is fixed.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. Start a new chain job once the problem is fixed.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity and Budget Roll Chain: Load Budget Transactions Job

Job Name	Load Budget Transactions
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This job step is a System Maintenance Utility load of the XML file of budget transactions created in the previous job step.

Process Steps	Messages
1. Parameter Validation	• Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log.
2. XML load	• Job with Id:## sequence number:# completed successfully • Job with Id:## sequence number:# completed successfully

Major Input

- OABRDocs.xml

- OARBLoad.txt

Batch Parameters

For the entire chain process, a user has to enter parameters for Job 1 – Roll Update Preprocess and then on Job 8 - Budget Roll only. The process propagates required parameters down the chained jobs. Some of them are non-overrideable parameters, which will be provided with the Job Setup and need not be specified for each run. Many of these are just meant for system use and should not be changed. Those are so noted in the Description column.

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Load Budget Transactions Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Logs
AMSPARM	Parameter Location at Load Budget Transactions Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Parms
PARM_FILE	Parameters File (.txt) This parameter value should be the same as that of the CHAIN_PARM_FILE parameter of Job 1. For system use and should not be changed.	OABRParams.txt
ROUND_NO	Round Number For system use and should not be changed.	5
RUN_TYPE	Run Type For system use and should not be changed.	1 (create)

Major Output

- Budget Transactions loaded with lines at the lowest required budget level

Job Return Code

Return Code	Condition
Successful (1)	Parameter validation is successful XML is loaded.
Warning (4)	• Job does not end with this code
Non-Fatal Error (8)	Job does not end with this code
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected

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

Sort Criteria

None

Selection Criteria

None

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. Restarting has a condition that no Budget Roll chain process should be scheduled between the time frame the job was discontinued and is being restarted and the parameters provided to the chain process should not be changed.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Parameter validation is successful and XML is loaded 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 conditions. 1) Encounters any runtime exceptions and 2) Invalid parameter found 3) text files used could	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively

		not be found If the job fails because of these, investigate the problem. Start a new chain job at this point once the problem is fixed.	over.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Start a new chain job at this point once the problem is fixed.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. Start a new chain job at this point once the problem is fixed.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity and Budget Roll Chain: Rollup & Submit Budget Transactions Job

Job Name	Rollup & Submit Budget
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This job step is a System Maintenance Utility that performs two actions on the transactions loaded from the previous step. The first is a smart budget rollup action to take the budget lines loaded at the lowest required budget level and generate 'parent' budget lines at higher levels. The second action is a submit action.

Process Steps	Messages
3. Parameter Validation	• Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log.

4. Budget Rollup & Submit	• Job with Id:## sequence number:# completed successfully • Job with Id:## sequence number:# failed

Major Input

- Budget Transaction Lines (BG_DOC_LN)
- OARBLoad.txt

Batch Parameters

For the entire chain process, a user has to enter parameters for Job 1 – Roll Update Preprocess and then on Job 8 - Budget Roll only. The process propagates required parameters down the chained jobs. Some of them are non-overrideable parameters, which will be provided with the Job Setup and need not be specified for each run. Many of these are just meant for system use and should not be changed. Those are so noted in the Description column.

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Rollup and Submit Budget Transactions Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Logs
AMSPARM	Parameter Location at Rollup and Submit Budget Transactions Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Parms
PARM_FILE	Parameters File (.txt) This parameter value should be the same as that of the CHAIN_PARM_FILE parameter of Job 1. For system use and should not be changed.	OABRParams.txt
ROUND_NO	Round Number For system use and should not be changed.	5
RUN_TYPE	Run Type For system use and should not be changed.	2
SUBMIT_FILE_NM	Report file (.txt) for Budget Transactions For system use and should not be changed.	OABRRollupSubmit.txt

Major Output

- Budget Transactions submitted

- Budget Journal (JBUD / JRNL_BUD)
- Budget Structure Level Inquiry (BQ*LV* / BUD_STRU_*_LVL_*)

Job Return Code

Return Code	Condition
Successful (1)	Parameter validation is successful and 2 actions completed.
Warning (4)	• Job does not end with this code
Non-Fatal Error (8)	Job does not end with this code
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

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. Restarting has a condition that no Budget Roll chain process should be scheduled between the time frame the job was discontinued and is being restarted and the parameters provided to the chain process should not be changed.

Roll reports job ends up in non-fatal error if the preceding Rollup and Submit transactions job had transaction submission errors, thus deactivating other jobs down the chain. If these errors were due to some data setup, fix the data setup errors and reschedule the chain enabling the preceding transaction submit job and Roll Reports job and disabling the prior jobs that ran successfully. This reschedule has a condition that no Budget Roll chain process should be

scheduled between the time frame the job was discontinued and is being scheduled again and the parameters provided to the chain process should not be changed.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Parameter validation is successful and job completes 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 conditions. 1) Encounters any runtime exceptions and 2) Invalid parameter found 3) text files could not be found If the job fails because of these, investigate the problem. Start a new chain job starting at this point once the problem is fixed.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Start a new chain job starting at this point once the problem is fixed.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. Start a new chain job starting at this point once the problem is fixed.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity and Budget Roll Chain: Budget Roll Reports Job

Job Name	Budget Roll Rpts
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes - Budget Roll Crosswalk of Successful Transactions and Budget Roll Listing of Unsuccessful Transactions

Overview

This job in OABR chain generates the reports for the Budget Roll Crosswalk of Successful Transactions Report and Budget Roll Listing of Unsuccessful Transactions Report. This job is not expected to be restarted, as the next job in the chain does not require successful completion in order to be run.

This job will read the Report file (OABRRollupSubmit.txt) created from the submit step and produce a report listing those budget transactions that failed to submit along with a limited amount of information on the errors issued for that transaction. If no transactions fail, a report is still produced to state "No Transaction Errors". In the case of no errors, the job ends as *Successful*. If any transaction fails to submit to final, then the job step ends as *Non-Fatal* and the chain stops. Please see the Problem Resolution section for details about what to do in this case.

Process Steps	Messages
1. Parameter validation	• Batch input parameters are listed. If any is invalid, an error message is issued stating such.
2. Report generation	• Reports output folder mapped (followed by pdf and html report information) for both reports • Successful transactions processed: # • Unsuccessful transactions processed: #

Major Input

- OABRRollupSubmit.txt

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSPARM	Parameter Location at Budget Roll Rpts Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	• $$AMSROOT$$/Parms
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) For system use and should not be	• OABRParams.txt

	changed.	
ROLLUP_SUBMIT_FILE_NM	Report File for Budget Transactions (.txt) For system use and should not be changed.	• OABRRollupSubmit.txt

Major Output

- Budget Roll Crosswalk of Successful Transactions Report
- Budget Roll Listing of Unsuccessful Transactions Report

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and report was generated successfully
Warning (4)	• Job does not end with this return code
Non-Fatal Error (8)	At least one transaction failed to submit
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

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. Restarting has a condition that no Budget Roll chain process should be scheduled between the time frame the job was discontinued and is being restarted and the parameters provided to the chain process should not be changed.

- If the report lists any failed transactions due to wrong data setup that caused the budget adjustments to fail, then following steps can be executed to resubmit the failed rolling transactions.
- Correct any data setup problems that caused transactions to fail. Often it is just a very limited number of control table settings.
- If the number of failed transactions is low, they can be manually submitted. If the number is beyond manual intervention, then an independent SMU job can be run to submit the modifications.
- After either method, a new chain can be submitted starting with the Budget Roll Reports to verify all were submitted successfully. It is critical to ensure that no parameters are changed on the Parameter table and no roll process has run in update mode between the actual run and restart run, else any .txt files marked for system use (see parameter list above) are overwritten and the restart run will not work correctly.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and job completed successfully	N/A	N/A
Warning (4)	Job does not end with this return code.	N/A	N/A
Non-Fatal Error (8)	One or more budget transactions failed to submit.	Need to complete submit process after the reason for the failure is resolved.	See problem resolution above
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following conditions. 1) Encounters any runtime exceptions 2) Invalid parameter found 3) text files could not be found 4) Exception report file could not be found. If the job fails because of these, investigate the problem. Need to schedule a new chain starting at this point after reason for the failure is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Need to schedule a new	Ensure no OABR chain has started with the Pre Process job

		chain starting at this point after reason for the termination is resolved.	step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. Need to schedule a new chain starting at this point after reason for the failure is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity and Budget Roll Chain: Post Roll Update Job

Job Name	Post Roll Update
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

This job restores the Auto Generation flag in the REQBUD table for all Budget Structures in the target fiscal year to the original value before Job 1 – **Roll Update Preprocess** was run.

Process Steps	Messages
1. Parameter validation	• Batch input parameters are listed. If any is invalid, an error message is issued stating such.
2. Restore REQBUD	• No messages

Major Input

- Required Budget (REQBUD / R_FN_BUD_RULE)

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSPARM	Parameter Location at Post Roll	• $$AMSROOT$$/Par

	Update Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	ms
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) For system use and should not be changed.	• OABRParams.txt

Major Output

- Required Budget (REQBUD / R_FN_BUD_RULE)

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and report was generated successfully
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 • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

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

	successfully and job completed successfully		
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 conditions. 1) Encounters any runtime exceptions 2) Invalid parameter found 3) text files could not be found 4) Exception report file could not be found. If the job fails because of these, investigate the problem. Need to schedule a new chain starting at this point after reason for the failure is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Need to schedule a new chain starting at this point after reason for the termination is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. Need to schedule a new chain starting at this point after reason for the failure is resolved.	Ensure no OABR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

2.1.23 Open Activity & Budget Update

Job Name	Open Actv & Budget Update
Recommended Frequency	Never run as a stand-alone job, but automatically run by either the Open Activity Roll or Open Activity & Budget Roll chain jobs.
Single Instance Required	No
Can be restarted?	Yes
Reports generated	No

Overview

The Create Round 1 and Create Round 2 jobs of the Open Activity Roll and the Open Activity & Budget Roll chain jobs spawn this batch job to load modification drafts of transactions rolling. Those modifications are also populated with data to support the roll, to include:

- Transaction Description of “Roll Transaction from (Source BFY) to (Target BFY)”
- BFY changed at the accounting line to the Target BFY value
- Any COA crosswalked

The Create Round jobs will keep running until all spawned Open Actv & Budget Roll jobs are spawned and completed. When those are done, this job step will end.

Major Input

- OABR Workload ((OABR_WRKLD)
- Parameter file (OABRParams.txt)
- COA Crosswalk (COAX / R_COA_CROSSWALK)

Batch Parameters

This job has no batch input parameters.

Major Output

- Modification drafts loaded with modified data

Job Return Code

Return Code	Condition
Successful (1)	Job completed by loading all transactions assigned, even if no transactions were in the round.
Warning (4)	Job does not end with this condition

Non-Fatal Error (8)	Job does not end with this condition
Failed (12)	The job will fail under the following conditions: • Create Round job was run incorrectly when Run Mode of RLPA ID was *Pre Selection* so that the Open Actv & Roll Update found no information. • Transaction modification failed to load successfully • Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process was discontinued for some reason (for example, job was killed) the user has an option to restart the job and the process will continue from the point it left off. This restart has a condition that no roll process must be scheduled between the time frame the job was discontinued and is being restarted. The Restart option is provided for jobs – "Create Round n" where n is 1, or 2 as well as the spawned jobs for those different Create Rounds.

In the case of a system failure during any Create Round, all spawned Create jobs (Open Actv Roll Update + <chain job id>) will return a Return Code of *Failed* or *System Failure*. The Create Round job will also return a *Return Code* of *Failed*, and further jobs down the chain are *Inactive*. In this case, the user should first restart each spawned Open Actv Roll Update job so that individual job can complete work on the set of transactions it was assigned. Then after each spawned job completes successfully, the user should restart the Create Round job. If there is further work to be done in that Create, the chain will spawn additional Open Actv Roll Update jobs to complete that round's work. After all the applicable work is completed, the chain will automatically move on to the next job in the chain.

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	Job does not end with	N/A	N/A

Error (8)	this return code		
Failed (12)	Job failed due to Fatal conditions.	Job can fail under the following conditions. 1) Modification fails to load successfully 2) Runtime exception. The problem needs to be addressed and the job restarted.	N/A
Terminated (16)	Job is terminated manually by the user.	After the reason for the termination has been resolved, the job should be restarted if the chain should continue.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	After the reason for the system failure has been resolved, the job should be restarted so the chain can continue.	N/A

2.1.24 Open Activity Lapse

Chain Job Name	Open Activity Lapse
Recommended Frequency	To be run as the end of the Budget Fiscal Year approaches, is reached, or has passed for a given amount of time. The timing of when the chain is run depends on client needs, which can even vary by the type of accounting activity so that one type is lapsed before another. One batch program should be run prior to the chain. The Matching chain job so that any purchase orders that have met matching rules for being received and/or invoiced are closed out with a payment request and not lapsed. Refer to the Match Payment Creation run sheet in *the CGI Advantage Financial - Procurement Run Sheets* guide for more information. Orders with only half of the matching requirements (that is received but not invoiced or the opposite) require manual attention or they will be lapsed.
Single Instance Required	Yes
Can be restarted?	See individual jobs for more restartability information
Reports generated	Report is generated for transactions being Lapse and exception report if there is error during submitting the Lapse Transactions.

Overview

When a year comes to a close, there are many transactions still un-referenced with accounting activity that has not yet reached its final state. The Open Activity Lapse (OAL) chain job lapses the activity in the old year, leaving it up to the sites to re-establish it in the new year manually. The lapse is done by a new transaction doing a zero-dollar reference that is a final reference. This way the transaction being lapsed remains unmodified, but has the open amount reduced to zero dollars.

The primary reason for lapsing open activity is that such activity is not desired in the fiscal year at the time of annual close. The alternative to lapsing is rolling the open activity. It is a policy decision for each site to determine what rolls and what lapses. The two processes can even act upon the same type of accounting activity, with one taking certain records and leaving the other with the remainder. One such example is where all pre-encumbrances over $100K are rolled using the Roll Minimum amount first, then a lapse is performed on all those pre-encumbrances that remain.

The OAL chain can lapse several types of activity. What activity is a function of specifying one or more event types on the Parameters for Lapse Process (LPPA) page. The Fund (FUND) page does contain several override controls for use when LPPA selection is not based on fund codes.

Accounting Activity	Examples
Pre-Encumbrances	Event types with a posting code defined to update the Pre-Encumbrances (12) budget amount. PR02 and PR03 are two such delivered event types.

Encumbrances	Event types with a posting code defined to update the Encumbrances (13) budget amount. PR05 and PR06 are two such delivered event types.
Receivables	Event types with a posting code defined to update the Billed Earned Revenue (22) or Billed Unearned/Deferred Revenue (25) budget amount. AR01 and AR10 are two such delivered event types.
Other Items	Event types with a posting code defined to update no budget amount, while not also defined to make a Disbursement Request update. PR01 and PR07 are two such delivered event types. Event types without posting codes. PR08 is one such delivered event type.

Note – The system looks only at posting pair A for an event type to see what type of accounting activity is booked.

Such activity recorded on the Requisition (RQ), Purchase Order (PO), Accounting Based Spending (ABS), Receivable (RE), and Travel (TRVL) transaction types can be selected. If the Allow Fund Action flag for the type of accounting activity is checked on the LPPA parameter ID used, then the Close Action at the fund code level will be read, which is on Fund (FUND). Further selection of activity is then based on settings on the Parameters for Lapse Process (LPPA) table. If the FUND option is other than *Lapse* then no open activity matching LPPA selection criteria will be selected for that fund code.

Notes:

1. The Lapse programs currently do not support lapsing Accrued Receivable (ARE) or Stock Requisition (SRQ) transactions.
2. Travel Authorization (TRAUTH) is the only Travel Transaction Sub Type within the Travel (TRVL) Transaction Type that can be lapsed.

Lapsing Receivable transactions with the lapse program does not perform any aging such as over 365 days old. There is another process for that in the Accounts Receivable chain job folder called the Generate Write Off. The OAL lapses purely on budget fiscal year.

The chain process comprises of following batch jobs below that are given with very brief summaries. Parameters have to be entered at job #1 only. More details can be found later in the individual job steps.

Job 1 **Lapse**

Batch parameter validation is performed. Selection of records from either the Detail Pre Selection table or directly from the transaction catalogs is next. The job when run in update mode only, writes selected records to an XML file for loading in later processing. Lastly, a report is produced to show lines selected for lapsing.

Job 2 **Load and Submit Lapse Transaction**

This job step produces one or more System Maintenance Utility (SMU) jobs to load the XML file created in the Lapse job step and then submit those transactions. The number of jobs is parameter-driven.

Job 3 **Transaction Exception Report**

A report is compiled to list those transactions that failed to submit from the prior job step.

If using an LPPA ID with a Run Mode of *Pre-Selection*, only job #1 of the chain is needed. All others should be disabled. Please remember to check all the Disable flags and hit save to fully disable all other jobs in the chain. If not, job #2 will fail, but that is really an acceptable outcome as there is nothing to load.

If using an LPPA ID with a Run Mode of *Update*, the whole chain should be run.

If using an LPPA ID with a Run Mode of *Report*, do not use the chain job. There is a job in the Reports-GA folder of General Accounting called Open Actv Lapse that should be used. If the chain is used, it will fail on the first job step with an error stating that the Run Mode was incorrect for the chain.

A good practice is to query for any remaining open activity that should have lapsed. This can be done from a variety of sources: ledgers, journals, and the BBAL Fiscal Year Details (BBALFY) page. If ledgers are to be used, then ensure the Ledger Engine has run to record all previous lapsing activity.

Pre-Processing Tasks:

1. If a site is to control lapsing the same for all funds, then they should ensure that all the override flags on LPPA are unchecked to prevent Fund lookups that will return the same value or worse, different values that are not intended.
2. Sites should run the Open Activity Lapse report, setting up the Run Mode parameter to report only for the LPPA ID. The report provides an activity count to gauge run times by displaying the open accounting lines that will be lapsed. One of the fields on the report also shows if a pending version of any final transactions to be lapsed exists. Final transactions with Pending versions are skipped. The Open Activity Lapse report has a setting that will generate a text file (*.txt) of all transactions in a Pending state, which can then be input to a SMU job to reject all pending transactions back to Draft. When this is done before a lapse, all transactions will be lapsed and the pending ones will not be skipped.
3. For sites where the selection criteria on LPPA is not sufficient, sites should run with the *Pre-selection* mode by setting up the Run Mode parameter to *Pre-selection* and run only the first job step of the OAL process. Using these pages, a user can decide which transactions or separate accounting lines to lapse. When the OAL process is then run in update mode and the Pre-selection table that was created during the pre-selection mode is to be used as input, sites should remember to set the USE_PRETABLE batch parameter to True or else the transaction catalogs will be used.
4. Negative accounting lines that are open will not be lapsed. They will have to be dealt with manually.
5. To ensure fewer transaction lapse failures, sites must ensure the transaction codes used to lapse are established on Transaction Control (DCTRL) with the following flags checked: Approval Override Allowed and Inactive COA Codes Allowed.
6. This batch program produces transactions which are subject to the line limit functionality constraints defined on the Transaction Component Requirements (DCREQ) in the Administration application. Sites must ensure that limits have not been set to lower limits than existing transactions being lapsed contain. Additionally, if there is a limit for budget transaction lines, then the Transaction Break parameter on PBRP should be set to ensure that limit is not exceeded. For reference, the CBDL transaction code is in the PO transaction type and the ABDL is in the ABS transaction type.
7. All transaction codes on the Parameters for Lapse Process should have valid Automatic Transaction Numbering entries.
8. Run first in a test environment that is a recent copy of production to shake out setup problems and other system configurations that can cause lapses to fail.

Major Input

- Transaction Accounting Line Catalog (DOC_ACTG)
- Lapse Parameters (LP_PROC_PARM)

Peripheral Input

- Fund (FUND / R_FUND)
- Posting Code (PSCD / R_PSCD)
- Event Type (ETYP / R_EVNT_TYP)
- Auto Transaction Numbering (ADNT / AUTO_DOC_NO)
- Fiscal Year (FY / R_FY)
- Transaction Control (DCTRL / R_GEN_DOC_CTRL)
- Department (DEPT / R_DEPT)
- Appropriation (APPR / R_APPR)
- Appropriation Type (APTYP / R_APTYP)
- Program (PROG / R_PROG)
- Accounting Period (APD / R_APD)
- Transaction Header (*_DOC_HDR)
- Transaction Vendor (*_DOC_VEND)
- Transaction Commodity (*_DOC_COMM)
- Transaction Accounting (*_DOC_ACTG)

*Items above with a * denote that different values may be substituted in place of the *.*

Major Output

- Generated Lapse transactions (Commonly = CBDL, ABDL and WO)
- Open Lines Lapsed Report listing Lapsed and Lapsing transaction information.
- Lapse Transaction Detailed/Summarized Exception Report
- Updated budget lines (BQ*LV* / BUD_STRU_*_LVL_*)
- Updated allotment lines (A LOT_STRU_*_LVL_*)
- Updated Accounting Journal (JACTG / JRNL_ACTG)
- Transactions lapsed

Chain Job Return Code

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

Return Code	Condition
Successful (1)	All 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 on each job step, but a couple of the more common are presented here at the chain job level.

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process was discontinued for some reason (return codes 12, 16, or 20) the user has an option to restart the job step that stopped and the process will continue from the point it left off. This restart has a condition that no lapse process must be scheduled between the time frame the job was discontinued and is being restarted. The Restart option is provided for jobs 1, 2 and 3, namely “Lapse”, “Load Lapse Transaction” and “Submit Lapse Transaction”.

- If any lapse transactions fail to submit, the Transaction Exception Report will end with a return code of 8. Reasons for the transaction failures should be addressed and those transactions submitted manually or with an SMU job. For assurance purposes, a new instance of the chain can be run with only the Transaction Exception Report enabled to verify all lapse transactions submitted successfully.

Open Activity Lapse Chain: Lapse Job

Job Name	Lapse
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	Yes – Please see the Problem Resolution section for details
Reports generated	Yes - Open Lines Lapsed

Overview

This first job step starts with basic editing of parameters and goes on to perform accounting line selection based on Run Mode;

If *Pre-Selection,* then the job will look to the transaction catalogs for open accounting lines that match LPPA and optionally FUND criteria. Selected accounting lines not already on RLPSD will be added. Open accounting lines already on RLPA will be updated for any information that has changed since the last update. If the RLPSD record has a Selection Date and the accounting line is still open, that date will be cleared. Any RLPSD records found that are no longer open with a blank Selection Date will be removed.

If *Update*, the job uses the 'Use Pre-Selection' batch parameter to determine where to look for open accounting lines. If that parameter is 1 – Use Pre-Selection, then selection will be from the Roll Lapse Pre Selection Detail (RLPSD) page for records with an Action = *Lapse*, a checked Approved flag, and a blank Selection Date. Even after all this selection, if the accounting line is no longer open, it will not be selected. Be aware that when using RLPSD as input, selection criteria on LPPA and FUND are not read, as they were used when RLPSD was loaded.

When the job is run in *Update* mode, it writes selected records to an XML file for loading in later processing. Lastly, a report is produced to show lines selected for lapsing.

Process Steps	Messages
1. Parameter Validation	• Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log.
2. Selection of Records	• Records processed: XX • Rendering report started. • Rendering report completed.
3. Report Creation	• Reports output folder mapped • Rendering report started. • Rendering report completed.

Major Input

- Transaction Accounting Line Catalog (DOC_ACTG)
- Lapse Parameters (LP_PROC_PARM)
- Roll/Lapse Pre-Selection Detail (RLPSD / RLLP_PRE_DET)

Peripheral Input

- Fund (FUND / R_FUND)
- Posting Code (PSCD / R_PSCD)
- Event Type (ETYP / R_EVNT_TYP)
- Auto Transaction Numbering (ADNT / AUTO_DOC_NO)
- Fiscal Year (FY / R_FY)
- Transaction Control (DCTRL / R_GEN_DOC_CTRL)
- Department (DEPT / R_DEPT)
- Appropriation (APPR / R_APPR)
- Appropriation Type (APTYP / R_APTYP)
- Program (PROG / R_PROG)
- Accounting Period (APD / R_APD)
- Transaction Header (*_DOC_HDR)
- Transaction Vendor (*_DOC_VEND)

- Transaction Commodity (*_DOC_COMM)
- Transaction Accounting (*_DOC_ACTG)

Batch Parameters

For the entire chain process, a user has to enter parameters for Job 1 – Lapse. The process propagates required parameters down the chained jobs. Some of them are non-overrideable parameters, which will be provided with the Job Setup and need not be specified for each run. Many of these are just meant for system use and should not be changed. Those are so noted in the Description column.

Parameter	Description	Default Values
AMSEXPORT	Export Location at Lapse Job	
AMSPARM	Parameter Location at Lapse Job	
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) For system use and should not be changed.	LapseParams.txt
CHK_PT_SIZE	Checkpoint Block Size (Number of transactions processed after which checkpoint is to be updated)	100
CLIENT_NM	Client name for “Open Lines Lapsed Report” and “Lapse Transaction Detailed/Summarized Exception Report”.	
COMMIT_SIZE	Commit Block Size for Transaction Submit (required). This figure determines the number of transactions to be committed at a time for all jobs in chain that perform transaction load and submit and can be used for performance enhancement depending upon the server memory configuration.	1
DT_LAST_ACTV	An optional date for the specification of an aging selection parameter that will lapse entire transactions that have not had any activity since this date, including references as well as modifications. The parameter is intended for lapsing BFY 9999 activity but can also be used to lapse regular BFY activity that keeps rolling year to year but never gets referenced.	(blank)
EXCP_RPT_TYP	Exception Report Type for “Lapse Transaction Exception Report” (1=Detailed, 2=Summarized, if not entered then defaulted to Detailed)	1
EXP_FILE_NM	Transaction XML File(.xml) For system use and should not be changed.	LapseDoc.xml
LOAD_DATA	How to load the Detail Pre-Selection Table (1= Load all records with the approval flag	1

	checked, 2 = Load all records with the approval flag unchecked)	
LOAD_FILE_NM	Load Parameter File(.txt) For system use and should not be changed.	LapseLoad.txt
LP_PROC_PARM_ID	If you are running the job in the Financial application, you can click on the **Custom Parameter** link and select a parameter record. If you are running the job in the Administration application, enter a valid Parameter ID from the Custom Parameter table from the Financial application.	
PRE_SEL_COMMIT_BLOCK	Commit block Size for loading Pre-Selection tables. If not entered then defaulted to 1000. It is the number of records that are loaded into Pre-Selection tables after which a commit is issued. Can be used for Performance tuning. Too low value can cause frequent database commits, thereby increasing processing time. Too high value will demand for higher memory. Can be used for performance enhancement depending upon the server memory configuration.	1000
PRE_SEL_SELECT_BLOCK	Select block Size for loading Pre-Selection tables. If not entered then defaulted to 1000. It is the number records that will be fetched as a single block from Transaction Catalog Accounting Table. Can be used for Performance tuning. Too low value can cause higher number of database fetches, thereby increasing processing time. Too high value will demand for higher memory. Can be used for performance enhancement depending upon the server memory configuration.	1000
PURGE_DATA	How to purge the Detail Pre-Selection Table (1= Purge all records, 2 = Purge only approved records and 3 to purge closed records only record Parameter is read only for Pre Selection mode. Use parameter 2 or 3 when 'refreshing' or loading additional data to RLPSD. Use parameter 1 for the first run of a new year.	1
SUBMIT_FILE_NM	Submit Parameter File(.txt) For system use and should not be changed.	LapseSubmit.txt
USE_PRETABLE	Use Detail Pre-Selection Table (1= Use Pre-Selection table as input, 2= do not use Detail Pre-Selection table but use transaction catalogs.)	1

Custom Batch Parameters (LPPA)

Please keep in mind that when using the COA selection fields of Department, Program, Appropriation, and Appropriation Type that these elements may not exist on all event types being selected. If an event type is selected and used without one of these elements used as selection criteria on LPPA, the accounting line will not be selected.

Also, if the selection criteria are not enough to identify what is to be lapsed, then the pre selection mode is available to load the Roll/Lapse Pre Selection Summary and Detail tables for manual selection. Reports can be run to take records from the detail table and join with other information to produce a very detailed listing of potential records for selection.

Field	Description	Edits
Parameter ID	Unique Parameter ID	Error issued if no value is entered
Closing BFY	Budget Fiscal Year for selection of records to lapse. Please enter as CCYY.	Error issued if the year does not exist on R_FY. Error issued if the field is null Error issued if multiple years are entered In each case, the job does not run successfully and an error is issued in the log.
Run Mode	CVL with values-Report Only, Update, and Pre-Selection	Error issued if the value is blank. In each case, the job does not run successfully and an error is issued in the log. Error will be issued when Mode is not pre-selection and purge data parameter set to 2 or 3.
Event Types	Enter event type(s) for selection of open activity. Commas should separate multiple event types.	Error issued if the event type does not exist on R_EVNT_TYP. Error if the Expense Budget Bucket ID is not 12 (pre-encumbrance) or 13 (encumbrance) and the Expense Budget Bucket Update flag is checked for either posting code in Posting Pair A of the event type. Error is the Revenue Budget Bucket ID is not 22 (billed earned revenue) and the Revenue Budget Bucket Update flag is checked. Error is issued if event type entered is marked as Disbursement Request Update on Event Type. Error issued if the field is null. In each case, the job does not run successfully and an error is issued in the log.
Transaction Codes	Optional transaction code(s) for selection of	Error issued if the transaction code does not exist on R_GEN_DOC_CTRL.

	open activity. Commas should separate multiple transaction codes. This parameter can enter text up to 255 characters in size.	Error is issued if the Transaction code entered are not of transaction type ABS, RE, PO, RQ, or TRVL. In this case, the job does not run successfully and an error is issued in the log.
Funds	Leave blank for selection of all funds. Individual funds can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any fund code entered is not valid for the closing BFY as FY on the Fund (R_FUND) table, and job does not run successfully.
Departments	Leave blank for selection of all departments. Individual departments can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any department code entered is not valid on the Department (R_DEPT) table, and job does not run successfully.
Appropriations	Leave blank for selection of all appropriations. Individual appropriations can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any appropriation code entered is not valid for the closing BFY as FY on the Appropriation (R_APPR) table, and job does not run successfully.
Appropriation Type	Leave blank for selection of all appropriation types. Individual appropriations can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any appropriation code entered is not valid for the closing BFY as FY on the Appropriation (R_APTYP) table, and job does not run successfully.
Programs	Leave blank for selection of all programs. Individual programs can be listed, separated by commas if multiple. (Blank) means program is not part of the selection criteria. This parameter can enter text up to 255 characters in size.	Error issued if no department code is entered and field is not blank. Error issued if more than one department code is entered and more than one program is entered. Multiple values of each are not allowed as the batch job would not know what combinations to form. Multiple runs are required in this situation. Error issued if any program code entered is not valid on the Program (R_PROG) table for the department. In each case, the job does not run successfully and an error is issued in the log.

Fiscal Year	• Optional fiscal year (FY) to be placed on all lapsing accounting lines. The value will be used on all posting lines. Must be entered if lapsing after the start of a new year. • Please enter as CCYY. Leave blank to default from Transaction Record Date (see parameter below) if used or the Application Date when the lapse is performed.	Error issued if the FY does not exist on the Fiscal Year (R_FY) table. Error issued if multiple periods are entered It can be blank when Run mode is Report only In each case, the job does not run successfully and an error is issued in the log.
Accounting Period	• Optional accounting period (APD) to be placed on all lapsing accounting lines lapsed. The value will be used on all posting lines. Must be entered if lapsing after the start of a new year. • Please enter as seen on the Accounting Period table. Leave blank to default from Transaction Record Date (see parameter below) if used or the Application Date when the lapse is performed.	Error issued if the APD does not exist on the Accounting Period (R_APD) table. Error issued if multiple periods are entered Error issued if entered and is '99' or '0'. In each case, the job does not run successfully and an error is issued in the log.
Transaction Code for Commodity-based Spending Transactions	Specific transaction code to be created to lapse transactions with commodity lines.	Error issued if the transaction code not exist on R_DCTRL with the PO transaction type. Error issued if multiple codes are entered Error issued if field is blank. In each case, the job does not run successfully and an error is issued in the log.
Transaction Code for Non-Commodity Spending Transactions	Specific transaction code to be created to lapse transactions without commodity lines.	Error issued if the transaction code not exist on R_DCTRL with the ABS transaction type. Error issued if multiple codes are entered Error issued if field is blank.

		In each case, the job does not run successfully and an error is issued in the log.
Transaction Code for Revenue Transactions	Specific transaction code to be created to lapse transactions that contain receivables.	Error issued if the transaction code not exist on R_DCTRL with the WO transaction type. Error issued if multiple codes are entered Error issued if field is blank. In each case, the job does not run successfully and an error is issued in the log.
Spending Transaction Lapse Event Type	Event type to be used when lapsing the transactions	Error issued if the event type does not exist on R_ETYP Error issued if event types are entered Error issued if field is blank. In each case, the job does not run successfully and an error is issued in the log.
Record Date	Optional Transaction Record Date for lapsing transactions. The date will be used instead of a defaulting of the Application Date, and will supply an FY and APD for the accounting line if the above parameters for the same are not used.	Usually transactions generated are stamped with Date of Record at the time of transaction submit. If the process is run over night and the Begin Day job runs then generated transactions will have different dates of record. Hence to have a consistent date of record on all lapsing transactions supply a value to this parameter. If not entered then the parameter is defaulted to the current date at the time of parameter processing by job1. If invalid date is entered an error is logged.
Transaction Department	Transaction Department Code for Generated Lapse Transactions	All generated lapse transactions will have transaction department set to this parameter value. The combination of Closing BFY, Transaction Code, Transaction Department Code and Prefix is used to pick up the ADNT (Auto Transaction Numbering) Record for Lapse Transaction Id numbering. If Transaction Department Code is left blank, then process derives Transaction Department for Lapse transactions from the Transaction Department of the original Lapsed (open) transaction. The value is used to pick the ADNT record. If ADNT record was not found then ADNT entry with Transaction Department value “****” (implies any Transaction Department) is picked up. If this record is not found then

		an error is logged to define ADNT entry.
Transaction Prefix	Prefix for Generated Lapse Transactions	All generated lapse transactions will have transaction id prefixed with this parameter value. The combination of Closing BFY, Transaction Code, Transaction Department Code and Prefix is used to pick up the ADNT (Auto Transaction Numbering) Record for Lapse Transaction Id numbering. If Prefix is entered then process looks for ADNT record with the given prefix. If this record is not found then an error is logged to define ADNT entry for given prefix. If prefix is left blank, then process looks for ADNT record with prefix value "****" (no prefix) is picked up. If this record is not found then an error is logged to define ADNT entry.
Allow Fund Close Actions: Pre Enc Enc Recv Other Items	When set to YES, an individual fund code may supply a different close action.	N/A
Lapse Threshold Pre Enc Enc Recv Other Items	A minimum dollar amount which an open accounting line or transaction must be less than for selection. Set very high to ensure lapsing of all matching open activity.	N/A
Allow Fund Lapse Threshold Pre Enc Enc Recv Other Items	When set to YES, an individual fund code may supply a different threshold.	N/A
Lapse Threshold Transaction/Line Pre Enc Enc Recv Other Items	The setting determines at which level the threshold is applied: transaction or accounting line.	N/A
Date Selection	The setting determines the date to be considered for selecting the transaction for lapse based on the date	N/A

	entered in the Date of Last Activity batch parameter. • Record Date enables selection for the last modification, including the first version if never modified. • Last Update Date enables selection based on the last reference or modification.	

Major Output

- XML transaction file (LapseDoc.xml)
- LapseLoad.txt
- LapseSubmit.txt
- Open Lines Lapsed Report

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and selection was successful, and at least one record was selected. For pre-selection mode, if no records are selected, the return code is successful.
Warning (4)	The job does not end with this return code
Non-Fatal Error (8)	No records were selected for updated mode based on FUND and LPPA
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations. • Restart failed because another instance of the job has already been run successfully When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

- Transaction Department Code (DOC_DEPT_CD)
- Transaction Code (DOC_CD)
- Transaction ID (DOC_ID)
- Transaction Version Number (DOC_VERS_NO)
- Transaction Vendor Line Number (DOC_VEND_LN_NO)
- Transaction Commodity Line Number (DOC_COMM_LN_NO)
- Transaction Accounting Line Number (DOC_ACTG_LN_NO)

Selection Criteria

1. Selection criteria for obtaining Transaction Accounting lines (DOC_ACTG) to be lapsed is –where the Accounting Line value below matches the LPPA value:
    a. BFY matches Closing BFY
    b. Event Type matches one of Event Type Selection Criteria
    c. Transaction Code matches one of the optional Transaction Codes
    d. Fund matches one of the optional Fund Selection Criteria
    e. Department (accounting line one and not Transaction Department) matches one of the optional Department Selection Criteria
    f. Appropriation matches on of the optional Appropriation Selection Criteria
    g. Appropriation Type matches on of the optional Appropriation Type Selection Criteria
    h. Program matches one of the optional Program Selection Criteria
    i. Open Amount is > $0.00
2. If the respective Fund Override flag is checked, get respective Closing Action by lookup to FUND using Closing BFY as FY. Please see the Chain Job overview for more information on matching action to type of open activity.
    a. If that FUND action is *Lapse*, then select.
    b. If that Fund action is other than *Lapse*, do not select.
3. When using the RLPSD table for selection, if the Approved flag is checked, the Action is *Lapse* and the Selection Date is blank, then select.
4. When using the Last Action Date parameter to select transactions without any activity since a pre-defined date, the Selection Date on LPPA is used as follows. In both cases, if the transaction date is before the Last Action Date, the transaction will be considered.
    a. If LPPA is *Last Update Date,* then selection will compare the Modified On header date against the Last Action Date.
    b. If LPPA is *Record Date,* then selection will compare the Record Date on the header against the Last Action Date.

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If no records are selected, review the selection criteria on the LPPA ID used. Also, review the FUND page to verify that the appropriate Action field has a value of *Lapse*.

If the chain process was discontinued for some reason (for example, job was killed) the user has an option to restart the job and the process will continue from the point it left off. This restart has a

condition that no lapse process must be scheduled between the time frame the job was discontinued and is being restarted.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated and XML created successfully.	N/A	N/A
Warning (4)	Job does not end with this return code.	N/A	N/A
Non-Fatal Error (8)	No records were selected	Review FUND and LPPA settings to ensure desired records get selected.	Run in report mode until parameters are correct as it is an easier job to run. Then change LPPA to Update or Pre Selection and run the chain.
Failed (12)	Job failed due to fatal conditions.	Job can fail under the following conditions: 1) Invalid parameter 2) Runtime exception. Need to reschedule or restart the job after reason for the failure is resolved.	Ensure no OAL chain has started with the Lapse job step (#1) in the interim. If so, this chain is effectively over.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Need to reschedule or restart the job.	Ensure no OAL chain has started with the Lapse job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the failure needs to be investigated. Need to reschedule or restart the job.	Ensure no OAL chain has started with the Lapse job step (#1) in the interim. If

			so, this chain is effectively over.

Open Activity Lapse Chain: Load and Submit Lapse Transaction Job

Job Name	Load Lapse Transaction
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	Yes – Please see the Problem Resolution section for details
Reports generated	No

Overview

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

Process Steps	Messages
1. Parameter Validation	• Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log.
2. Multiple Job Processing	• Setting up job for parameter file: (Export Import directory path)\ACTVLAPSEParm_######_#.txt • Enabled job with Id=###### • Created Job ###### for Parm file (Export Import directory path)\ACTVLAPSEParm_######_#.txt (Repeated for each file and job combination) • Committed child jobs • Child job pending count:# Job Id[######] Status=X (Repeated for each job) • Deleting part exception file: (Export Import directory path)\null_######_#.txt (Repeated for each file) • Deleting split input XML file: (Export Import directory path)\ACTVLAPSEXML_######_#.xml (Repeated for each file) • Run Ended

Major Input

- Lapse transaction xml

Batch Parameters

There are performance parameters in this second job in the chain that should be adjusted at least once for default values if not for each run depending on volume.

Parameter	Description	Default Values
BLOCK_SIZE	(Required) Number of records in each split segment of the input file.	100
COMMIT_BLOCK_SIZE	(Required) Number of records to commit at a time.	1000
FILE_INPUT_DIR	(Required) The location of the source files of records.	$$AMSROOT$$/ExportImport
FILE_LIST	(Required) Comma separated list of files to be uploaded with multi-threaded processing.	LapseDoc.xml
FILE_OUTPUT_DIR	(Required) Output location for the file segments.	$$AMSROOT$$/ExportImport
FILE_PREFIX	(Required) Prefix used on the filenames for the output file segments.	ACTVLAPSE
I_SMU_BYPS_ADNT_FL	(Required) Bypass ADNT Flag	true
I_SMU_COMMIT_BLOCK	Commit for Import	150
LOG_STATUS_INTERVAL	(Required) Logging frequency (in seconds) for controller thread reporting status of child threads to the system log.	30
MODE	(Required) Mode of operation. (1=Import, 2=Import and Submit, 3=Import and Other Action)	2
SLEEP_INTERVAL	(Required) Polling frequency (in seconds) for internal controller thread for checking child processes.	5
SMU_CTLG_ID	(Required) Catalog Id of the System Maintenance Utility job, which is spawned as the child process.	3
STAGGER_TIME	(Required) The lag time, in seconds, between the spawning of each child process.	30
THREAD_COUNT	(Required) Parameter defines the number of SMU threads to use for processing.	5

Major Output

- ABDL transactions submitted to final in rejected status.
- CBDL transactions submitted to final in rejected status.
- WO transactions submitted to final in rejected status.

Job Return Code

Return Code	Condition
Successful (1)	All of the transactions loaded were submitted successfully.
Warning (4)	This return code will be issued when some of the records failed to load where as all other records were loaded successfully.
Non-Fatal Error (8)	This return code will be issued when all records failed to load.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid. • When the input file is not found in the specified directory. • Restart failed because another instance of the Accrual Process chain has already been run successfully. • Runtime exceptions encountered for any unexpected situations. When the job ends with a return code of Failed, subsequent jobs in the chain will be set to Inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

- The following table shows the possible return codes and recommendations for each processing step specific to the job in the chain. For general errors and recommendations, refer to the SMU Transaction Upload run sheet in the CGI Advantage Financial – Utilities Run Sheet Guide.
- If lapsing transactions failed to submit, errors can be corrected on them individually or on the setup page that caused the failure. An individual SMU job can be run to submit those

transactions or they can be discarded and the Lapse process run again to lapse the remaining open activity.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated and XML successfully loaded.	N/A	N/A
Warning (4)	At least 1 transaction did not load causing a spawned job to end Non Fatal.	Correct what is most likely a setup issue and restart the spawned SMU job and then restart the load/submit job.	N/A
Non Fatal Error (8)	None of the transactions loaded	Correct what is most likely a setup issue and restart the spawned SMU job and then restart the load/submit job.	N/A
Failed (12)	In this step, the job can fail under the following conditions: • Issues in spawning jobs • Files not found • Runtime exceptions • Parameter Edits	The reason for the failure needs to be investigated and fixed before restarting the job.	Ensure no OAL chain has started with the Lapse job step (#1) in the interim. If so, this chain is effectively over.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Need to schedule a new chain starting at this point or restart the job after reason for the termination is resolved.	Ensure no OAL chain has started with the Lapse job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the failure needs to be investigated. Need to schedule a new chain starting at this point or restart the job after reason for the failure is resolved.	Ensure no OAL chain has started with the Lapse job step (#1) in the interim. If so, this chain is effectively over.

Open Activity Lapse Chain: Transaction Exception Report Job

Job Name	Transaction Exception Report
Recommended Frequency	See chain job.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes – Lapse Transaction Exception Report

Overview

This job in the Lapse chain generates the report for errors encountered during submitting the Lapse process. The report contains the information regarding the lapse transaction that failed to submit and an error message for the same.

Process Steps	Messages
1. Parameter Validation	• Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log.
2. Reporting	• Error records processed : XX • Rendering report started. • Rendering report completed.

Major Input

- Error log created from Submit Lapse Transactions job step

Batch Parameters

For the entire chain process, a user has to enter parameters for Job 1 – Lapse. The process propagates required parameters down the chained jobs. Some of them are non-overrideable parameters, which will be provided with the Job Setup and need not be specified for each run. Many of these are just meant for system use and should not be changed. Those are noted in the Description column.

Parameter	Description	Default Values
AMSPARM	Parameter Location at Transaction Exception Report Job	-
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) Parameter value should be setup same as for CHAIN_PARM_FILE in Job 1. For system use and should not be changed.	LapseParams.txt
SUBMIT_FILE_NM	Transaction List File (.txt) For system use and should not be changed.	LapseSubmit.txt

Major Output

- Lapse Transaction Exception Report

Job Return Code

Return Code	Condition
Successful (1)	All transactions submitted successfully.
Warning (4)	Job does not end with this return code.
Non-Fatal Error (8)	At least one transaction failed to submit.
Failed (12)	Parameters are invalid • When the input file is not found in the specified directory • Runtime exceptions encountered for any unexpected situations When the job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process is scheduled and run and job 4 throws a Lapse Transaction Detailed/Summarized Exception Report listing Transaction submission errors and the errors were due to wrong data setup then the following steps can be executed to resubmit the failed Lapse transactions.

1. Correct any data setup problems that caused the transactions to fail. Often it is just a very limited number of control table settings.
2. If the number of failed transactions is low, they can be manually submitted and then the Transaction Exception Report restarted to verify all were submitted.

3. If the number is beyond manual intervention, then a new OAL chain can be submitted with the first two job steps disabled.
4. Alternative to the new chain, an independent SMU can be submitted with the Parameter File name as LapseSubmit.txt. The remaining lapse transactions will then be submitted. See the job log of the SMU for any errors.
5. For any of these solutions, ensure that no Lapse chain process has to be scheduled in the interim. If so, the Submit Parameter (.txt) file will be changed.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated and the report was created with no failures reported.	N/A	N/A
Warning (4)	Job does not end with this return code.	N/A	N/A
Non-Fatal Error (8)	One or more transactions failed to submit.	Need to complete submit process after the reason for the failure is resolved.	See problem resolution above
Failed (12)	Job failed due to fatal conditions.	Job can fail under the following conditions: 1) Invalid parameter 2) Runtime exception. Need to schedule a new chain starting at this point after reason for the failure is resolved.	Ensure no OAL chain has started with the Lapse job step (#1) in the interim. If so, manual verification is required to see all lapse transactions submitted.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Need to schedule a new chain starting at this point after reason for the termination is resolved.	Ensure no OAL chain has started with the Lapse job step (#1) in the interim. If so, manual verification is required to see all lapse transactions submitted.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the failure needs to be investigated. Need to schedule a new chain starting at this point	Ensure no OAL chain has started with the

		after reason for the failure is resolved.	Lapse job step (#1) in the interim. If so, manual verification is required to see all lapse transactions submitted.

2.1.25 Open Activity Options by Department

Job Name	Open Activity Options by Department
Recommended Frequency	This process can be run any time after data has been loaded to the Roll/Lapse Pre Section pages by either a run of Open Activity Roll or Lapse in Pre Selection mode
Single Instance Required	This Job requires single instance
Can be restarted?	No
Reports generated	No

Overview

There is selection criteria on both the Roll Parameters (RLPA) and Lapse Parameters (LPPA) pages for Department so that a run of either can be done in a fashion that meets the needs of one or more departments entered in that selection criteria. If a site has a very large number of departments and a large enough number of them requires a run that is different from all the others, the size limit of the Department selection criteria field will be reached. That problem could be addressed with multiple runs to break up the large group; however, a better option is to use the Open Activity Options by Department (OAOD) page and the batch job by the same name.

The page allows for the specification of a value for the Action field on the Roll/Lapse Pre Selection Detail (RLPSD) page by department and transaction code. All open activity lines can be loaded to RLPSD with one Action setting based on where loaded by the Open Activity Roll, Open Activity & Budget Roll, or Open Activity Lapse chains in pre-selection mode. The OAOD batch process can then be used to change the Action for a subset of records to: *No Action, Roll, Lapse,* or *Accrue*. The process will also mark matching records as 'Approved' by checking the flag by that name.

The scenario is then to load the RLPSD page with the Approved flag unchecked (LOAD_DATA parameter set to 2) by either the OAR, OAL, or OABR chain jobs in pre-selection mode. Then run the OAOD batch process using data setup on the OAOD page to make changes to RLPSD records. Examples would be:

- If the department does not want records rolled or lapsed, use OAOD to set the Action to 'No Action'. The next run or OAR, OAL, or OABR that uses pre-selected data (USE_PRETABLE parameter set to 2) will not select those records.
- If the department wants to roll records instead of lapse them as loaded by OAL, use OAOD to set the Action to Roll. Then the next run of OAL in update mode that uses pre-selected data (USE_PRETABLE parameter set to 2) will not select those records. Instead they will be selected by OAR or OABR.
- For departments that like to make a manual decision on a transaction by transaction basis, then they do not need an OAOD entry as all RLPSD records would be 'Unapproved'. These users can go into the RLPSD or RLPSS pages and make manual updates.

The OAOD job may only be run once or may be run multiple times if either the OAR, OAL, or OABR jobs are run in pre-selection mode with the PURGE_DATA parameter set to 3 (refresh mode or 'purge only closed') so that data on the Roll/Lapse Pre Selection pages is refreshed. Such a run could load new records to the pages so that OAOD would need to be run again.

Please be aware that any manual changes made could be overlaid by a matching OAOD record on this second run.

Another scenario for running the OAOD job multiple times is that there is a batch parameter for Department. This way updates to RLPSD data can be made for all departments (leave parameter blank), for one department (enter just one), or several departments (enter multiple codes separated by departments).

The process steps in the batch job are:

1. Parameter validation of input batch parameters
2. Record selection from the Roll/Lapse Pre Selection Detail page using selected OAOD page records that matched batch parameters for BFY and Department.
3. Roll/Lapse Pre Selection Detail updates

Process Steps	Messages
1. Parameter Validation	• Run Started • *Each parameter is listed with value* • *Any parameter errors are listed* • Parameter Validation was successful or • Parameter Validation failed
2. Record Selection	• Processing record for: BFY = CCYY, Transaction Code = XXX and Department = YYY *(where values are supplied from OAOD record)*
3. Roll/Lapse Pre-Selection Updates	• Number of rows processed: ## *(where ## is the number of RLPSD records updated) This message will repeat if more records than the Commit Block size are updated* • In Commit • Total number of rows processed: ## • *If no matching RLPSD records updated*: No matching records were found on the Roll/Lapse Pre-Selection table for: BFY = CCYY, Transaction Code = XXX and Department = YYY • Run Ended

Major Input

- Open Activity Options by Department (R_OA_OPT_DEPT)

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

Job or Job # in a Chain	Parameter	Description	Default Values

Open Activity Options by Department	SELECT_BLOCK	(Required)This parameter defines the number of RLPSD records selected in a query. It exists for performance tuning.	3000
	PROG_CTR_SZ	(Required) This parameter defines the number of RLPSD records updated that will trigger a progression message in the job log.	5000
	COMMIT_BLOCK	(Required)This parameter defines the number of RLPSD records after which a commit will be performed. Please understand that the commit is not only for writing to the DB but also limits the memory used by the job. It exists for performance tuning.	1000
	BFY	(Required) Parameter for BFY to be used in selecting records from the Open Activity Options by Department table.	No default
	DEPT_CD	(Optional) Parameter for Department. This parameter will accept a comma separated list of Departments to be used in selecting records from the Open Activity Options by Department table.	No default
	USE_DEPT	Use Department (1 = Use Transaction Department Code (DOC_DEPT_CD), 2 = Use Accounting Line Department Code (DEPT_CD))	1

Output

- Roll/Lapse pre-selection table RLLP_PRE_DET

Chain / Job Return code

Return Code	Condition
Successful	All parameters were valid and the job encountered no errors. *Note: Not finding any matching OAOD records or matching RLPSD records to update will not cause the*

(1)	*job to end as warning.*
Warning (4)	Job does not end with this return code.
Non-Fatal (8)	Job does not end with 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 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

N/A

Selection Criteria

Select records from the R_OA_OPT_DEPT table for the specified BFY and DEPT_CD. Select all records from the R_OA_OPT_DEPT table for the specified BFY if DEPT_CD is left empty.

For each record selected from the above step, the job would select records from the RLLP_PRE_DET table if they meet the following criteria:

APRV_FL (Approve Flag) = 0 and

DOC_CD (Transaction Code) matches DOC_CD (Transaction Code) of the selected R_OA_OPT_DEPT (OAOD) record and

Based on the value specified on batch parameter USE_DEPT,

check if DOC_DEPT_CD (Transaction Department Code) or AL_DEPT_CD (Accounting Line Department Code) matches the DEPT_CD of the selected R_OA_OPT_DEPT (OAOD) record.

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
Warning (4)	Job does not end with this return code.	N/A	N/A
Non-Fatal (8)	Job does not end with this return code.	N/A	N/A
Failed (12)	Job failed due to Fatal conditions.	Check parameters	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated and a new job scheduled after the reason is addressed.	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 and a new job scheduled after the reason is addressed	N/A

2.1.26 Open Activity Roll

Chain Job Name	Open Activity Roll
Recommended Frequency	To be run as the end of the Budget Fiscal Year approaches, is reached, or has passed for a given amount of time. The timing of when the chain is run depends on client needs, which can even vary by the type of accounting activity so that one type is rolled before another. Three batch programs should be run before this chain. • The New Year Table Initialization (NYTI) program is one that is critical to the success of a roll. Refer to the NYTI run sheet in the *CGI Advantage Financial – Utilities Run Sheets* guide for more information. • Another is the Matching chain job so that any purchase orders that have met matching rules for being received and/or invoiced are closed out with a payment request and not rolled. Refer to the Matching Payment Creation run sheet in the *CGI Advantage Financial – Procurement Run Sheets* guide for more information. • The last is the Systems Assurance 15 program to ensure the open amounts found on accounting lines is accurate. Refer to the System Assurance 15 run sheet in the *CGI Advantage Financial – System Assurance Run Sheets* guide for more information.
Single Instance Required	Yes
Can be restarted?	Yes, please see individual job steps for more detail.
Reports generated	Report is generated for transactions matching selection criteria. An exception report is generated if there is an error during submitting the rolled transactions.

Overview

When a year comes to a close, there are many transactions still un-liquidated with activity that has not yet reached its final state. The Open Activity Roll (OAR) chain job can do two things with that open activity. Most common is rolling the activity from the old budget fiscal year forward into the next year through a transaction modification. When activity is rolled, the program generates the necessary transaction modification, changing the existing BFY on selected accounting lines to the 'Target Year' BFY, which is specified in parameter setup on the Parameters for Roll Process (RLPA) page. An optional part of that modification would be COA changes made on selected accounting lines that match records of a specified COA Crosswalk (COAX) ID.

Another action of the OAR chain is to accrue open activity. Such activity would be an encumbrance where it is known the goods or services have been received but not yet paid for. OAR will modify such accounting lines to close them and create a new accounting line (tied to the former by the Related Accounting Line field) where the event type accrues an amount equal to what was open on the former accounting line.

The Open Activity and Budget Roll (OABR) chain job also rolls the accounting activity but combines that functionality with the rolling of budget authority to cover the accounting activity. It

is used when budget authority is rolled along with the accounting activity. If that applies, then that run sheet should be reviewed.

When rolling (not accruing) the OAR chain does not roll budget authority so budget availability is increased in the older BFY and decreased in the new BFY. Proper BFY Staging setup will prevent that increased authority in the old BFY from being used for other than allowed activity.

The Open Activity Roll chain job can roll several types of activity. What activity is a function of specifying one or more event types on RLPA. The Fund (FUND) page does contain several override controls for use when RLPA selection is not based on fund codes.

Accounting Activity	Examples
Pre-Encumbrances	• Event types with a posting code defined to update the Pre Encumbrances (12) budget amount. PR02, PR03, and ST03 are three such delivered event types.
Encumbrances	• Event types with a posting code defined to update the Encumbrances (13) budget amount. PR05, PR06, and ST04 are three such delivered event types.
Receivables	• Event types with a posting code defined to update the Billed Earned Revenue (22), Unbilled Earned Revenue (23), or Billed Unearned/Deferred Revenue (25) budget amount. AR01, AR06, and AR10 are three such delivered event types.
Other Activity	• Event types with a posting code to update either a Revenue Budget or an Expense Budget that are not defined above but whose associated Bucket ID is listed on the Allowed Open Activity Roll Amounts (ALW_ROLL_BKTS) Application Parameter (APPCTRL) will be allowed on RLPA.
Other Items	• Event types with a posting code defined to update no budget amount, while not also defined to make a Disbursement Request update. PR01, PR07, AR20, AR50, AR52, and AR54 are six such delivered event types. • Event type without posting codes. PR08, ST01, and ST02 are three such delivered event types.

Note – The system looks only at posting pair A for an event type to see what type of accounting activity is booked.

Such activity recorded on the Requisition (RQ), Purchase Order (PO), Accounting Based Spending (ABS), Receivable (RE), Accrued Receivable (ARE), Stock Requisition (SRQ), and Travel (TRVL) transaction types can be selected based on the activity having a Close Action of *Roll* on the Parameters for Roll Process (RLPA) page. If the Allow Fund Action flag for the type of accounting activity is checked on the RLPA Parameter ID used, then the Close Action at the fund code level will be read, which is found on Fund (FUND). If the FUND option is other than *Roll* then no open activity matching RLPA selection criteria will be selected for that fund code.

Further selection is based on the Roll Minimum of the activity. If an accounting line with the open activity equals or exceeds that minimum, it will be selected. The same override is available for the minimum on the FUND table as well. When the header and not the accounting line should be the point of comparison between the Open Amount and Roll Minimum, the Transaction/Line field for each action should be set to *Transaction.*

Notes:

1. If RLPA selection criteria can select a subset of open accounting lines on a transaction, then the Transaction/Line field should be Line to avoid an incorrect comparison of the header Open Amount to the Roll Minimum.
2. Travel Authorization (TRAUTH) is the only Travel Transaction Sub Type within the Travel (TRVL) Transaction Type that can be rolled.

Each instance of the chain is associated with a single record on that table that provides selection criteria as well as some of the date information to be placed on the transaction modifications. Other date information will come from the BACKOUT APD and BACKOUT FY parameters on the Application Parameters (APPCTRL) table.

At a high level, the Open Activity Roll chain selects an open accounting line to roll belonging to an individual BFY and not multi-year lines with 9999, creates a modification draft with changes to the accounting line to include the BFY and any COA values cross-walked, and submits that modification. The budget line for the last BFY is decreased if the rolled activity was budgeted. All accounting impacts from the open amount are removed from the prior fiscal year. If a matching budget line exists in the new year, it will be updated. If a budget line does not exist in the new year, the modification will fail if that budget structure is required unless the Auto Generate option is selected for the new BFY on the Required Budget (REQBUD) table. All budget control errors will be enforced in the roll so any line auto generated needs to have controls that are not set to reject but something less severe.

An alternate method of OAR exists where the chain is run with an RLPA ID that has the Reorg flag checked. This method does a mid-year reorganization of COA according to the required COAX ID specified into the chain. What does not occur is a changing of the BFY value.

The chain process is comprised of the following batch jobs below that are given with very brief summaries. Parameters have to be entered at job #1 only. More details can be found later in the individual job steps.

Job 1 **Roll Update Preprocess**

Batch parameter validation is performed. Selection of records from either the Detail Pre Selection table or directly from the transaction catalogs is next. Finally, when the job is run in update mode only, the workload table is loaded with selected records to guide later processing.

Job 2 **Create Round 1**

The workload table is reviewed for activity that belongs in round 1. That activity is grouped according to parameters that limit the size of work to be loaded. These groups are then logged in a facilitator table. An Open Activity Roll Update job is then spawned for each unit of work. When those spawned jobs have finished, the create job will finish with a report of those transaction modifications.

Job 3 **Submit Round 1**

The workload table is reviewed for activity that belongs in round 1. That activity is grouped according to parameters that limit the size of work to be submitted. These groups are then logged in a facilitator table. A System Maintenance Utility job is then spawned for each unit of work. An extra is spawned if there was a transaction that failed to submit previously as a cleanup step. If there is no round 1 activity, then there is no spawning. When those spawned jobs have finished, the submit job will finish.

Job 4 **Exception Report for Round 1**

A report is compiled to list those transactions that failed to submit from the one or more spawned SMU jobs.

Job 5 **Create Round 2**

The workload table is reviewed for activity that belongs in round 2. That activity is grouped according to parameters that limit the size of work to be loaded. These groups are then logged in a facilitator table. An Open Activity Roll Update job is then spawned for each unit of work. When those spawned jobs have finished, the create job will finish with a report of those transaction modifications.

Job 6 Submit Round 2

The workload table is reviewed for activity that belongs in round 2. That activity is grouped according to parameters that limit the size of work to be submitted. These groups are then logged in a facilitator table. A System Maintenance Utility job is then spawned for each unit of work. An extra is spawned if there was a transaction that failed to submit previously as a cleanup step. If there is no round 1 activity, then there is no spawning. When those spawned jobs have finished, the submit job will finish.

Job 7 Exception Report for Round 2

A report is compiled to list those transactions that failed to submit from the one or more spawned Open Activity Roll batch jobs.

If using an RLPA ID with a Run Mode of *Pre Selection*, only job #1 of the chain is needed. All others should be disabled. Please remember to check all of the Disable flags and hit save to fully disable all other jobs in the chain. If not, job #2 will fail, but that is really an acceptable outcome as there is nothing to load. If using an RLPA ID with a Run Mode of *Update*, usually the whole chain should be run, but there are exceptions when a chain fails midstream. Problem Resolution sections provide details in that case. If using an RLPA ID with a Run Mode of *Report*, do not use the chain job. There is a job in the Reports-GA folder of General Accounting called Open Actv Roll that should be used. If the chain is used, it will fail on the first job step with an error stating that the Run Mode was incorrect for the chain.

Pre-Processing Tasks:

1. Sites have the ability to specify the Accounting Period and Fiscal Year to be recorded on activity being backed out of the prior BFY. The APPCTRL parameters for BACKOUT_APD and BACKOUT_FY must be set if the roll is to happen after the beginning of the new fiscal year. If the next year has not started and the Transaction Record Date parameter on RLPA is blank or not dated to the next fiscal year, the system will default the current fiscal year and accounting period as normal. Please remember to change the BACKOUT FY each year as the BACKOUT APD will not likely change.
2. For the target year posting lines, the Modification Fiscal Year and Modification Accounting Period on RLPA are used. If not specified, the job will use the Transaction Record Date on RLPA to determine these values. If the Transaction Record Date value is blank, the job will use the current system date to determine these values. If rolling into a fiscal year that has not yet started (rolling on the last day of the prior year), these values must be set.
3. Whether rolling before the end of a year or after the new year has started, sites must ensure the proper Transaction Control (DCTRL) settings for each transaction code are set to allow either future or past transaction date, accounting period, and fiscal year.
4. For maximum performance, the COA Crosswalk table could be placed in Service Data Object (SDO) cache. This must be done for all Versata Logic Servers (VLS) that are pointing to the production database, followed by a bounce of each. The table must be removed from SDO cache and each VLS bounced after the processing is complete. This action is helpful if many combinations are being cross-walked.
5. For maximum performance when specifying RLPA selection parameters, a non-unique index could be added to the DOC_ACTG table.

6. Sites should try to select only activity that will be in round 1 or 2. Mixing the two is possible, but can become confusing if the chain job stops processing.
7. If a site is to control rolling the same for all funds, then they should ensure that all of the override flags on RLPA are unchecked to prevent Fund lookups that will return the same value or worse, different values that are not intended.
8. Sites should run the Open Activity Roll report, setting up the Run Mode parameter to report only for the RLPA ID. The report provides an activity count to gauge run times by displaying the open accounting lines that will be rolled. One of the fields on the report also shows if a pending version of any final transactions to be rolled exists. Final transactions with Pending versions are skipped by OAR. The Open Activity Roll report has a setting that will generate a text file (*.txt) of all transactions in a Pending state, which can then be input to a SMU job to reject all pending transactions back to Draft. When this is done before a roll, all transactions will be rolled and the pending ones will not be skipped.
9. For sites where the selection criteria on RLPA is not sufficient, sites should run with the Pre selection mode by setting up the Run Mode parameter to Pre selection and run only the first job step of the OABR process. Using these tables, a user can decide which transactions or separate accounting lines to roll. When the OABR process is then run in update mode and the Pre-selection table that was created during the pre-selection mode is to be used as input, sites should remember to set the USE_PRETABLE batch parameter to True or else the transaction catalogs will be used.
10. If the COA Crosswalk (COAX) feature is to be used, sites should setup all entries before rolling. Transaction codes being cross-walked should be setup on Transaction Control (DCTRL) with the Change Closed Allowed flag checked.
11. Negative and zero-dollar accounting lines that are open will not be rolled. They will have to be dealt with manually.
12. Accounting lines with the multi budget year of 9999 will not be rolled.
13. To ensure fewer transaction modification failures, sites must ensure the transaction codes being rolled are established on Transaction Control (DCTRL) with the following flags checked: Approval Override Allowed and Inactive COA Codes Allowed.
14. Deactivated transactions should be reactivated before rolling, as they will cause the program to fail. If reactivation is not appropriate so that they can be rolled, then they should be reactivated and then cancelled. The reason is that a deactivated transaction cannot be modified.
15. This batch program produces transactions which are subject to the line limit functionality constraints defined on the Transaction Component Requirements (DCREQ) in the Administration application. Sites must ensure that limits have not been set to lower limits than existing transactions being rolled contain.

Common Run Instructions:

1. Run first in a test environment that is a recent copy of production to shake out setup problems and other system configurations that can cause modifications to fail.
2. Establish parameter ID on the RLPA table. There is the option of setting this up when scheduling the chain job by using the Setup Custom Parameters link found with job step 1. However, as sites that schedule the chain and those that determine rolling rules are often different, establishing the table records outside of the chain job is recommended. This step often requires only copying the prior year's record or making changes to that record, updating the Closing BFY, Modification Fiscal Year, Modification Accounting Period, and Transaction Record Date on RLPA.

3. Double check the controls on FUND are correct for the FY value equal to the Closing BFY on RLPA.
4. Run the Open Activity Roll Report with the RLPA parameter that will be used for the actual roll (making sure to adjust the run mode to Report) as an optional step to ensure activity that is desired is all that will be selected and to get a load estimate.
5. For Pre-Selection mode, disable all the jobs except the first one, since only the first job is required for this mode.
6. A good practice is to query for any remaining open activity that should have rolled. This can be done from a variety of sources: ledgers, journals, and the BBAL Fiscal Year Details (BBALFY) page. If ledgers are to be used, then ensure the Ledger Engine has run to ledgerize all rolling activity.

Major Input

- Transaction Accounting Line Catalog (DOC_ACTG)
- OABR Facilitator (OABR_FACILITATOR)
- OABR Workload (OABR_WRKLD)
- Roll/Lapse Pre Selection Detail (RLPSD / RLLP_PRE_DET)
- Roll Parameter (RLPA / RL_PROC_PARM)

Peripheral Input

- Application Parameters (APPCTRL / IN_APP_CTRL)
- Fund (FUND / R_FUND)
- Posting Code (PSCD / R_PSCD)
- Event Type (ETYP / R_EVNT_TYP)
- Auto Transaction Numbering (ADNT / AUTO_DOC_NO)
- Fiscal Year (FY / R_FY)
- Transaction Control (DCTRL / R_GEN_DOC_CTRL)
- Department (DEPT / R_DEPT)
- Appropriation (APPR / R_APPR)
- Appropriation Type (APTYP / R_APTYP)
- Program (PROG / R_PROG)
- Accounting Period (APD / R_APD)
- Transaction Header (*_DOC_HDR)
- Transaction Vendor (*_DOC_VEND)
- Transaction Commodity (*_DOC_COMM)
- Transaction Accounting (*_DOC_ACTG)
- Budget Structure Level Inquiry (BQ*LV* / BUD_STRU_*_LVL_*)
- COA Crosswalk (COAX / R_COA_CROSSWALK)

*Items above with a * denote that different values may be substituted in place of the *.*

Major Output

- OABR Facilitator (OABR_FACILITATOR)
- OABR Workload (OABR_WRKLD)
- XML file of Budget Transactions (OABR*.xml)
- Modifications transactions (Common & specific transaction catalogs)
- Updated budget lines (BQ*LV* / BUD_STRU_*_LVL_*)
- Updated allotment lines (A LOT_STRU_*_LVL_*)
- Updated Accounting Journal (JACTG / JRNL_ACTG)
- Roll/Lapse Pre Selection Summary (RLPSS / RLLP_PRE_SUM)
- Roll/Lapse Pre Selection Detail (RLPSD / RLLP_PRE_DET)
- Rolled Transaction Report per round
- Rolled Transaction Detailed/Summarized Exception Report per round

Chain / Job Return Code

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

Return Code	Condition
Successful (1)	All of the jobs end successfully.
Warning (4)	One of the jobs in the chain 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 on each job step, but several are presented here at the chain job level because they can occur at multiple points in the chain.

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process was discontinued for some reason (for example, job was killed) the user has an option to restart the job and the process will continue from the point it left off. This restart has

a condition that no roll process must be scheduled between the time frame the job was discontinued and is being restarted. The Restart option is provided for jobs – "Create Round n" and "Submit Round n" where n is 1, or 2 as well as the spawned jobs for those different Create and Submit rounds.

If Create Round 1 or 2 jobs successfully created drafts but the submit for that round encountered some transaction submission exceptions, then the user can view the log for the Submit for the round or the Roll Transaction Detailed/Summarized Exception Report generated by the Exception Report for the Round for the errors. In this example, the Exception Report for the round returns a Return Code of *Non-Fatal*, and further jobs down the chain are discontinued. If the errors were due to data setup, correct the errors first. Then, submit the rejected transactions manually via a new System Maintenance Utility (SMU) job or with the Automatic Transaction Corrections batch job because it is necessary to change COA values that were not caught with COAX setup. Finally, restart the Exception Report for the round job which will return a Return Code of *Successful*, thus allowing the rest of the chain to continue. It is critical to ensure that no parameters are changed on RLPA and no OABR or OAR process has been run between the actual run and restart run, else any *.txt files marked for system use are overwritten and the restart run will not work correctly.

In the case of a system failure during any Create Round or Submit Round, all spawned Create jobs (Open Actv Roll Update + <chain job id>) or spawned Submit jobs (System Maintenance Utility + <chain job id>) will return a Return Code of *Failed* or *System Failure*. The Submit Round or Create Round job will also return a Return Code of *Failed*, and further jobs down the chain are Inactive. In this case, the user should first restart each spawned job (Open Actv Roll Update or System Maintenance Utility) so that the individual job can complete work on the set of transactions it was assigned. Then after each spawned job completes successfully, the user should restart the Create Round or Submit Round job. If there is further work to be done in that Create or Submit Round, the chain will spawn additional Open Actv Roll Update or SMU jobs to complete that round's work. After all the applicable work is completed, the chain will automatically move on to the next job in the chain - Exception Report Round.

Application errors can occur on a roll as a result of options being changed or data being deleted from reference tables during a year. Such errors will cause modifications to fail. The source of these errors must be addressed. If online options can be changed, data put back, or other methods to make the errors go away, then that is the recommended approach. If not, then the severity of the error message must be turned down on the Message (MESG) table. Extreme caution should be taken when changing the severity of an error message as that may cause data integrity problems.

Application errors can occur on a roll as a result of options being changed or data being deleted from reference tables during a year. Such errors will cause modifications to fail. The source of these errors must be addressed. If online options can be changed, data put back, or other methods to make the errors go away, then that is the recommended approach. If not, then the severity of the error message must be turned down on the Message (MESG) table. Extreme caution should be taken when changing the severity of an error message as that may cause data integrity problems.

Open Activity Roll Chain: Roll Update Pre Processor Job

Job Name	Roll Update Preprocessor
Recommended Frequency	See chain job. May execute this single job step in a chain just for pre selection or as part of a larger chain with other job steps.
Single Instance Required	Yes

Can be restarted?	No
Reports generated	No

Overview

This first job step starts with basic editing of parameters and goes on to perform accounting line selection based on Run Mode;

If *Pre Selection,* then the job will look to the transaction catalogs for open accounting lines that match RLPA and optionally FUND criteria. Selected accounting lines not already on RLPSD will be added. Open accounting lines already on RLPA will be updated for any information that has changed since the last update. If the RLPSD record has a Selection Date and the accounting line is still open, that date will be cleared. Any RLPSD records found that are no longer open with a blank Selection Date will be removed.

If *Update*, the job uses the 'Use Pre Selection' batch parameter to determine where to look for open accounting lines. If that parameter is 1 – Use Pre Selection, then selection will be from the Roll Lapse Pre Selection Detail (RLPSD) page for records with an Action = *Roll*, a checked Approved flag, and a blank Selection Date. Even after all this selection, if the accounting line is no longer open, it will not be selected. Be aware that when using RLPSD as input, selection criteria on RLPA and FUND are not read, as they were used when RLPSD was loaded.

When the RLPA parameter ID used has a Run Mode of *Update*, then after the selection of records that work is listed on the OARB Workload (OABR_WRKLD) table with information that will be used by later jobs. This workload table contains information to record each accounting line to be rolled along with what round it will be rolled and what the amount being rolled is. Each record is assigned a sequence number (SEQ_NO) that will be used by the next job in the chain when forming units of work for the modification process.

Finally, the job step creates an OARParms.txt file to pass necessary parameters to later jobs steps in the chain.

Process Steps	Messages
1. Parameter validation	Batch input parameters are listed along with APPCTRL parameters. If any is invalid an error message is issued stating such.
2. Deletion of Roll/Lapse Pre Selection records	Number of summarized records deleted from workload table: ## (this is really the pre selection records but the message states workload table)
3. Selection of records	Messages issued for the number of records selected to load in pre selection table.
4. Loading of Roll/Lapse Pre Selection records	Number of summarized records deleted from workload table: ##
5. OABR Workload update	Number of Workload records loaded for Round 1: ## Number of Workload records loaded for Round 2: ## Accounting lines to be rolled: ##
6. REQBUD update	No messages issued for this step

7. Parameter file creation	No messages issued for this step

Restartability Information

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

Major Input

- Transaction Accounting Line Catalog (DOC_ACTG)
- Roll/Lapse Pre Selection Detail (RLPSD / RLLP_PRE_DET)
- OABR Workload (OABR_WRKLD)
- Roll Parameter (RLPA / RL_PROC_PARM)
- Fund (FUND / R_FUND)
- Posting Code (PSCD / R_PSCD)
- Event Type (ETYP / R_EVNT_TYP)

Peripheral Input

- Application Parameters (APPCTRL / IN_APP_CTRL)
- Auto Transaction Numbering (ADNT / AUTO_DOC_NO)
- Fiscal Year (FY / R_FY)
- Accounting Period (APD / R_APD)
- Transaction Control (DCTRL / R_GEN_DOC_CTRL)
- Department (DEPT / R_DEPT)
- Appropriation (APPR / R_APPR)
- Appropriation Type (APTYP / R_APTYP)
- Program (PROG / R_PROG)

Batch Parameters

For the entire chain process, a user has to enter parameters for Job 1 – Roll Update Preprocess and then on Job 8 - Budget Roll only. The process propagates required parameters down the chained jobs. Some of them are non-overrideable parameters, which will be provided with the Job Setup and need not be specified for each run. Many of these are just meant for system use and should not be changed. Those are noted in the Description column.

Parameter Name	Description	Default Value
AMSPARM	Parameter Location for Roll Update Preprocess Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Parms

CHAIN_PARM_FILE	Common Chain Parameters File (.txt). • For system use and should not be changed.	RollParams.txt
CLIENT_NM	Client name to be used for report headers produced from jobs in the chain.	(blank)
COMMIT_SIZE	Required commit block size for the transaction load and submit job steps. This figure determines the number of transactions to be committed at a time for all jobs in the chain that perform transaction load and submit action. It can be used for performance enhancements depending upon the server memory configuration.	1
EXCP_RPT_TYP	Exception report type for the exception report generated in the chain. 1 = Detailed & 2 = Summarized. If not entered, then defaults to 1.	1
JOB_BLOCK_SIZE	Number of accounting lines per SMU Facilitator Parameter file.	100
LOAD_DATA	How to load the Detail Pre Selection table. 1 = load all records with as approved (checked) or 2 = load all records unapproved.	1
PARALLEL_JOB_COUNT	Number of jobs for SMU Facilitator to keep running.	1
POLLING_TIME	Number of seconds to wait between polling occurrences.	30
PRE_SEL_COMMIT_BLOCK	Commit block size for loading pre selection tables. If not entered then defaulted to 1000. It is the number of records that are loaded into the tables after which a commit is issued. Can be used for performance tuning. Too low of a value can cause frequent database commits, thereby increasing processing time. Too high a value will demand for higher memory. Can be used for a performance enhancement depending upon the server memory configuration.	1000
PRE_SELECT_BLOCK	Select block size for loading the pre selection tables. If not entered then defaults to 1000. It is the number of accounting lines that will be fetched	1000

	as a single block from DOC_ACTG. Too low of a value can cause higher number of database fetches, thereby increasing processing time. Too high of a value will demand higher memory. Can be used for a performance enhancement depending upon the server memory configuration.	
PURGE_DATA	How to purge the Detail Pre Selection. 1 to purge all records, 2 to purge only approved records and 3 to purge closed records only record Parameter is read only for Pre Selection mode. Use parameter 2 or 3 when 'refreshing' or loading additional data to RLPSD. Use parameter 1 for the first run of a new year.	1
RL_PROC_PARM_ID	Parameter ID to identify the RLPA record used. The Setup Custom Parameters link opens the RLPA page for record selection, update, or addition.	(blank)
ROLL_CATALOG_ID	• Batch Job Catalog ID for the Open Activity Roll Update job. For use when employing customized jobs for transaction creation. • For system use and should not be changed.	1238
SMU_CATALOG_ID	Batch Job Catalog ID for the System Maintenance Utility job. For use when employing customized jobs for transaction loading, budget rollup, and submitting. • For system use and should not be changed.	3
SMU_FILE_PREFIX	SMU Facilitator parameter file prefix. For system use and should not be changed.	Roll
USE_PRETABLE	Use Detail Pre-Selection Table (1= Use Pre-Selection table as input, 2= do not use Detail Pre-Selection table but use transaction catalogs.") If pre selection is never used, change the default value of this to 2 on BATSETUP as the parameter is often overlooked.	1

XWALK_PROC_ID	Process Id to pick COA Crosswalk records.	(blank)

If you are running the job in the Financial application, you can click on the Custom Parameter link and select a parameter record. If you are running the job in the Administration application, enter a valid Parameter ID from the Custom Parameter table from the Financial application.

Schedule the Roll Batch process by specifying the Parameter ID as the Batch Parameter ID.

Custom Batch Parameters (RLPA)

Please keep in mind that when using the COA selection fields of Department, Program, Appropriation, and Appropriation Type that these elements may not exist on all event types being selected. If an event type is selected and used without one of these elements used as selection criteria on RLPA, the accounting line will not be selected.

Also, if the selection criteria are not enough to identify what is to be rolled, then the pre selection mode is available to load the Roll/Lapse Pre Selection Summary and Detail tables for manual selection. Reports can be run to take records from the detail table and join with other information to produce a very detailed listing of potential records for selection.

Field	Description	Edits
Parameter ID	Unique Parameter ID	Error issued if no value is entered.
Closing BFY	BFY for selection of record to roll. Please enter as CCYY.	Error issued if the year does not exist on the FY page. Error issued if the field is blank. Error issued if multiple years are entered. In each case, the job does not run successfully and an error is issued in the job log.
Reorg	Indication that a roll should not increment the BFY but only make the COA changes indicated by the COAX ID specified as a batch parameter.	When checked the Roll or Accrual Processing field must be set to Roll.
Mode	CVL with values of: Report Only, Update, and Pre Selection	Error issued if the value is blank. In each case, the job does not run successfully and an error is issued in the log. Error message will be issued when Mode is not pre-selection and purge data parameter set to 2 or 3.
Event Type Selection Criteria	Enter event type(s) for selection of open activity. Commas should separate multiple event types.	Error issued if the event type does not exist on R_EVNT_TYP. Error if the Expense Budget Bucket ID is not 12 (pre

		encumbrance) or 13 (encumbrance) and the Expense Budget Bucket Update flag is checked for either posting code in Posting Pair A of the event type, unless the Expense Budget Bucket ID is listed on the Allowed Open Activity Roll Amounts (ALW_ROLL_BKTS) Application Parameter (APPCTRL). Error if the Revenue Budget Bucket ID is not 22 (billed earned revenue), 23 (unbilled earned revenue), or 25 (billed unearned/deferred revenue) and the Revenue Budget Bucket Update flag is checked, unless the Revenue Budget Bucket ID is listed on the Allowed Open Activity Roll Amounts (ALW_ROLL_BKTS) Application Parameter (APPCTRL). Error is issued if event type entered is marked as Disbursement Request Update on Event Type. Error issued if the field is null. In each case, the job does not run successfully and an error is issued in the log.
Target Event Type	Optional event type to be used in place of the existing event type on modification transactions.	Error issued if the event type does not exist on R_EVNT_TYP. This field is not used in the roll process unless it is populated with an Event Type.
Transaction Codes	Optional transaction code(s) for selection of open activity. Commas should separate multiple transaction codes. This parameter can enter text up to 255 characters in size and is optional.	Error issued if the transaction code does not exist on R_GEN_DOC_CTRL. Error is issued if the Transaction code entered is not of transaction type ABS, RE, ARE, PO, RQ, SRQ, or TRVL. In this case, the job does not run successfully and an error is issued in the log.
Fund Selection Criteria	Leave blank for selection of all funds. Individual funds can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any fund code entered is not valid for the closing BFY as FY on the Fund (R_FUND) table, and job does not run successfully.

Department Selection Criteria	Leave blank for selection of all departments. Individual departments can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any department code entered is not valid on the Department (R_DEPT) table, and job does not run successfully.
Appropriation Selection Criteria	Leave blank for selection of all appropriations. Individual appropriations can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any appropriation code entered is not valid for the closing BFY as FY on the Appropriation (R_APPR) table, and job does not run successfully.
Appropriation Type Selection Criteria	Leave blank for selection of all appropriation types. Individual appropriations can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any appropriation code entered is not valid for the closing BFY as FY on the Appropriation (R_APTYP) table, and job does not run successfully.
Program Selection Criteria	Leave blank for selection of all programs. Individual programs can be listed, separated by commas if multiple. (Blank) means program is not part of the selection criteria. This parameter can enter text up to 255 characters in size.	Error issued if no department code is entered and field is not blank. Error issued if more than one department code is entered and more than one program is entered. Multiple values of each are not allowed as the batch job would not know what combinations to form. Multiple runs are required in this situation. Error issued if any program code entered is not valid on the Program (R_PROG) table for the department. In each case, the job does not run successfully and an error is issued in the log.
Modification Fiscal Year	Optional fiscal year (FY) to be placed on all accounting lines rolled. The value will be used on the posting line recording the increase in the new BFY. Must be entered if rolling before the start of a new FY. Please enter as CCYY. Leave blank to default from Transaction Record Date (see parameter below) if used or the Application Date when the roll is performed.	Error issued if the FY does not exist on the Fiscal Year (R_FY) table. Error issued if multiple periods are entered It can be blank when Run mode is Report only In each case, the job does not run successfully and an error is issued in the log.

Modification Accounting Period	Optional accounting period (APD) to be placed on all accounting lines rolled. The value will be used on the posting line recording the increase in the new BFY. Must be entered if rolling before the start of a new FY. Please enter as seen on the Accounting Period table. Leave blank to default from Transaction Record Date (see parameter below) if used or the Application Date when the roll is performed.	Error issued if the APD does not exist on the Accounting Period (R_APD) table. Error issued if multiple periods are entered Error issued if entered and is '99' or '0'. In each case, the job does not run successfully and an error is issued in the log.
Adjustment Reason	Adjustment reason code from the adjustment reason table to be used on RE transaction modifications.	Error issued if the adjustment reason code does not exist on the Adjustment Reason (R_ADJ_REAS) table. Error issued if the field is null and any of the Event Types entered belong to the DEP, REV, or REF event categories. In each case, the job does not run successfully and an error is issued in the log.
Transaction Record Date	Optional Transaction Record Date for rolled transactions. The date will be used instead of a defaulting of the Application Date, and will supply an FY and APD for the accounting line if the above parameters for the same are not used.	Usually transactions generated are stamped with Date of Record at the time of transaction submit. If the process is run over night and the Begin Day job runs then generated transactions will have different dates of record. Hence to have a consistent date of record on all rolled transactions supply a value to this parameter. If not entered then the parameter is defaulted to the current date at the time of parameter processing by job1. If invalid date is entered an error is logged.
Roll or Accrual Processing	The Roll or Accrual Processing field on the Parameters for Roll Process (RLPA) page drives how the Open Activity Roll process will function. If set to *Roll*, OAR will increment BFY on open lines. If set to *Accrue*, OAR will close out selected lines and create a copied line with the	.

	Accrual Event Type.	
Accrual Event Type	The Accrual Event Type field on the Parameters for Roll Process (RLPA) page will populate the Event Type in this field on each Accounting Line that will be inserted on modification transactions as a result of running the Open Activity Roll process.	N/A
Allow Fund Pre Enc Enc Recv Other Items Close Actions	When set to YES, an individual fund code may supply a different close action.	N/A
Pre Enc Enc Recv Other Items Roll Min	A minimum dollar amount which an open accounting line or transaction must be equal to or greater than for selection. Set to $0.00 if all matching open activity should be rolled.	N/A
Allow Fund Pre Enc Enc Recv Other Items Roll Min	When set to YES, an individual fund code may supply a different minimum.	N/A
Pre Enc Enc Recv Other Items Roll Min Transaction/Line	The setting determines at which level the roll minimum is applied: Transaction or Accounting Line.	N/A

Application Parameters (APPCTRL)

Field	Description
BACKOUT_FY	Backout fiscal year on the posting line(s) that remove open amounts from the Source BFY. Sites must change this each year. If left blank, if a Modification FY is used on RLPA that will be the FY used. If Modification FY is left blank and BACKOUT_FY is left blank, then all posting lines will use the default FY. This would only be correct accounting if rolling receivables into the new BFY before

	the end of the FY.
BACKOUT_APD	Backout accounting period on the posting line(s) that remove open amounts from the Source BFY. Unlike BACKOUT_FY, this parameter is often set to 12 or 13 and left alone. If left blank, if a Modification APD is used on RLPA that will be the APD used. If Modification APD is left blank and BACKOUT_APD is left blank, then all posting lines will use the default APD. This would only be correct accounting if rolling receivables into the new BFY before the end of the FY.

Major Output

- OABR Workload (OABR_WRKLD)
- Parameter file (OARParms.txt)

Job Return Code

Return Code	Condition
Successful (1)	If pre processing runs without any parameter errors. Pre selection ends as successful whether or not records were selected.
Warning (4)	Job does not end with this code as it returns successful even if no records are selected.
Non-Fatal Error (8)	No records were selected for updated mode based on FUND and RLPA.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

- Transaction Department Code (DOC_DEPT_CD)
- Transactions Code (DOC_CD)

- Transaction ID (DOC_ID)
- Transaction Version Number (DOC_VERS_NO)
- Transaction Vendor Line Number (DOC_VEND_LN_NO)
- Transaction Commodity Line Number (DOC_COMM_LN_NO)
- Transaction Accounting Line Number (DOC_ACTG_LN_NO)

Selection Criteria

1. Selection criteria for obtaining Transaction Accounting lines (DOC_ACTG) to be rolled is –where the Accounting Line value below matches the RLPA value:
    a. BFY matches Closing BFY
    b. Event Type matches one of Event Type Selection Criteria
    c. Transaction Code matches one of the optional Transaction Codes
    d. Fund matches one of the optional Fund Selection Criteria
    e. Department (accounting line one and not Transaction Department) matches one of the optional Department Selection Criteria
    f. Appropriation matches on of the optional Appropriation Selection Criteria
    g. Appropriation Type matches on of the optional Appropriation Type Selection Criteria
    h. Program matches one of the optional Program Selection Criteria
    i. Open Amount is > $0.00
    j. Transaction Type is not 'JV'
2. If the respective Fund Override flag is checked, get respective Closing Action and Minimum Roll Amount by lookup to FUND using Closing BFY as FY. Please see the Chain Job overview for more information on matching action to type of open activity.
    a. If that FUND action is Roll, then select.
    b. If that Fund action is other than Roll, do not select.

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If no records are selected, review the selection criteria on the RLPA ID used. Also, review the FUND page to verify that the appropriate Action field has a value of *Roll*.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	The pre processor did not encounter any severe parameter errors. If in pre select mode, then successful whether or not records loaded.	If no records were selected, check the job log for messages as certain parameter errors will not cause the job to fail, but will be logged.	N/A
Warning (4)	Job does not end with	N/A	N/A

	this return code.		
Non-Fatal Error (8)	No records were selected for rolling.	Check RLPA ID settings.	Run in report mode until parameters are correct as it is an easier job to run. Then change RPPA to Update or Pre Selection and run the chain.
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following conditions: 1) Encounters any runtime exceptions and 2) Invalid parameter found 3) text files used as templates could not be found If the job fails because of these, investigate the problem. Start a new chain job once the problem is fixed.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Reschedule a new job after reason for termination is resolved.	
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. Reschedule a new job after reason for termination is resolved.	

Open Activity Roll Chain: Create Round 1 Job

Job Name	Create Round 1
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	Yes – Please see the Problem Resolution section for details

Reports generated	Yes – Roll Transaction Report to list transactions being rolled in round 1

Overview

This job will read the RollParams.txt file and the OABR Workload table to create units of work on the OABR Facilitator table and then spawn one or more Open Actv & Budget Roll jobs that will create the necessary modification drafts.

The Create Round 1 job inserts one or more rows into the OARB Facilitator to define a unit of work from one sequence number to another (numbers defined on the OABR Workload table by the previous job step) based on the JOB_BLOCK_SIZE parameter specified in the first job step. Each row inserted is then assigned to a spawned Open Actv & Budget Update job through the population of the Agent ID with the Job ID of the spawned Open Activity Roll Update job). All facilitator records are tied by a Run Number (RUN_NO), which is the Chain Job ID.

The Create Round 1 job will keep running until all spawned Open Actv & Budget Roll jobs are spawned and completed. When those are done, this job step will end.

Process Steps	Messages
1. Parameter Validation	Batch input parameters are listed. If any is invalid, an error message is issued stating such.
2. Spawning of Open Actv & Budget Update jobs	Job with Id:### sequence number:# completed successfully (listed for each spawned job that completed successfully) Batch job ### failed (listed for each spawned job that failed.
3. Report generation	Reports output folder mapped (followed by pdf and html report information) Rendering report started Rendering report completed

Major Input

- OABR Workload (OABR_WRKLD)
- Parameter file (RollParams.txt)

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Create Round 1 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	• $$AMSROOT$$/Logs
AMSPARM	Parameter Location at Create Round 1 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	• $$AMSROOT$$/Parms

PARM_FILE	Parameter File (.txt) For system use and should not be changed.	• RollParams.txt
ROUND_NO	Round Number For system use and should not be changed.	• 1
RUN_TYPE	Run Type For system use and should not be changed.	• 3 (create)

Major Output

- Open Actv & Budget Update job(s) created
- Roll Transaction Report

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and all spawned jobs ended successfully.
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 • One or more spawned jobs failed (ID of that job is specified in job log) • Run time exceptions for unexpected situations When this job ends as failed, subsequent jobs in the chain will become inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process was discontinued for some reason (for example, job was killed) the user has an option to restart the job and the process will continue from the point it left off. This restart has a condition that no roll process must be scheduled between the time frame the job was discontinued and is being restarted. The Restart option is provided for jobs – "Create Round n" and "Submit Round n" where n is 1, or 2 as well as the spawned jobs for those different Create and Submit rounds.

In the case of a system failure during any Create Round, all spawned submit jobs (Open Actv & Budget Update + <chain job id>) will return a Return Code of *Failed* or *System Failure*. The Create Round or will also return a Return Code of *Failed*, and further jobs down the chain are Inactive. In this case, the user should first restart each spawned job (Open Actv & Budget Update) so that individual job can complete work on the set of transactions it was assigned. Then after each spawned job completes successfully, the user should restart the Create Round job. If there is further work to be done in that Create Round, the chain will spawn additional Open Actv & Budget Update jobs to complete that round's work. After all the applicable work is completed, the chain will automatically move on to the next job in the chain – Submit Round.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and all spawned jobs completed 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 conditions: 1) Encounters any runtime exceptions and 2) Invalid parameter found 3) text files used as templates could not be found 4) Spawned job failed If the job fails because of these, investigate the problem. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity Roll Chain: Submit Round 1 Job

Job Name	Submit Round 1
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	Yes – Please see the Problem Resolution section for details
Reports generated	No

Overview

This job will read the RollParams.txt file and OABR Workload table to create units of work on the OABR Facilitator table and then spawn one or more System Maintenance Utility (SMU) jobs that will attempt to submit the modification drafts previously loaded.

The Submit Round 1 job inserts one or more rows into the OARB Facilitator with a txt file created to log a block of transactions for the spawned SMU job to submit. One final SMU submit is also spawned to catch any transactions that failed on the first submit as a cleanup measure to catch any transactions that failed because another was making an update at the same exact moment.

The Submit Round 1 job will keep running until all spawned SMU jobs are spawned and completed. When those are done, this job step will end.

Process Steps	Messages
1. Parameter Validation	Batch input parameters are listed. If any is invalid, an error message is issued stating such
2. Spawning of SMU jobs	Job with Id:### sequence number:# completed successfully (listed for each spawned job that completed successfully)

	Batch job ### failed (listed for each spawned job that failed.

Major Input

- RollParams.txt file
- OABR Workload (OABR_WRKLD)

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Create Round 1 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	• $$AMSROOT$$/Logs
AMSPARM	Parameter Location for Roll Update Preprocess Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	• $$AMSROOT$$/Parms
PARM_FILE	Parameter File (.txt) For system use and should not be changed.	• RollParams.txt
ROUND_NO	Round Number For system use and should not be changed.	• 1
RUN_TYPE	Run Type For system use and should not be changed.	• 2 (submit)
SUBMIT_FILE_NM	Report File for Round 1 (.txt) For system use and should not be changed.	• RollReportRndOne.txt

Major Output

- Spawned SMU job(s)
- Accepted modification transactions (all common and specific transaction components)
- Accounting Journal (JACTG/JRNL_ACTG)
- Report file (RollReportRndOne.txt)

Job Return Code

Return Code	Condition

Successful (1)	All parameters were valid and all spawned jobs ended successfully.
Warning (4)	• Job does not end with this return code, but spawned SMU jobs may.
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 • Spawned SMU job fails • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process was discontinued for some reason (for example, job was killed) the user has an option to restart the job and the process will continue from the point it left off. This restart has a condition that no roll process must be scheduled between the time frame the job was discontinued and is being restarted. The Restart option is provided for jobs – “Create Round n” and “Submit Round n” where n is 1, or 2 as well as the spawned jobs for those different Create and Submit rounds.

In the case of a system failure during any Submit Round, all spawned submit jobs (System Maintenance Utility + <chain job id>) will return a Return Code of *Failed* or *System Failure*. The Submit Round will also return a Return Code of *Failed*, and further jobs down the chain are Inactive. In this case, the user should first restart each spawned job (System Maintenance Utility) so that individual job can complete work on the set of transactions it was assigned. Then after each spawned job completes successfully, the user should restart the Submit Round job. If there is further work to be done in that Submit Round, the chain will spawn additional SMU jobs to complete that round’s work. After all the applicable work is completed, the chain will automatically move on to the next job in the chain - Exception Report Round.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and all spawned jobs completed successfully.	N/A	N/A
Warning (4)	Job does not end with this return code, but spawned SMU jobs may because of optimistic locking errors.	Restart the SMU job	If the Load Round 1 job will not restart after the SMU job completes, then start a new chain at the Round 1 Exception Report. Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
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 conditions: 1) Encounters any runtime exceptions 2) Invalid parameter found 3) text files used as templates could not be found 4) Spawned job failed If the job fails because of these, investigate the problem. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity Roll Chain: Exception Report Round 1 Job

Job Name	Except Rpt Rnd 1
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Rolled Transaction Exception Report

Overview

This job will read the Report file (RollReportRndOne.txt) created from the submit step and produce a report listing those modifications that failed to submit along with a limited amount of information on the errors issued for that transaction. If no transactions fail, a report is still produced to state "No Transaction Errors". In the case of no errors, the job ends as *Successful*. If any transaction fails to submit to final, then the job step ends as *Non Fatal* and the chain stops. Please see the Problem Resolution section for details about what to do in this case.

Process Steps	Messages
1. Parameter validation	• Batch input parameters are listed. If any is invalid, an error message is issued stating such.
2. Report generation	• Reports output folder mapped (followed by pdf and html report information) • No transaction errors or • Error records processed ## • Rendering report started • Rendering report completed

Major Input

- RollReportRndOne.txt

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Except Rpt Round 1 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	• $$AMSROOT$$/Logs
AMSPARM	Parameter Location at Except Rpt Rnd 1 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	• $$AMSROOT$$/Parms
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) For system use and should not be changed.	• RollParams.txt
SUBMIT_FILE_NM	Report File for Round 1 (.txt) For system use and should not be changed.	• RollReportRndOne.txt

Major Output

- Exception Report

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and report was generated successfully.
Warning (4)	• Job does not end with this return code.
Non-Fatal Error (8)	At least one transaction failed to submit.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

- If an Exception Report lists any failed transactions due to wrong data setup that caused the modifications to fail, then the following steps can be executed to resubmit the failed rolling transactions.
    a. Correct any data setup problems that caused transactions to fail. Often it is just a very limited number of control table settings.
    b. If the number of failed transactions is low, they can be manually submitted. If the number is beyond manual intervention, then an independent SMU job can be run to submit the modifications.
    c. After either method, a new chain can be submitted starting with the Transaction Exception Report to verify all were submitted successfully. It is critical to ensure that no parameters are changed on the Parameter table and no roll process has run in update mode between the actual run and restart run, else any .txt files marked for system use (see parameter list above) are overwritten and the restart run will not work correctly.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and all spawned jobs completed successfully.	N/A	N/A
Warning (4)	Job does not end with this return code.	N/A	N/A
Non-Fatal Error (8)	One or more modifications failed to submit.	Need to complete submit process after the reason for the failure is resolved.	See problem resolution above
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following conditions: 1) Encounters any runtime exceptions 2) Invalid parameter found 3) text files used as templates could not be found 4) Exception report file could not be found. If the job fails because of these, investigate the problem.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

		Need to schedule a new chain starting at this point after reason for the failure is resolved.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Need to schedule a new chain starting at this point after reason for the termination is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. Need to schedule a new chain starting at this point after reason for the failure is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity Roll Chain: Create Round 2 Job

Job Name	Create Round 2
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	Yes – Please see the Problem Resolution section for details
Reports generated	Yes – Roll Transaction Report to list transactions being rolled in round 2

Overview

This job will read the RollParams.txt file and the OABR Workload table to create units of work on the OABR Facilitator table and then spawn one or more Open Actv & Budget Roll jobs that will create the necessary modification drafts.

The Create Round 2 job inserts one or more rows into the OABR Facilitator to define a unit of work from one sequence number to another (numbers defined on the OABR Workload table by the previous job step) based on the JOB_BLOCK_SIZE parameter specified in the first job step. Each row inserted is then assigned to a spawned Open Actv & Budget Update job through the population of the Agent ID with the Job ID of the spawned Open Activity Roll Update job). All facilitator records are tied by a Run Number (RUN_NO), which is the Chain Job ID.

The Create Round 2 job will keep running until all spawned Open Actv & Budget Roll jobs are spawned and completed. When those are done, this job step will end.

Process Steps	Messages
• Parameter	Batch input parameters are listed. If any is invalid, an error message is issued stating such.

Validation	
• Spawning of Open Actv & Budget Update jobs	Job with Id:### sequence number:# completed successfully (listed for each spawned job that completed successfully) Batch job ### failed (listed for each spawned job that failed.
• Report generation	Reports output folder mapped (followed by pdf and html report information) Rendering report started Rendering report completed

Major Input

- OABR Workload (OABR_WRKLD)
- Parameter file (RollParams.txt)

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Create Round 2 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Logs
AMSPARM	Parameter Location at Create Round 2 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Parms
PARM_FILE	Parameter File (.txt) For system use and should not be changed.	RollParams.txt
ROUND_NO	Round Number For system use and should not be changed.	2
RUN_TYPE	Run Type For system use and should not be changed.	3 (create)

Major Output

- Open Actv & Budget Update job(s) created
- Roll Transaction Report

Job Return Code

Return Code	Condition

Successful (1)	All parameters were valid and all spawned jobs ended successfully.
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 • One or more spawned jobs failed (ID of that job is specified in job log) • Run time exceptions for unexpected situations When this job ends as failed, subsequent jobs in the chain will become inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process was discontinued for some reason (for example, job was killed) the user has an option to restart the job and the process will continue from the point it left off. This restart has a condition that no roll process must be scheduled between the time frame the job was discontinued and is being restarted. The Restart option is provided for jobs – “Create Round n” and “Submit Round n” where n is 1, or 2 as well as the spawned jobs for those different Create and Submit rounds.

In the case of a system failure during any Create Round, all spawned submit jobs (Open Actv & Budget Update + <chain job id>) will return a Return Code of *Failed* or *System Failure*. The Create Round or will also return a Return Code of *Failed*, and further jobs down the chain are Inactive. In this case, the user should first restart each spawned job (Open Actv & Budget Update) so that the individual job can complete work on the set of transactions it was assigned. Then after each spawned job completes successfully, the user should restart the Create Round job. If there is further work to be done in that Create Round, the chain will spawn additional Open

Actv & Budget Update jobs to complete that round's work. After all the applicable work is completed, the chain will automatically move on to the next job in the chain – Submit Round.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and all spawned jobs completed 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 conditions: 1) Encounters any runtime exceptions and 2) Invalid parameter found 3) text files used as templates could not be found 4) Spawned job failed If the job fails because of these, investigate the problem. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity Roll Chain: Submit Round 2 Job

Job Name	Submit Round 2
Recommended Frequency	See chain job

Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	No

Overview

This job will read the RollParams.txt file and OABR Workload table to create units of work on the OABR Facilitator table and then spawn one or more System Maintenance Utility (SMU) jobs that will attempt to submit the modification drafts previously loaded.

The Submit Round 2 job inserts one or more rows into the OARB Facilitator with a txt file created to log a block of transactions for the spawned SMU job to submit. One final SMU submit is also spawned to catch any transactions that failed on the first submit as a cleanup measure to catch any transactions that failed because another was making an update at the same exact moment.

The Submit Round 2 job will keep running until all spawned SMU jobs are spawned and completed. When those are done, this job step will end.

Process Steps	Messages
1. Parameter Validation	Batch input parameters are listed. If any is invalid an error message is issued stating such
2. Spawning of SMU jobs	Job with Id:### sequence number:# completed successfully (listed for each spawned job that completed successfully) Batch job ### failed (listed for each spawned job that failed.

Major Input

- RollParams.txt file
- OABR Workload (OABR_WRKLD)

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Create Round 2 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Logs
AMSPARM	Parameter Location for Roll Update Preprocess Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	$$AMSROOT$$/Parms
PARM_FILE	Parameter File (.txt) For system use and should not be changed.	RollParams.txt

ROUND_NO	Round Number For system use and should not be changed.	2
RUN_TYPE	Run Type For system use and should not be changed.	2 (submit)
SUBMIT_FILE_NM	Report File for Round 2 (.txt) For system use and should not be changed.	RollReportRndTwo.txt

Major Output

- Spawned SMU job(s)
- Accepted modification transactions (all common and specific transaction components)
- Accounting Journal (JACTG/JRNL_ACTG)
- Report file (OARBRReportRndTwo.txt)

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and all spawned jobs ended successfully.
Warning (4)	Job does not end with this return code, but spawned SMU jobs may.
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 • Spawned SMU job fails • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If the chain process was discontinued for some reason (for example, job was killed) the user has an option to restart the job and the process will continue from the point it left off. This restart has a condition that no roll process must be scheduled between the time frame the job was discontinued and is being restarted. The Restart option is provided for jobs – "Create Round n" and "Submit Round n" where n is 1, or 2 as well as the spawned jobs for those different Create and Submit rounds.

In the case of a system failure during any Submit Round, all spawned submit jobs (System Maintenance Utility + <chain job id>) will return a Return Code of *Failed* or *System Failure*. The Submit Round or will also return a Return Code of *Failed*, and further jobs down the chain are Inactive. In this case, the user should first restart each spawned job (System Maintenance Utility) so that individual job can complete work on the set of transactions it was assigned. Then after each spawned job completes successfully, the user should restart the Submit Round job. If there is further work to be done in that Submit Round, the chain will spawn additional SMU jobs to complete that round's work. After all the applicable work is completed, the chain will automatically move on to the next job in the chain - Exception Report Round.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and all spawned jobs completed successfully.	N/A	N/A
Warning (4)	Job does not end with this return code, but spawned SMU jobs may because of optimistic locking errors.	Restart the SMU job	If the Load Round 1 job will not restart after the SMU job completes, then start a new chain at the Round 1 Exception Report. Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

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 conditions: 1) Encounters any runtime exceptions 2) Invalid parameter found 3) text files used as templates could not be found 4) Spawned job failed If the job fails because of these, investigate the problem. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. This and any spawned job that failed can be restarted after the problem is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.

Open Activity Roll Chain: Exception Report Round 2 Job

Job Name	Except Rpt Rnd 2
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	Rolled Transaction Exception Report

Overview

This job will read the Report file (RollReportRndTwo.txt) created from the submit step and produce a report listing those modifications that failed to submit along with a limited amount of information on the errors issued for that transaction. If no transactions fail, a report is still produced to state "No Transaction Errors". In the case of no errors, the job ends as *Successful*. If any transaction fails to submit to final, then the job step ends as *Non Fatal* and the chain stops. Please see the Problem Resolution section for details about what to do in this case.

Process Steps	Messages
1. Parameter validation	• Batch input parameters are listed. If any is invalid, an error message is issued stating such.
2. Report generation	• Reports output folder mapped (followed by pdf and html report information) • No transaction errors or • Error records processed ## • Rendering report started • Rendering report completed

Major Input

- RollReportRndTwo.txt

Batch Parameters

Parameters entered in the Preprocess job step are passed into this job step for many parameters. The few unique to the job step are as follows:

Parameter Name	Description	Default Value
AMSLOGS	Log Location at Except Rpt Round 2 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	• $$AMSROOT$$/Logs
AMSPARM	Parameter Location at Except Rpt Rnd 2 Job (**** Refer to Note:** Assumptions for SWBP on page no. 6)	• $$AMSROOT$$/Parms
CHAIN_PARM_FILE	Common Chain Parameters File (.txt) For system use and should not be changed.	• RollParams.txt
SUBMIT_FILE_NM	Report File for Round 2 (.txt) For system use and should not be changed.	• RollReportRndTwo.txt

Major Output

- Exception Report

Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid and report was generated successfully
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 • Run time exceptions for unexpected situations When this job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated subsequent jobs in the chain will be set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain will be set to inactive.

Sort Criteria

None

Selection Criteria

None

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

- If an Exception Report lists any failed transactions due to wrong data setup that caused the modifications to fail, then the following steps can be executed to resubmit the failed rolling transactions.

  a. Correct any data setup problems that caused transactions to fail. Often it is just a very limited number of control table settings.

  b. If the number of failed transactions is low, they can be manually submitted. If the number is beyond manual intervention, then an independent SMU job can be run to submit the modifications.

  c. After either method, a new chain can be submitted starting with the Transaction Exception Report to verify all were submitted successfully. It is critical to ensure that no parameters are changed on the Parameter table and no roll process has run in

update mode between the actual run and restart run; otherwise, any .txt files marked for system use (see parameter list above) are overwritten and the restart run will not work correctly.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters validated successfully and all spawned jobs completed successfully.	N/A	N/A
Warning (4)	Job does not end with this return code.	N/A	N/A
Non-Fatal Error (8)	One or more modifications failed to submit.	Need to complete submit process after the reason for the failure is resolved.	See problem resolution above.
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following conditions: 1) Encounters any runtime exceptions 2) Invalid parameter found 3) text files used as templates could not be found 4) Exception report file could not be found. If the job fails because of these, investigate the problem. Need to schedule a new chain starting at this point after reason for the failure is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Need to schedule a new chain starting at this point after reason for the termination is resolved.	Ensure no OAR chain has started with the Pre Process job step (#1) in the interim. If so, this chain is effectively over.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. Need to schedule a new	Ensure no OAR chain has started with the Pre

		chain starting at this point after reason for the failure is resolved.	Process job step (#1) in the interim. If so, this chain is effectively over.

2.1.27 Partial Transaction Ledger Engine

Job Name	Partial Transaction Ledger Engine
Recommended Frequency	On demand
Single Instance Required	Yes
Can be restarted?	No
Reports generated	None

Overview

This job handles such a rare scenario that, ideally, it should never have to be used. It is used when an instance of the Ledger Engine failed to write out all the journal records for a transaction to all the ledgers built from that journal. If this partial ledgerization scenario takes place, it was most likely due to some unusual situation. The functional and technical recommendation for this outcome is to rebuild the ledger(s) affected, so as to ensure that Journal Record(s) are neither missed nor double counted. In the case where rebuilding of the ledgers is not feasible then this Partial Transaction Ledger Engine job can be utilized. Before using this job, thorough research should be performed to identify those journal records that have not been ledgerized to which ledgers. This approach is strongly recommended as there are a very limited number of assurances in this job to ensure that the specific journal record to be ledgerized has not already been ledgerized.

As mentioned above, this job is for a transaction where a partial number of journal records didn't post to all necessary ledgers. If an entire transaction didn't post to any ledgers, the process is to use the Ledger Engine in 'Failed Work Mode'. Please see the Ledger Engine runsheet for more information on that process.

The Partial Transaction Ledger Engine job first validates batch parameters supplied. When found to be valid, the process then looks for that specified record in the defined Source Journal of the input ledger parameter. When found, that journal record is ledgerized with the normal Ledger Engine logic. After successful ledgerization, the job then updates the Journal Log (JLOG) page. Any existing JLOG records with the specified journal record has that record removed by splitting that JLOG record into two records and placing that journal record by itself into a new JLOG record. Thus, there are three JLOG records where before there was one.

Shown below is a small set of sample data to demonstrate how this batch process works.

Suppose following entries were logged in JRNL_LOG:

UNID	PROC ID	ST FL	BGN DOC UNID	BGN DOC LN ID	END DOC UNID	END DOC LN ID	DSCR
1	LDGRPOST	1 (Intended)	1	1	6	2	Ledgerizing journal 1 (Orig Sel Instr)

Suppose JRNL_ACTG has following records:

REC NO	DOC UNID	DOC LN ID
1	1	1
2	1	2
3	2	1

4	2	2
5	3	1
6	3	2
7	4	1
8	4	2
9	5	1
10	5	2
11	5	3
12	5	4
13	6	1
14	6	2

This sample scenario had only line 1 of transaction #4 to successfully ledgerize. Line 2 of transaction #4 was only written to 1 of the necessary ledgers (IDs 12, 13, 14, 15, and 16). After careful research identified that journal record #2 for transaction #4 didn't write to ledgers 13 to 16, the user would run the Partial Transaction Ledger Engine 4 times as follows:

Run 1
Ledger 13
Transaction 4
Line 2

Run 2
Ledger 14
Transaction 4
Line 2

Run 3
Ledger 15
Transaction 4
Line 2

Run 4
Ledger 16
Transaction 4
Line 2

The following shows the updated JLOG record (1st row) from the original starting point to just before the point of failure, the new JLOG record from just after the point of failure to the end of the original block (2nd row), and the new JLGO records from the Partial Transaction Ledger Engine.

UNID	PROC ID	ST FL	BGN DOC UNID	BGN DOC LN ID	END DOC UNID	END DOC LN ID	DSCR
1	LDGRPOST	2 (Final)	1	1	4	1	Ledgerizing journal 1 (Orig Sel Instr)
2	LDGRPOST	2 (Final)	5	1	6	2	Ledgerizing journal 1 (Orig Sel Instr)

3	LDGRPOST	2 (Final)	4	2	4	2	Processed by PartialLedgerizingEngine

Processing steps:
As the job progresses, messages will be issued to the Job Log: parameter validation and ledgerization.

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 will be displayed in the log. • Batch Parameter validation completed.
2. Ledgerization	• Ledgerizing journal started. • Journal Record is ledgerized successfully. • Run ended.

Major Input

- Journal/Ledger Control Detail (R_JRNL_LDGR_CTRL) - provides summarization rules describing which ledgers are summarized from which journals and how are they summarized. These Journal/Ledger Control records specify the names of the Source Journals. These are typically named *JRNL_xxxx*, where *xxxx* is a descriptive identifier (for example, JRNL_ACTG is the Accounting Journal and JRNL_CA is the Cost Accounting Journal).
- Potential input journals (in baseline CGI Advantage Financial) include:
- Accounting Journal (JRNL_ACTG)
- Cost Accounting Journal (JRNL_CA)
- Cash Journal (JRNL_CASH)
- 1099 Journal (JRNL_1099)
- Internal Journal (JRNL_INT)
- Cross-Year (Budget FY ≠ FY) Journal (JRNL_BFYNOTFY)
- Journal Log (JRNL_LOG) provides information for record selection.

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
Partial Transaction Ldgrzied Ldgr ID	Ledger to Post To Required. Identifies the single Ledger ID to be processed. Valid values are valid active Ledgers Identifiers (numbers) from the Journal/Ledger Control (JLCTRL) page.	(blank)
Partial Transaction Ldgrzied Transaction UNID	Transaction to Post Required. Identifies the single transaction to ledgerize by DOC_UNID value assigned in the source journal for the input ledger.	(blank)
Partial Transaction Ldgrzied Transaction LNID	Transaction Line to Post Required. Identifies the single transaction line (journal record) to ledgerize by DOC_LN_ID.	(blank)

Major Output

- Primary output is a ledger table (LDGR_###) where a new record is added and existing record updated with a new amount.
- Journal Ledger Cross Reference (JRNL_LDGR_XREF) updated ledgerized journal record.
- Journal Log (JRNL_LOG) is updated listing all blocks of work and the resulting status of each. Those records are identified with the Process ID's of LDGRPOST.

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)	Journal record found and successfully ledgerized.
Warning (4)	This return code will be issued when no journal record found for specified Transaction UNID and Line Number.
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. • Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure	This return code will be issued when the job is terminated because of

(20)	database server or network issues.

Sort Criteria

None

Selection Criteria

The job selects only the journal record identified by values provided in Partial Transaction Ldgrzied Transaction UNID and Partial Transaction Ldgrzied Transaction LNID parameters and ledgerizes it to the ledger table associated with the Ledger Id supplied in the Partial Transaction Ldgrzied Ldgr Id parameter.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the parameters are validated successfully	N/A	N/A
Warning (4)	Job ended with a Warning because there is no matching journal record for ledgerization. **Sample Message:** No unledgerized Journal Records found for Ledger # Journal JRNL_ACTG(say) Transaction UNID # Line Number #	Check to ensure the correct DOC UNID and DOC_LN_ID are being used for the journal record that needs to be ledgerized.	N/A
Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	Job failed due to Fatal conditions	In this step, the job can fail under the following two conditions. 1) Parameter errors 2) 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. Then 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. Then schedule a new job.	N/A

2.1.28 Payroll Transaction Generator

Chain or Job Name	Payroll Transaction Generator
Recommended Frequency	Periodic basis
Single Instance Required	Yes
Can be restarted?	Yes, see individual job steps for more information
Reports generated	Yes, Payroll Exception Report

Overview

The Payroll Transaction Generator chain has the following job steps. Even though these job steps in the chain can be run individually by disabling other jobs, it is recommended to always run the entire chain.

1. Generate Payroll Transactions
2. Load And Submit Transactions
3. Generate Payroll Exception Report

The Payroll Transaction Generator chain job is a group of job steps that work together to build Payroll transactions inside of Advantage Financial (FIN), which are based on data from a payroll run in Advantage Human Resource Management (HRM); upload them to FIN; mark records as processed in HRM; submit the transactions; and finally report on any transactions that rejected.

The FIN chain has to be independently started as there is no trigger between HRM and FIN to know when HRM has finished building the transaction data. In fact, there may be multiple runs of HRM data for FIN processing at a given time. Through the submitting of the transactions generated in FIN, records are created on Payroll Accounting Reconciliation (PYRLAR) that are the basis of a subsequent HRM job, Payroll Reconciliation. If a flag [Payroll Accounting Reconciliation Update] is checked on Payroll Event Type Defaults (PYRLETD) for the combination of transaction code and event type, then PYRLAR is updated.

Record Selection

HRM records selected for processing will depend upon batch parameter setup, however, a theoretical run would show the acquisition of all records from the HRM table R_PYRL_HDR where the GTN Run Number is the value supplied to the chain job and all Payroll Transaction Sub-Types except for PREXC and the record had not already been processed by the Payroll Transaction Generator job. The PREXC must have a dedicated run and when it is to be processed the GTN Run Number is optional.

Major Input

- R_PYRL_HDR (inside of HRM)
- R_PYRL_VEND (inside of HRM)
- R_PYRL_ACTG (inside of HRM)
- R_FY
- R_FY_DEPT

- R_FY_FUND
- R_APD
- R_APD_DEPT
- R_APD_FUND

Major Output

- PYRL transactions in the XML File
- Update to the FIN_PROCESSED_FL field on R_PYRL_HDR to signify a transaction has been processed
- Chain Job Parameter File
- Failed transaction report
- Payroll Accounting Reconciliation (PYRLAR) – updated by processed transactions and not the chain job

Chain Job Return Codes

The following table shows the potential return codes for the Payroll Transaction Generator chain job. The Chain job will end with the highest return code across all of the jobs.

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

Problem Resolution

If any of the jobs in the chain failed due to application errors it is advisable to restart the job (when possible) after correcting the errors instead of rescheduling the job. Restarting the job will reduce the processing time since the job will resume from where it was last committed and will select only the unprocessed records. When restarting is not an option, the entire chain can be run again.

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

Payroll Transaction Generator: Generate Payroll Transactions

Job Name	Generate Payroll Transactions
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	No
Reports Generated	No

Overview

This is the first step of the chain job, which is responsible for acquiring the set of Payroll transactions to be generated from the HRM R_PYRL_HDR, VEND and ACTG tables, based upon the successful validation of the batch parameters, and then generate the XML file. It is important to understand that the logic for creating the PYRL transactions resides inside of HRM. HRM creates the entries inside of the R_PYRL_XXX tables and that is the logic for creating the vast majority of transaction data. The logic in FIN, except for the Soft Closed FY and APD Logic, is to create the transactions as HRM desires. Of course, FIN may add information when processing transactions with inferences of additional data based on what HRM supplied.

This job step consists of the following basic steps:

1. **Parameter Validation:** In this step, the process verifies the parameters. If the parameter validation is successful, the job proceeds to the next step. Otherwise, the job ends with a return code of *Failed,* listing the invalid parameter in the job log.
2. **Cross Field Validations:** Once the parameters have been validated successfully there is an interconnection of some of the Batch Parameters that will be validated here. The main interconnection is that if the GTN Run Number parameter is empty then the Type of Payroll Transactions parameter must be *Adjustment* or '*A*', which signifies that it should only process Transaction Subtype PREXC.
3. **Write Out Parameters to a Chain Parameter File:** This step passes certain parameters in a file to the last step of the chain job to build the report of any failed transactions. The name of this file is in the Chain Parameter File Name parameter.
4. **Initialize Additional Values:** This step determines whether initialize values should be used to determine the soft closed status of FY and APD values when building accounting lines and possibly PRLID vendor lines.
5. **Initialize Transaction XML:** This step performs the necessary steps to create the output file used to write all of the Payroll transactions to be uploaded and submitted using an XML file.
6. **Build Payroll Transactions:** This is the main bulk of logic which executes a query into HRM's Database table R_PYRL_HDR using the values supplied as batch parameters. It then walks through each R_PYRL_HDR building the Payroll transaction inside of the output XML file. It will, for each R_PYRL_HDR record, get all R_PYRL_VEND records and R_PYRL_ACTG records linked to the Header record to write out the Payroll transaction. Once all of the Payroll Headers have been successfully processed it will update the FIN_PROCESSED_FL field in R_PYRL_HDR, marking those headers as processed.

7. **Save XML File:** The final step will be to save the XML File that holds all of the Payroll Transactions created.

The main step – Build Payroll Transactions - will acquire all of the R_PYRL_HDR records in HRM that match the supplied parameters. Then with each R_PYRL_HDR record it will write the R_PYRL_HDR record out to the XML file. The next step will be to acquire all of the R_PYRL_VEND records mapped to the R_PYRL_HDR record. For each R_PYRL_VEND record it will write it out to the XML file and will then see if this vendor line is from Transaction Sub-type PRLID and, if so, soft close logic is implemented. If so, it will then calculate the FY or Accounting Period to be set on the vendor line; if soft close logic is not desired then no FY or Accounting Period will be written out. Once the vendor line has been written to the XML file it will then find all of the accounting lines for that vendor line inside of R_PYRL_ACTG. It will then write out each accounting line to the XML file and again see if Soft Closed Logic is needed, in which case it will acquire the necessary FY or Accounting Period and write those out as needed, calculating the value(s) for each accounting line.

Once all of the header records have been completed the xml file is completed and then an update is performed on R_PYRL_HDR to mark which Headers were written to an XML file, so that during the next run the same transaction(s) will not be created again.

Process Step	Messages
1. Parameter Validation	• Begin generation of PYRL transactions • Parameter validation is going to begin • Validating Batch Parameters • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, then the invalid value is displayed in the log. • If Parameter validation failed the failed parameters are listed and the following message is written out: “Parameter validation failed.” • Parameter validation successfully completed
2. Cross Field Validations	• Cross field validation begins • If the GTN Run Number is empty and the TYPE_PYRL_DOC is not ‘A’, then following message is issued: “GTN_RUN_NO is required if TYPE_PYRL_DOC is P, L or C” • If cross field validations failed, then the following message is issued: “Cross field validation(s) failed.” • Cross field validation(s) successfully completed
3. Write Out Parameters to a Chain Parameter File	• Going to write out common parameters to Parm File. • Completed writing out common parameters to Parm File.

4. Initialize Additional Values	• If failure encountered trying to capture important information to assist in deciding if Soft Closed logic is needed, then the following is written out “Setting additional values failed”
5. Initialize Transaction XML	• Going to initialize the Output XML File GenPYRLDocs.xml. • Completed initializing the Output XML File GenPYRLDocs.xml.
6. Build Payroll Transactions	• Time to begin building PYRL Transactions. • Start PYRL Header record selection • End PYRL Header record selection • Start PYRL Header record processing • If no records are found in selection then the following message issued: “Query to find PYRL Header returned was null which appears to be a problem.” • Periodically a commit is performed to assist in clearing out memory. If a problem is encountered during a commit the following message is issued: “Failed to commit the PYRL Header transaction.” [The prior transaction code, dept code, transaction id and transaction version number will be listed in the message]. • A progression message is issued based upon the batch parameter value supplied. The periodic message is: “PYRL Header record(s) processed (the number of Headers processed) [the current transaction header information being processed].” • If there is a problem building a Payroll Transaction the following error message is written to log: “Problem creating transaction for [supplying the specific transaction information].” • If there is a problem building the Header inside of the output XML file the following message is written: “Problem creating Header for [supplying transaction information].” • If there is a problem getting some critical information for writing out the header the following message is written to the log: “Problem creating critical data for Header with [supplying transaction information].”

	• If there is a problem trying to build Vendor or Accounting Line(s), then the following message is written to the log: “Problem creating Vendor or Accounting Lines for [supplying transaction information].” • If a query returned an empty result set, then the following message is written to the log: “Problem getting Vendor Lines for [supplying transaction information].” • If the Vendor Line Number of the record is empty, then the following message is written: “Problem getting Vendor Line Number for [supplying transaction information].” • If there is a problem getting some critical information for writing out the header, then the following message is written to the log: “Problem creating critical data for VL with [supplying transaction information].” • If soft closed logic is desired for APD or FY on Vendor Line and there was a problem calculating FY or Period, then the following message is written to the log: “Problem calculating FY or Period for Vendor with [supplying transaction information].” • If there is a problem writing the VL out to the XML file, then the following message is written to the log: “Problem creating Vendor Line for [supplying transaction information including VL Number].” • If there is a problem writing the AL(s) out to XML file, then the following message is written to the log: “Problem creating Vendor Line for [supplying transaction information including VL Number].” • If the query returned an empty result set, then the following message is written to the log: “Problem getting Accounting Lines for [supplying transaction information including VL Line Number].” • If AL Line Number is empty, then the following message is written to the log: “Problem getting Accounting Line Number for [supplying transaction information including VL Line Number].”

	• If there is a problem writing the AL out to the XML File, then the following message is written to the log: Problem creating Accounting Line for [supplying transaction information including VL Line Number]. • If there is a problem getting some critical information for writing out the Accounting Line, then the following message is written to the log: "Problem creating critical data for Accounting with [supplying transaction information]." • If soft closed logic is desired for APD or FY on the Accounting Line and there was a problem calculating FY or Period, then the following message is written to the log: "Problem calculating FY or Period for Accounting with [supplying transaction information]." • If no records returned for the Vendor, then the following message is written to the log: "No Accounting Lines found to process for [supplying transaction information including VL Line Number]." • If no records are returned for the Header, then the following message is written to the log: "No Vendor Lines found to process for [supplying transaction information]." • Once all headers are processed or a problem is encountered, the following message is written to log: "End PYRL Header record processing." • If a problem is encountered committing after all work is performed, then the following message written to the log: "Failed to commit the records after PYRL Header work." • Before ending logic, the following message is written to log: [This will be the number of header records processed] "PYRL Header record(s) processed." • If a problem is encountered building transactions, then the following message is written to the log: "Problem building PYRL transactions." • Completed building PYRL Transactions.

7. Save XML File	• Total [Number of Header records processed] record(s) are processed • End generation of PYRL transactions

Restartability Information

This job step is not restartable.

Major Input (Tables)

- R_PYRL_HDR (inside of HRM)
- R_PYRL_VEND (inside of HRM)
- R_PYRL_ACTG (inside of HRM)
- R_FY
- R_FY_DEPT
- R_FY_FUND
- R_APD
- R_APD_DEPT
- R_APD_FUND

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
Export Directory (AMSEXPORT)	(Required) Export Location for Payroll Job	$$AMSROOT$$/ExportImport
Parameter File Location (AMSPARM)	(Required) Parameter Location for Payroll Job	$$AMSROOT$$/Parms
Chain Parameter File Name (CHAIN_PARM_FILE)	(Required) Common Chain Parameters File (.txt)	PyrlActgChain.txt
Client Name for Report (CLIENT_NM)	(Optional) Client Name for Payroll Exception Report heading	No Default
Commit Block Size (COMMIT_BLOCK)	(Required) Commit block size for performance tuning of updates. If left blank, 200 will default.	200

Exception Report File Location (EXCP_RPT_FILE_LOC)	Exception Report File Location for Generate Payroll Exception Report Job. Must be same through all steps.	$$AMSROOT$$/ExportImport/GenPYRL
Exception Report File Name (EXCP_RPT_FILE_NM)	Exception Payroll Report File Name. Must be same through all steps.	FailedPYRLExcepReport.txt
Exception Report Indicator (EXCP_RPT_IND)	(Required) Exception Report Indicator. 1 = Detailed, 2 = Summarized, if not entered then defaulted to 1 (Detailed).	1
Gross to Net Run Number (GTN_RUN_NO)	(Conditionally Required) Gross To Net Run Number to focus in on set transactions to generate. This can be a comma separated list of GTN_RUN_NOs. Required if Type of Payroll Transaction is P; optional if Type of Payroll is A; otherwise, it is prohibited.	No Default
Progression Message Count (PROG_CTR_SZ)	(Required) Progression Counter Size used to display periodic messages in the job log. If left blank, 5000 will default.	5000
Read Fiscal Year by Department Option (READ_FY_BY_DEPT)	(Required) Read FYDEPT data: 1 = Yes or 0 = No.	0
Read Fiscal Year by Fund Option (READ_FY_BY_FUND)	(Required) Read FYFD data: 1 = Yes or 0 = No.	0
Read Accounting Period by Department Option (READ_PER_BY_DEPT)	(Required) Read APDDEPT data: 1 = Yes or 0 = No.	0
Read Accounting Period by Fund Option (READ_PER_BY_FUND)	(Required) Read APDFD data: 1 = Yes or 0 = No.	0
Select Block Size (SELECT_BLOCK)	(Required) Select block size for performance tuning of record selection. If left blank, 500 will	500

	default.	
Type of Payroll Transaction (TYPE_PYRL_DOC)	(Required) Type of Payroll Transaction to process. P = All Transactions; C = Contract Pay: PRLCE only; L = Leave Liability: PRLLL only; A = Adjustment: PREXC only. Will not default to P if left blank.	P

Of particular note are the Read FY and APD parameters. If the job step finds that a combination of FY and APD or just the FY is closed for the Fund or the Department on an accounting line (vendor lines too of the PRLID), then the job will increment by 1 until it finds the next open value (APD 0 and 99 excluded). If any of these four are not used, then turning off the lookup may improve performance slightly. If used, but you do not want the value incremented because either an override applied later will bypass the closed FY or APD (based on DCTRL settings to do so) or you wish the transaction to reject (uncommon but possible), then you should leave the parameter as *No (0).*

Major Output

- PYRL Transactions in XML File
- Chain Job Parameter File
- Update HRM table of R_PYRL_HDR with FIN_PROCESSED_FL to be *true (1)* for all of the transactions written to XML file.

Batch Job Return Codes

The following table shows the potential return codes for the Generate Payroll Transactions job step.

Return Code	Condition
Successful (1)	Everything is successfully processed.
Warning (4)	No records were found to match selection criteria.
Non-Fatal Error (8)	This job step does not return this code.
Failed (12)	Failures as documented in the Process Step messages section, which shows failure messages.
Terminated (16)	The job is manually terminated.
System Failure (20)	The system encountered fatal failure.

Sort Sequence

The Sort Order for acquiring R_PYRL_HDR records is: DOC_CD, DOC_DEPT_CD, DOC_ID, and DOC_VERS_NO.

The Sort Order for acquiring Vendor Line(s) for the associated Header inside of R_PYRL_VEND is: DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, and DOC_VEND_LN_NO.

The Sort Order for acquiring Accounting Line(s) for the associated Vendor Line inside of R_PYRL_ACTG is: DOC_CD, DOC_DEPT_CD, DOC_ID, DOC_VERS_NO, DOC_VEND_LN_NO, and DOC_ACTG_LN_NO.

Selection Criteria

For All records, FIN_PRCESSED_FL is false. Additionally, for TYPE_PYRL_DOC,

If P (ALL):

- GTN_RUN_NO is in the set of supplied GTN_RUN_NOs supplied as a batch parameter.
- DOC_SUBTYP does not equal PREXC

If A (Adjustment):

- If GTN_RUN_NO is provided, GTN_RUN_NO is in the set of supplied GTN_RUN_NO's as a batch parameter.
- If GTN_RUN_NO is not provided, all GTN_RUN_NO's are picked up.

If C (Contract:)

- GTN_RUN_NO is NULL.
- DOC_SUBTYP equals PRLCE

If L (Leave Liability):

- GTN_RUN_NO is NULL.
- DOC_SUBTYP equals PRLLL

Soft Closed Logic

The capability exists in FIN to leave a fiscal year or accounting period open on the FY or APD pages, while disallowing access to a specific fund or department with a soft close. As a part of this step of the chain job, logic was added to allow the person running this step to choose if they wish to ensure that Accounting Lines being created for a specific PYRL transaction will not use an FY or an Accounting Period that is closed for the Fund or Department on the Accounting Line. HRM continues to check for the soft closed status on the main FY and APD pages replicated from FIN to HRM.

If your site uses one to all of these pages and wishes to increment beyond a soft close FY or APD, then when running this job you should set the appropriate parameter(s) to *true*. When the accounting line (and vendor line for the PRLID) is being written out to the XML file, the job will execute logic that will ensure that for that Fund or Department the FY or APD on the R_PYRL_HDR is not soft closed. If it is, the logic will calculate the next available FY which is open for the Fund or Department by querying FYFD or FYDEPT. Next, logic will use the calculated FY and the header's APD to see if it is open for the Department or Fund on APDFD or APDDEPT. If it is not, then logic will calculate the next open Accounting Period for the FY (99 and 0 excluded). If there is no open APD for the calculated FY, it will then move to the next future FY and will look for the first open APD (0 and 99 excluded).

If the user requests the checking of the soft closed table(s) for the Fund or the Department and the Accounting Line being processed does not have a Department listed, then it will not perform the soft closed check for FYDEPT or APDDEPT.

Problem Resolution

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

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	Required Parameters are not entered.	Schedule a new job by passing required parameters.	N/A
	Failed because of an invalid parameter	Schedule a new job by passing required parameters.	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)	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

Step 2: Cross Field Validations

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 Error (8)	N/A	N/A	N/A
Failed (12)	Conflict between GTN_RUN_NO and the TYPE_PYRL_DOC parameter value	Please correct the incorrect value and schedule another chain 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

Step 3: Write Out Parameters to a Chain Parameter File

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All values written out successfully to Chain Parm File.	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	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. 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

Step 4: Initialize Transaction XML

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Transaction XML File was successfully initialized.	N/A	N/A
Warning (4)	N/A	N/A	N/A

Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	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. 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

Step 5: Build Payroll Transactions

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All desired transactions were successfully built and written to the XML File.	N/A	N/A
Warning (4)	If no records were found to process.	Investigate whether the parameters supplied were incorrect or if there was some other unexpected problem in identifying the R_PYRL_HDR records.	Verify R_PYRL_HDR records have been created for the activity expected. Verify whether Payroll Transaction Generator has already been run for the data set.
Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	There are many items that could cause a failure. Example: “Problem creating transaction for [transaction information supplied]”	The reason for the failure needs to be investigated before scheduling a new job. There could have been no VL(s) found for the Header or no AL(s) found for the VL(s). Or some critical information was not found.	N/A
	Failed because of runtime exceptions for an unexpected	The reason for the failure needs to be investigated before scheduling a new	N/A

	situation.	job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	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

Step 6: Save XML File

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Transaction XML File was successfully saved.	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	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. 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

Payroll Transaction Generator: Load and Submit Transactions

Job Name	Load and Submit Transactions
Recommended Frequency	See chain job

Single Instance Required	Yes
Can the job be restarted?	Yes. Please see the "Problem Resolution" section for details.
Reports generated	No

Overview

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

During this automatic submission of the PYRL transactions as well as any manual submits of rejected transactions, many expenditure entries will get written into PYRLAR capturing accounting lines in their final state. The PYRLAR data is used by a different job inside of HRM to reconcile the accounting lines HRM created and the accounting line FIN eventually submitted to final.

This job step consists of following basic steps:

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters. • Each valid parameter is listed followed by any value. • If the Thread Count is not a valid integer then "Invalid value received for parameter THREAD_COUNT (expected number): ##" is recorded in the job log. • If the Thread Count is not a valid integer > 0, then "The valid job thread count starts with 1. Receive value: ##" is recorded in the job log. • If the Mode is not provided then "MODE parameter cannot be left blank (expected values 1, 2, or 3)" is recorded in the job log. • If the Mode is provided and Mode is not => 1 and not =< 3 then "Invalid value received for parameter MODE (expected values 1, 2, or 3): ##" is recorded in the job log. • If the Commit Block Size is not a valid integer > 1 then "Invalid value received for parameter COMMIT_BLOCK_SIZE (expected value >1): ##" is recorded in the job log. • If the Block Size is not a valid integer > 1 then "Invalid value received for parameter BLOCK_SIZE (expected value >1): ##" is recorded in the job log. • If the Polling Frequency is a valid integer and < 5 then "Invalid value received for parameter SLEEP_INTERVAL (expected value >=5): ##" is recorded in the job log. • If the Logging Frequency is a valid integer and less than 30 then "Invalid value received for parameter

	LOG_STATUS_INTERVAL (expected value >= 30): ##" is recorded in the job log. • If the Catalog ID of the System Maintenance Utility job is not a valid integer greater than one then "Invalid value received for parameter SMU_CTLG_ID (expected value >1): ##" is recorded in the job log. • If the Other Action is not provided and run mode is 3, then "Other Action is required when run on mode 3" is recorded in the job log. • If the Other Action is provided and run mode is 1 or 2 then "Other Action is valid only for Mode 3" is recorded in the job log. • If the location of the source file of records is not a valid directory, then ''Invalid value read for parameter FILE_INPUT_DIR (expected value = a valid directory): ####" is recorded in the job log. • If the Stagger Time is not a valid integer > 1 then ''Invalid value received for parameter STAGGER_TIME (expected value >1): ##" is recorded in the job log. • If the Comma Separated list of input XML files is not provided, then "No input file to process" is recorded in job log. • If the Comma Separated list of input XML files is provided and a file does not exist or does not end with xml then "Invalid / non existing file input part in value for parameter FILE_LIST (expected value = a valid list if file):####" is recorded in job log. • If the Output Location for the file segments is not a valid directory then ''Invalid value read for parameter FILE_OUTPUT_DIR (expected value = a valid directory): ####" is recorded in the job log. • Parameter validation completed.
2. Multiple Job Processing	• Setting up job for parameter file: (Parameter Location • path)\Parm_######_#.txt • Enabled job with Id=###### • Created Job ###### for Parm file (Export Import directory path)\Parm_######_#.txt • (Repeated for each file and job combination) • Committed child jobs • Child job pending count:# Job Id [######] Status=X • (Repeated for each job) • Deleting part exception file: (FILE_OUTPUT_DIR)\FailedPYRLDocExcepReport _######_#.txt • (Repeated for each file) • Deleting split input XML file: (FILE_OUTPUT_DIR)\PYRLDocuments_######_#.xml, • (Repeated for each file)

	• Run Ended

Major Input

- GenPYRLDocs.xml

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
Transaction Split Bock Size (BLOCK_SIZE)	(Optional) Block size for the number of transactions to be split out into files. If left blank, 1 will be the default. Set this performance parameter to a value that will split an XML file into individual files sufficient for timely processing.	1
Commit Block Size (COMMIT_BLOCK_SIZE)	(Optional) Number of records to commit at a time. If left blank, the ADV30Parms.ini file will supply a value.	10
File Input Directory (FILE_INPUT_DIR)	(Required) Location of the source file of records.	$$AMSROOT$$/ExportImport/GenPYRL
XML File List (FILE_LIST)	(Required) Comma separated list of input XML files to load.	GenPYRLDocs.xml
File Input Directory (FILE_OUTPUT_DIR)	(Required) Output location for the file segments.	$$AMSROOT$$/ExportImport/GenPYRL
File Prefix (FILE_PREFIX)	(Optional) File prefix for Parameter and Data Files created by jobs.	No default
Apply Overrides in Import (I_SMU_APPLY_OVERRIDES)	(Required) System Maintenance Utility flag to indicate Apply Overrides in Import Mode.	true
Bypass Approvals in Import (I_SMU_BYPASS_APPROVAL)	(Required) System Maintenance Utility flag to indicate Bypass Approvals in Import Mode.	false
Bypass Automatic Transaction Numbering (I_SMU_BYPS_ADNT_FL)	(Required) System Maintenance Utility flag to indicate Bypass ADNT in Import Mode. Should always be true because HRM supplies Transaction ID values.	true

Import Transaction Status (I_SMU_DOC_STA_CD)	(Required) System Maintenance Utility Transaction Status for Import Mode.	2 (Ready)
Override Level in Import (I_SMU_OVERRIDE_LVL)	(Optional) System Maintenance Utility Override Level for Import Mode. If 0 or left blank and the Apply Overrides parameter is *true* then the level of the user executing the chain job will be used.	No default
Save Restart (I_SMU_RESTART_FL)	(Required) System Maintenance Utility flag to indicate Save Restart info in Import Mode.	true
Status Log Frequency (LOG_STATUS_INTERVAL)	(Required) Logging frequency (in seconds) for controller thread reporting status of child threads to the system log.	300
Run Mode (MODE)	(Required) Run mode (1 = Import only, 2 = Import and Submit, 3 = Import, Other Action, and Submit).	2
Apply Overrides in Submit (S_SMU_APPLY_OVERRIDES)	(Required) System Maintenance Utility flag to indicate Apply Overrides in Submit Mode.	true
Bypass Approvals in Submit (S_SMU_BYPASS_APPROVAL)	(Required) System Maintenance Utility flag to indicate Bypass Approvals in Submit Mode.	false
Exception File Name (S_SMU_EXCEP_REP_FILE_NM)	(Required) System Maintenance Utility Exception File Name for Submit Mode.	FailedPYRLDocExcepReport.txt
Exception Report Mode (S_SMU_EXCEP_REP_IND)	(Required) System Maintenance Utility Exception Report Indicator for Submit Mode. Values are: 1 = Detailed, 2 = Failed Transactions, 3 = Processed Transactions, 4 = Failed Transaction Lines and 5 = Transaction Status.	4
Set Severity Flag (S_SMU_EXP_SEV_FL)	(Optional) System Maintenance Utility flag to indicate the Severity Flag in Submit Mode.	1
Override Level in Submit (S_SMU_OVERRIDE_LVL)	(Optional) System Maintenance Utility Override Level for Submit Mode. If 0 or left blank and the Apply Overrides parameter is *true* then the level of the user executing the chain job will be	

	used.	
Save Restart Information (S_SMU_RESTART_FL)	(Required) System Maintenance Utility to indicate Save Restart info in Submit Mode.	true
Sleep Interval (SLEEP_INTERVAL)	(Required) Polling frequency (in seconds) for internal controller thread for checking child processes.	5
SMU Catalog ID (SMU_CTLG_ID)	(Required) Catalog ID of the System Maintenance Utility job that is spawned as the child process.	3
Stagger Time (STAGGER_TIME)	(Required) Lag time in seconds between the spawning of each child process.	30
Thread Count (THREAD_COUNT)	(Required) Number of jobs to start. Set for optimal performance given system capabilities and concurrent processing.	1

Major Output

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

Batch Job Return Codes

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

Return Code	Condition
Successful (1)	All of the records are loaded into the Transaction Catalog successfully or the input files are empty. All of the transactions loaded were submitted successfully.
Warning (4)	This return code is issued when some of the records failed to load but other records were loaded successfully.
Non Fatal Error (8)	None of the records were loaded into the Transaction Catalog.
Failed (12)	• Parameters are invalid

	• When the input file is not found in the specified directory • Restart failed because another instance of the Payroll Transaction Generator Chain has already been run successfully • Runtime exceptions encountered for any unexpected situations When the job ends with a return code of Failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain 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 will be set to inactive.

Sort Sequence

N/A

Selection Criteria

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

Problem Resolution

If the process fails for any reason, the 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, that is, the third and the fourth ones will be picked up and processed. The restarted job would not process child jobs one, two or three given they have already completed successfully in the earlier run.

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All transactions were loaded and submitted.	N/A	N/A
Warning (4)	This return code will be issued when the job fails to load some of the transactions. **Sample Message:**	Analyze the reason why records failed to load to the Transaction Catalog. When the current run is a re-execution of a	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted –

	Unable to load all of the transactions into the catalog.	previous run (the same input file is used in a new job) this error may be seen because the records that previously loaded successfully will not load again (duplicate entry on the Transaction Catalog is not allowed). If the records failed to load to the Transaction Catalog due to any other reason, then analyze the reason, resolve it, and schedule a new job.	only a new job should be scheduled.
Non-Fatal Error (8)	This return code is issued when the job failed to load all of the transactions.	Correct what is most likely a setup issue and restart the spawned SMU job and then restart the load/submit job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.
Failed (12)	In this step, the job can fail under the following conditions: • Issues in spawning jobs • Files not found • Runtime exceptions • Parameter Edits	The reason for the failure needs to be investigated and fixed before restarting the job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated and fixed before restarting the chain.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before restarting the chain	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.

Payroll Transaction Generator: Generate Payroll Exception Report

Job Name	Generate Payroll Exception Report
Recommended Frequency	See chain job
Single Instance Required	Yes
Can be restarted?	No
Report Generated	Yes

Overview

The final job in the Payroll Transaction Generator Chain generates an exception report that lists all of the transactions that did not submit to the *Final* Transaction Phase. When run in the detail mode the errors are also listed for each Payroll (PYRL) transaction from earlier submit job steps with reference to a line within the transaction.

Major Input

- FailedPYRLExcepReport.txt [EXCP_RPT_FILE_NM]
- PyrlActgChain.txt [CHAIN_PARM_FILE]

Batch Parameters

Parameter	Description	Default Value
Commit Block (COMMIT_BLOCK)	(Required) Commit Block Size used for performance. While no database records are altered it is used in assisting memory collection.	500
Progression Message Count (PROG_CTR_SZ)	(Required) Progression Counter Size used to display periodic messages	5000
Parameter File Location (AMSPARM)	(Required) Parameter Location for Generate Payroll Exception Report Job	$$AMSROOT$$/Parms
Chain Parameter File Name (CHAIN_PARM_FILE)	(Required) Common Chain Parameters File (.txt)	PyrlActgChain.txt
Exception Report File Name (EXCP_REP_FILE_NM)	(Required) Exception Payroll Report File Name	FailedPYRLDocExcepReport.txt
Exception Parameter File Location (EXCP_PARM_FILE_LO	(Required) Exception Report File Location for Generate Payroll Exception Report Job	$$AMSROOT$$/ExportImport/GenPYRL

C)		

Major Output

- Payroll Detail Exception Report when Exception Report Type [EXCP_RPT_IND] is 1 (Detail) in the first job of the Chain job
- Payroll Summary Exception Report when Exception Report Type [EXCP_RPT_IND] is 2 (Summary) in the first job of the Chain job.

Batch Job Return Codes

The following table shows the potential job return codes for the individual Generate Exception Report.

Return Code	Condition
Successful (1)	All of the validations are performed successfully and there are no records eligible for exception.
Warning (4)	N/A
Non-Fatal Error (8)	At least one transaction failed to submit to the *Final* Phase.
Failed (12)	• Input parameter file is not found. • Runtime exceptions encountered for any unexpected situations. • Technical failure. When the job ends with a return code of failed, subsequent jobs in the chain will be set to inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain are set to inactive.

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

Job cannot be restarted in this step. If the job fails a new job should be scheduled after correcting the errors that caused the job to fail.

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

Step 1: Batch Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All batch parameter validations are successful (that is, there are no errors raised during parameter validations.)	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non-Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Required parameters are not entered. **Sample Message**: Parameter file name is not specified.	Enter missing required parameters and schedule a new job. Enter the parameter file name.	N/A
	Failed because of runtime exceptions caused by unexpected conditions.	Failure reason needs to be investigated before rescheduling the job.	N/A
Terminated (16)	Job is terminated manually by the user.	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. A new job can be scheduled.	N/A

Step 2: Exception Report Generation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All PYRL transactions submitted to the *Final* transaction phase and the exception report was generated successfully.	N/A.	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non-Fatal Error (8)	At least one PYRL failed to submit to the *Final* transaction phase.	N/A	N/A
Failed (12)	Failed because of runtime exceptions caused by unexpected conditions.	Failure reason needs to be investigated before rescheduling the job.	N/A

Terminated (16)	Job is terminated manually by the user.	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. A new job can be scheduled.	N/A

2.1.29 Populate FY Beginning Balance

Chain or Job Name	Populate FY Beginning Balance
Recommended Frequency	Annually after the Annual Close Chain and the Ledger Engine has ledgerized closing journal records.
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	No

Overview

This batch job runs after an Annual Close and a running of the Ledger Engine to ledgerize the journal records from Annual Close. It makes updates to the FY Beginning Balance amount on the Fund (FUND) and Sub Fund (SFUND) pages with the cumulative amount from an input ledger. The job obtains the Close Into value from FUND for each fund code. The values for Close Into field correspond to one of three balance sheet fields on the Miscellaneous tab of Special Accounts (SPEC): Fund Balance, Retained Earnings, and Agency Due. Ledger records for those accounts are then selected from Accounting Period 0 for the year specified in the batch job.

There is one instance where this job will not directly follow the Ledger Engine after Annual Close. This batch job is the means for populating the FY Beginning Balance amount for all prior years already closed. If the original conversion year did not put opening balances in APD 0, then the amount will have to be populated by another means, but with the same selection logic.

Should updates be made to APD 0 after the running of this job, it can be run again as it will replace amounts instead of add to them.

Please note that any other accounts for other types of equity will not be selected by this program as it can only match on one of three balance sheets. However, the program does not select based on posting code, so if the same equity balance sheet account is used with different posting codes for reserved equity accounts, all such records will be combined.

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

Process Steps	Messages
1. Parameter Validation	• Run Started • Parameters are listed with any invalid ones followed by an error message
2. Selection of Records	No messages for this step
3. Update FY Beginning Balances	• Number of records updated on FUND • Number of records updated on SFUND

Restartability Information

If the job fails in any of the above steps the job can be rescheduled after resolving the error. The job will start from the beginning.

Major Input

- Fund (R_FUND)
- Sub Fund (R_SFUND)

Batch Parameters

Parameter	Description	Default Value
Opening AFY	Opening Accounting Fiscal Year. This should be the same value as the Opening Accounting Fiscal Year entered for the Annual Close process in step 1. Enter as CCYY.	No Default
Ledger Name	This is a required parameter. This parameter will be used to specify Ledger name.	No default, but one should be setup on BATSETUP such as LDGR_APD_ACTG.

Output

- Fund (R_FUND)
- Sub Fund (R_SFUND)

Job Return Codes

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

Return Code	Condition
Successful (1)	All of the selected payment records are processed successfully
Warning (4)	The job does not end with this return code
Non-Fatal Error (8)	The job does not end with 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 will be issued when the job is terminated by the user.

System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Sequence

- Fund, Sub Fund

Selection Criteria

- Fiscal Year (FY_DC) of ledger record matches Opening AFY batch parameter
- Balance Sheet (BSA_CD) of ledger record matches one of three balance sheets obtained from SPEC
- Accounting period (APD_DC)

Problem Resolution

If the job fails due to any reason, the job can be restarted or another instance can be run.

2.1.30 Pre-Annual Close Sweep

Chain or Job Name	Pre-Annual Close Sweep chain job
Recommended Frequency	On demand at year end before Annual Close
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	Exception report

Overview

When the time comes to run the Annual Close chain there can be undesired balances left in the closing year. These balances are undesired because they impact the Fund Balance, Retained Earnings, or Agency Due accounts. These balances result from two situations:

1. Pre Encumbrance and Encumbrance activity recorded with a BFY of 9999 because of budget setup that is referred to as 'multi-year'.
2. Pre Encumbrance and Encumbrance activity recorded with a regular BFY (that is, 2010) that was liquidated in a Fiscal Year (FY) greater than the BFY. This situation results when there is open prior-year activity that has not been lapsed or rolled yet and BFY Staging setup requires/allows users to liquidate that activity with an FY > BFY. At some point any remaining activity is then rolled or lapsed.

   This situation is not an issue for a site that leaves Pre Encumbrance or Encumbrance transactions open in the original BFY until paid (i.e. do not roll or lapse).

The Pre-Annual Close Sweep chain (referred to as PACS from this point forward) is designed to move balances in the above situations from one fiscal year to the next. PACS reads in a ledger to identify the open activity in the old year; create, load, and process journal voucher transactions to 'sweep' that activity out of the old year into the new; and then produces an exception report of any transactions that did not submit to final. As the process sweeps at a summary level and not an accounting line level, there will be no impact to the open transactions which recorded the open activity. Subsequent modifications, cancellations, and liquidations to that open activity are recorded in the fiscal year of that modification, cancellation, or liquidation.

The running of PACS should follow the running of one or more of the following processes:

- Open Activity Roll
- Open Activity & Budget Roll
- Open Activity Accrual
- Open Activity Lapse

After these processes, the only activity left in the year is that which needs to be swept or is closed as part of Annual Close. It is not an absolute that the sweep follows all of the above, but running it before all of the jobs are run requires input parameters of either Fund or Fund Type that use BFY 9999 exclusively. Lapsing inactive multi-year activity with the Open Activity Lapse chain is recommended prior to PACS as it reduces the amount that need to be swept from one fiscal year to the next, but this is not required. The inactive balances can be lapsed after being swept with parameters that ensure that lapse occurs in the year to which balances were swept.

The PACS should not be run in consecutive runs with the same selection or overlapping selection criteria unless the journal records created from the first run have first been ledgerized by running the Ledger Engine. To not do so would double count swept activity.

Please see the *Year End Manual* for more information on the various methods for running the sweep and interactions with other year end processes.

There is no report mode for this process as reports such as the Trial Balance can be used to capture before and after sweep activity. Online ledger pages such as the Accounting Ledger – FY (LFYACTG) can be used to capture before and after sweep activity.

As the process uses a specified ledger as input, pre-processing steps should include:

1. Getting pending transactions against the prior year through or out of workflow
2. Ensuring all posting lines for accepted transactions have been journalized by running the Journal Posting Initiator job and then the Journal Engine.
3. Ensuring all journal records have been ledgerized by running the Ledger Engine in *Normal*, *Failed Work*, and *Gap* modes. Then running System Assurance 3 and 7 against the input ledger.

As PACS uses a ledger as input, if there is a mixture of multi-year and regular BFY activity to be swept, it is strongly suggested that selection criteria of Fund or Fund Type be used to select only one or the other type of activity. The reason is that the summarization level of the input ledgers in both cases may be very different. Regular BFY activity needs only to be swept at a Fund, Sub Fund, BSA, and Sub BSA level, commonly found on the ledger used for annual close – LDGR_FYDAD. That ledger does not normally retain BFY. It is not a requirement to use this ledger as the sweep can be done at a lower COA level of detail, but it is not necessary and increases running time and volume. However, if additional detail is required, it can be done. Please note that the BFY on activity swept out of the old FY equals that FY and on activity swept into the new FY the BFY will again be equal to that new FY. Because of this, a sweep should not be done with sufficient COA detail to update budgets because the transaction activity that lead to the open activity situation was performed with a single BFY equal to the prior FY. That is to say a BFY/FY 2009 PO referenced by a BFY 2009 and FY 2010 PRC used a single BFY so the BFY 2009 budget is correct and the sweep should not impact it.

To sweep multi-year activity, a different ledger is needed other than the LDGR_FYDAD as it will not likely contain BFY. The PACS job needs the BFY in the ledger when sweeping 9999 activity as it places it in the generated XML. With the BFY in the ledger, all the offsetting equity accounts will also have 9999 so the journal vouchers created as well as all updated journals and ledgers will continue to balance by BFY.

When sweeping either type of activity, it is recommended that the input ledger have the Real Account Level setting on Journal Ledger Control (JLCTRL) set to Fund and BSA Levels. Any other setting only serves to increase run time and volume to only create equity accounting lines with extra COA detail.

The Pre-Annual Close Sweep chain has the following jobs (each of the jobs listed below, is described in the subsequent sections):

- **Generation of Sweep XML:** Generate XML for JV Header, JV Line Group and JV Accounting Line.
- **Sweep Transaction Load:** Existing SMU job is used to load the Journal Voucher (JV) transactions from the XML file created in the above step.
- **Sweep Transaction Submit:** Existing SMU job is used to submit the Journal Voucher (JV) transactions created in above step.
- **Sweep Exception Report:** An exception report is generated for each Header / Line Group / Accounting line which failed to submit in the previous step.

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

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

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

Major Input

- Ledger (LDGR_FYDAD, LDGR_SA_BUD, or other)

Major Output

- JV Transactions

Chain Job Return Code

The following table shows the potential return codes for the Pre-Annual Close Sweep chain job. Note that the Chain job ends with the highest return code across all of the jobs.

Return Code	Condition
Successful (1)	All of the jobs end successfully.
Warning (4)	The first job in the chain found no records for sweeping and ended with a return code of *Warning*.
Non-Fatal Error (8)	The Exception Report has ended with a return code of *Non-Fatal Error* because one or more transactions failed to submit.
Failed (12)	One of the jobs in the chain ends with a return code of *Failed*.
Terminated (16)	One of the jobs in the chain ends with a return code of *Terminated*.
System Failure (20)	One of the jobs in the chain ends with a return code of *System Failure*.

Problem Resolution

If any of the jobs in the chain failed due to application errors, it is advisable to restart the job after correcting the errors instead of rescheduling the job. Restarting the job reduces the processing time since the job resumes from where it has last committed and selects only the unprocessed records.

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

Pre-Annual Close Sweep Chain: Generation of Sweep XML

Chain or Job Name	Generation of Sweep XML

Recommended Frequency	See chain
Single Instance Required	See chain
Can be restarted?	Yes
Reports generated	Exception report

Overview

Common activity swept out of a fiscal year includes:

- Open amounts for procurement transactions which cannot be rolled or lapsed because they are multi-year (i.e. BFY 9999).
- Balances remaining in the prior year from modifications, cancellations, or liquidations to procurement transactions where the fiscal year was greater than the budget fiscal year.

The running of the sweep process should follow the running of one or more of the following processes: Open Activity Roll, Open Activity & Budget Roll, Open Activity Accrual, or Open Activity Lapse. The sweep can be performed more than once if required. It should not be run in consecutive runs with the same selection or overlapping selection criteria unless the journal records created from the first run have first been ledgerized by running the Ledger Engine. To not do so would double count swept activity.

Please see the *Year End Manual* for more information on the various methods for running the sweep and interactions with other year end processes.

There is no report mode for this process as reports such as the Trial Balance can be used to capture before and after sweep activity. Online ledger pages such as the Accounting Ledger – FY (LFYACTG) can be used to capture before and after sweep activity.

As the process uses a specified ledger as input, pre-processing steps should include:

- Getting pending transactions against the prior year through or out of workflow.
- Ensuring all posting lines for accepted transactions have been journalized by running the Journal Posting Initiator job and then the Journal Engine.
- Ensuring all journal records have been ledgerized by running the Ledger Engine in *Normal*, *Failed Work*, and *Gap* modes. Then running System Assurance 3 and 7 against the input ledger.

This first job step in the chain consists of following basic steps:

- **Parameter Validation:** In this step, the process verifies the parameters. If the parameter validation is successful, the job goes to the next step. Otherwise the job ends with a return code of *Failed*.
- **Processing Of Records:** In this step the selection of records based on various selection parameters is made from the input ledger.
- **Generate XML:** Generate XML for JV Header, JV Line Group and JV Accounting Line by using above selected records. Two accounting lines are created for each selected ledger record, one for the 'swept from' year and one for the 'swept into' year. A separate transaction is generated for each Fund and Sub Fund combination selected with a separate transaction for each year. At any point where the lower of the Number of Accounting Lines SOPT setting for the Selection FY parameter or MAX_LINE_LIMIT setting for JV_DOC_ACTG on DCREQ in the Administration application.

Process Steps	Messages

1. Parameter Validation	• Validating Batch Parameters. • If input batch parameters are invalid, the invalid value is displayed in the log along with the error message. • If the required parameters are not provided then <<PARM_NM>> is required message is added in Job Log. • If all parameters pass validation the following message is displayed in the log: "Parameter validation completed successfully."
2. Processing of Records	• Selecting the records from Input Ledger table for report mode. • Selection of records for generating report completed. • If the selection returns 0 records then the following message is issued: "No records selected for report generation".
3. Generate XML	• Selecting the records from Input Ledger table. • If no records are selected then the following message is issued: "No records selected for processing"

Restartability Information

This process is implemented with checkpoints. If the process fails for any reason (such as the network is down or the server is down), then the process can restart from the last point where it stopped.

If the job fails in any of the above steps the job can be restarted after resolving the error. If the job is restarted, it starts from the last point where it stopped.

For example, the job failed after processing 2 records from the total of 25 eligible records due to some fatal condition (table space error). If the job is restarted after resolving the table space issue, the job starts by selecting the 3rd record. Instead of restarting the chain, if a new chain is rescheduled with the same parameter values after resolving the table space issue, the job starts from the beginning.

A restart fails if another instance of this job has been scheduled and ran successfully before restarting the failed job.

Major Input

- Ledger (LDGR_FYDAD, LDGR_SA_BUD, or other)

Batch Parameters

Parameter	Description	Default Value
Export Location	Required location where the XML file is written.	$$AMSROOT$$/ExportImport

(AMSEXPORT)		
Log File Location (AMSLOGS)	Required location where the submit job step creates a file for the exception report job step.	$$AMSROOT$$/Logs
Parameter File Location (AMSPARMS)	Required location where the load and submit parameter files are written for later job steps.	$$AMSROOT$$/Parms
Balancing Posting Code (BAL_PSCD)	Required output parameter that is used to balance journal vouchers when an accounting line limit is reached.	A204
Client Name (CLIENT_NAME)	Optional parameter for the specification of a name to appear on the exception report.	No default
Transaction Code (JV_DOC_CD)	Required output parameter to specify which journal voucher transaction code is used.	JVA
Transaction Department (JV_DOC_DEPT)	Required output parameter for transaction numbering.	No default
Transaction Prefix (JV_DOC_PFX)	Optional output parameter for transaction creation for identification purposes.	No default
Transaction Unit (JV_DOC_UNIT)	Optional output parameter for transaction security.	No default
Record Date (DOC_DT)	Optional output parameter for transaction creation when the application date should not be used on generated transactions. Please enter as MM/DD/CCYY. If left blank the transactions default the current application date.	No default
Increase APD (DOC_INC_PER)	Required output parameter for accounting line APD on transactions created for the year after the SEL_FY.	No Default
Decrease APD (DOC_RED_PER)	Required output parameter for accounting line APD on transactions created for the SEL_FY.	No Default
XML File Name (DOC_XML_FILE)	Required parameter for an XML file name that contains all journal voucher transactions created.	PACSDocuments.xml

Encumbrance Increase Posting Code (ENC_INC_PSCD)	Optional output parameter that is used as the encumbrance increase posting code in the year after the SEL_FY. If left blank, ENC_SEL_PSCD is used. Using a different posting code here allows for a clear indication of encumbrances swept from the prior year.	P015
Encumbrance Reduction Posting Code (ENC_RED_PSCD)	Optional output parameter that is used as the encumbrance reduction posting code in the SEL_FY. If left blank, ENC_SEL_PSCD is used. Using a different posting code here allows for a clear indication of encumbrances reduced for a sweep separate from other types of reductions.	P015
Encumbrance Selection Posting Code (ENC_SEL_PSCD)	Required selection parameter to represent encumbrance activity to be swept. Multiple values can be entered if separated by commas.	P005, P006, P015
Input Ledger (LDGR)	Required input ledger. Please enter as the data object name.	LDGR_FYDAD
Load Parameter File (LOAD_PARM_FILE)	Required load parameter file (.txt) for subsequent load job step.	PACSLoadParams.txt
Pre-Encumbrance Increase Posting Code (PENC_INC_PSCD)	Optional output parameter that is used as the pre-encumbrance increase posting code in the year after the SEL_FY. If left blank, PENC_SEL_PSCD is used. Using a different posting code here allows for a clear indication of pre-encumbrances swept from the prior year.	P014
Pre-Encumbrance Reduction Posting Code (PENC_RED_PSCD)	Optional output parameter that is used as the pre-encumbrance reduction posting code in the SEL_FY. If left blank, PENC_SEL_PSCD is used. Using a different posting code here allows for a clear indication of pre-encumbrances reduced for a	P014

	sweep separate from other types of reductions.	
Pre-Encumbrance Selection Posting Code (PENC_SEL_PSCD)	Required selection parameter to represent pre-encumbrance activity to be swept. Multiple values can be entered if separated by commas.	P003, P004, P014
Selection Fund Type (SEL_FTYP)	Optional selection parameter of fund type when all activity should not be swept. Comma-separate multiple values.	No default
Selection Fund (SEL_FUND)	Optional selection parameter of fund when all activity should not be swept. Comma-separate multiple values.	No default
Selection Year (SEL_FY)	Required selection parameter to identify the Fiscal Year or Budget Fiscal Year for ledger record selection. The year entered is interpreted as either Fiscal Year (FY) or Budget Fiscal Year (BFY) based on the value entered in SEL_FY_TYP. Please enter as CCYY.	No default
Selection Year Type (SEL_FY_TYP)	Required selection parameter to differentiate between Fiscal Year (FY) and Budget Fiscal Year (BFY) for the year entered in SEL_FY. The entered value is checked against the Fiscal Year/Budget Fiscal Year indicator for the input ledger (LDGR) on Journal/Ledger Control Detail (JLCTRL). If FY is selected, the Fiscal Year indicator on the input ledger must be *true*. If BFY is selected, the Budget Fiscal Year indicator on the input ledger must be *true*.	FY
Submit Parameter File (SUBMIT_PARM_FILE)	Required submit parameter file (.txt) for subsequent submit job step.	PACSSubmitParams.txt

Major Output

- PACSDocuments.xml (XML file of JV transactions for the load job step)
- PACSLoadParams.txt (Instruction file for the load job step)
- PACSSubmitParams.txt (Instruction file for the submit job step)

Job Return Codes

The following table shows the potential job return codes for the Pre-Annual Close Sweep job.

Return Code	Condition
Successful (1)	All the selected ledger records were processed successfully.
Warning (4)	No eligible records found.
Non-Fatal Error (8)	n/a
Failed (12)	The job fails under the following conditions: • 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 Sequence

The sorting performed is a product of journal voucher creation where all lines for a Fund, Sub Fund, and Fiscal Year are placed on a single transaction.

Selection Criteria

- Fiscal Year (FY_DC) or Budget Fiscal Year (BFY) matches the Selection Fiscal Year parameter (SEL_FY) based on the Selection Year Type (SEL_FY_TYP).
- Amount (AM) of the ledger record is not $0.00.
- Fund Type (FTYP_CD) matches any supplied Selection Fund Type parameter (SEL_FTYP), otherwise all Fund Types are selected.
- Fund (FUND_CD) matches any supplied Selection Fund parameter (SEL_FUND), otherwise all Funds are selected.
- Posting code (PSTNG_CD_ID) matches one of the Pre-Encumbrance Selection (PENC_SEL_PSCD) or Encumbrance Selection (ENC_SEL_PSCD) parameters.

Problem Resolution

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

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters were valid.	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 not entered or invalid parameters entered. **Sample Message:** Select FY cannot be blank.	Correct the parameter issue and run the chain again.	n/a
	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before restarting the job.	n/a
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. A new job can be scheduled	n/a
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. A new job can be scheduled.	n/a

Step 2: Selection of Records

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	At least a pair of ledger records were found to sweep.	n/a	n/a
Warning (4)	No ledger records were found that matched selection criteria.	Review selection criteria to see if they were correct. If so, no further action is required. If not correct, then correct the parameters and run	n/a

		another instance of the chain.	
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: 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.	n/a
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. A new job can be scheduled	n/a
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. A new job can be scheduled.	n/a

Step 3: Generate XML

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Job step was able to successfully create and save the XML file.	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: 1) Encounters any runtime exceptions and	n/a

		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.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. A new job can be scheduled.	n/a
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the system failure needs to be investigated. A new job can be scheduled.	n/a

Pre-Annual Close Sweep Chain: Sweep Transaction Load

Chain or Job Name	Sweep Transaction Load
Recommended Frequency	See chain
Single Instance Required	See chain
Can be restarted?	No
Reports generated	No

Overview

The Sweep Transaction Load job loads the records from the PACSDocuments.xml file according to instructions in the PACSLoadParams.txt file. This job uses the System Maintenance Utility (SMU) to load the records into the Transaction Catalog. This job first validates the batch parameters. If the parameters are valid, then it loads the records into the Transaction Catalog. If the parameters are not valid, the job issues appropriate messages and ends with a status of *Failed*. Once the records are loaded into the Transaction Catalog, the summary information is written into the log to display the number of records that were in the input file and the number of records loaded successfully.

Major Input

- PACSDocuments.xml
- PACSLoadParams.txt

Batch Parameters

Parameter	Description	Default Value
Load Parameter File	Required parameter, which	$$AMSPARM$$/PACSLoadParams.txt

(PARM_FILE)	gives the location and file name that contains the information of the XML file containing the JV transactions to be loaded.	
Commit Block Size (COMMIT_SIZE)	Required performance parameter that determines the number of transactions to be committed at a time.	100
Action Code (ACTN_CD)	Required code designating the Import Action for SMU.	171

Please refer to the SMU Transaction Upload Job run sheet in the *CGI Advantage Administration – Utilities Run Sheets* guide for the full list of SMU Transaction upload batch parameters which can be specified in the PACSLoadParams.txt file.

Major Output

- JV Transactions in draft version

Job Return Code

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

Return Code	Condition
Successful (1)	The SMU was able to run successfully.
Warning (4)	Job step does not end with this return code.
Non-Fatal Error (8)	Job step does not end with this return code.
Failed (12)	• Parameters are invalid. • When the input file is not found in the specified directory. • Restart failed because another instance of the PACS chain has already been run successfully. • Runtime exceptions encountered for any unexpected situations. When the job ends with a return code of *Failed*, subsequent jobs in the chain are set to *Inactive*.
Terminated (16)	This return code is issued when the job is terminated by the user. When 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 Sequence

None

Selection Criteria

None

Problem resolution

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Job step was able to successfully load the XML file.	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: 1) XML or TXT Files could not be found. 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.	n/a

Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. A new job can be scheduled.	n/a
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the system failure needs to be investigated. A new job can be scheduled.	n/a

Pre-Annual Close Sweep Chain: Sweep Transaction Submit

Chain or Job Name	Sweep Transaction Submit
Recommended Frequency	See chain
Single Instance Required	See chain
Can be restarted?	Yes
Reports generated	No

Overview

This job submits the transactions listed in the input parameter file originally generated by the Pre-Annual Close Sweep job, and performs a submit action on a listing of the transactions previously loaded.

Major Input

- PACSSubmitParams.txt

Batch Parameters

Parameter	Description	Default Value
Submit Parameter File (PARM_FILE)	Required parameter, which gives the location and file name that contains the information of the XML file containing the JV transactions to be submitted.	$$AMSPARM$$/PACSSubmitParams.txt

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

Major Output

- Journal Vouchers submitted to final with all system updates made.
- Exception report file (JVExcepRep)

Batch Return Codes

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

Return Code	Condition
Successful (1)	The SMU was able to run successfully.
Warning (4)	Job step does not end with this return code.
Non-Fatal Error (8)	Job step does not end with this return code.
Failed (12)	• Parameters are invalid • When the input file is not found in the specified directory • Restart failed because another instance of the PACS chain has already been run successfully • Runtime exceptions encountered for any unexpected situations When the job ends with a return code of *Failed*, subsequent jobs in the chain are set to *Inactive*.
Terminated (16)	This return code is issued when the job is terminated by the user. When 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 Sequence

None

Selection Criteria

None

Problem Resolution

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

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

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Job step was able to perform the submit action on all transactions loaded.	If any transactions failed to submit, the Exception Report ends *as Non-Fatal* and list the transactions in the report.	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. • XML or TXT Files could not be found. • Encounters any runtime exceptions and The job 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.	Reason for the termination needs to be investigated. A new job can be scheduled.	n/a
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the System Failure needs to be investigated. A new job can be scheduled.	n/a

Pre-Annual Close Sweep Chain: Generate Exception Report

Chain or Job Name	Generate Exception Report

Recommended Frequency	See chain
Single Instance Required	See chain
Can be restarted?	No
Reports generated	Yes

Overview

This job in the Pre-Annual Close Sweep chain generates an exception report that lists all of the errors encountered when the JV transactions were submitted in the earlier job step. This report lists those JV transactions created that did not reach the *Final* transaction phase. Upon investigation and a remedy for the error(s) performed, the transactions can be manually submitted or submitted through a new system maintenance utility step. The report can then be produced again so that it reads the same parameter file listing the transaction IDs loaded to determine if any are still not *Final*. Alternatively, the Transaction Catalog can be searched for rejected transactions created from the sweep process.

When at least one journal voucher has rejected, the report returns as *Non-Fatal Error*. If all transactions accepted, the report returns *Successful* and the report contains one line: No Transaction Errors.

Major Input

- JVExp.txt (Transaction errors from submit step)
- JV Transactions header table

Batch Parameters

Parameter	Description	Default Value
EXCEP_PARM_FILE_NM	Exception text file created with any transaction errors for report job step.	$$AMSROOT$$/ExportImport/JVExcepRep

Major Output

- Exception Report

Batch Return codes

The following table shows the potential job return codes for the individual Sweep Exception Report job in the JV submitter.

Return Code	Condition
Successful (1)	All journal vouchers submitted to *Final* transaction phase and the exception report was generated successfully.
Warning (4)	Job step does not end with this return code.
Non-Fatal Error (8)	At least one journal voucher failed to submit to *Final* transaction phase.

Failed (12)	• Parameters are invalid • When the input file is not found in the specified directory • Restart failed because another instance of the PACS chain has already been run successfully • Runtime exceptions encountered for any unexpected situations When the job ends with a return code of *Failed*, subsequent jobs in the chain are set to *Inactive*.
Terminated (16)	This return code is issued when the job is terminated by the user. When 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 Sequence

- Transaction ID

Selection Criteria

None

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Report was generated with no transaction errors.	n/a	n/a
Warning (4)	This step does not issue this return code.	n/a	n/a
Non-Fatal Error (8)	At least one journal voucher rejected.	Take the report listing and investigate why the transactions rejected and address the errors.	Submit transactions manually or through another SMU job. Restart the Exception Report to review is all transactions are accepted now.
Failed (12)	Job failed due to fatal conditions.	In this step, the job can fail under the following three conditions. • JVExp.txt file not	n/a

		found • 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.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. A new job can be scheduled.	n/a
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the System Failure needs to be investigated. A new job can be scheduled.	n/a

2.1.31 Rebuild Ledger

Job Name	Rebuild Ledger
Recommended Frequency	Frequency depends on need. The process can be run for any of the following reasons: • Table-driven options initially set are changed later leading to an incorrect or incomplete ledger. • An inactive ledger can be activated and needs to be retro-filled to include past data. • In an event of data corruption on a ledger, a part or the entire ledger needs to be rebuilt from the source journal. • The source journal to a ledger has been rebuilt.
Single Instance Required	Yes. A single instance is required for a ledger. Multiple instances are allowed if against different ledgers.
Can be Restarted?	No
Reports Generated	None

Overview

This process initiates a rebuild for the given ledger ID by performing three actions:

1. Delete existing ledger records
2. Delete existing Journal Ledger Cross Reference records
3. Create an instruction record on Journal Log for the Ledger Engine job to read and perform the actual rebuilding of the ledger.

The Rebuild Ledger job provides a means to rebuild a specific ledger from scratch or to catch up a ledger activated after initial accounting activity. When rebuilding a ledger, journal records not ledgerized yet will not be selected. Only those records successfully posted to other ledgers from the input journal will be selected. Thus, any Journal Log (JRNL_LOG) records for failed work and any journal records processed since the last successful ledger posting to all ledgers will not be selected. The next time the Ledger Engine is run to process failed work, gap, or in normal mode, those records skipped in the rebuild will then be ledgerized.

The Rebuild Ledger job can be run as either a full rebuild or a partial rebuild. Use of the Fiscal Year (FY) or Budget Fiscal Year (BFY) selection parameter indicates a partial rebuild that will only delete records matching the year and select only journal records for the rebuild that match the year. Partial rebuilds can be combined into multiple runs to effectively perform a full rebuild when a full rebuild would require more time than available in a single window by breaking down the rebuilding work into manageable units of work spread across different time windows. A partial ledger rebuild may also be preferable to a full rebuild when incorrect or missing data is known to be a problem for only a subset of years.

Please note that only ledgers summarizing on Fiscal Year or Budget Fiscal Year can be partially rebuilt. For example, an Inception-to-Date (ITD) ledger without Fiscal Year or Budget Fiscal Year cannot be partially rebuilt.

The Rebuild Ledger job will not allow another Rebuild Ledger instance to process for the same ledger until all Ledger Rebuild work is completed for the ongoing rebuild. For example, when partially rebuilding a ledger, it is not possible to have two sets of ongoing rebuild work for different

years within the same ledger. A partial rebuild for one Fiscal Year or Budget Fiscal Year must fully complete before another partial rebuild can be initiated for the same ledger.

Rebuilding a ledger is a two-step process starting with the Rebuild Ledger batch job and continuing with multiple instances of the Ledger Engine running at the same time to perform the rebuild.

> Step 1 - Run Rebuild Ledger when no instance of the Ledger Engine is running. This will do all the required record deletions and write an instruction record into the Journal Log (JLOG) defining the range of DOC_UNID values that will be ledgerized later by the Ledger Engine. For partial rebuilds, the FY or BFY parameter value is also retained in the instruction record for the Ledger Engine to use.
>
> Step 2 - Once the Rebuild Ledger has finished successfully, schedule one or more instances of Ledger Engine in the 'continue ledger rebuild mode' by entering the ledger ID as a parameter. Exactly how many instances is a function of system resources. The Ledger Engine run sheet should be consulted for more details. **The Rebuild Ledger process should not be run when any instance of Ledger Engine is running in any of the other modes (Rebuild, Process Gaps, Normal, or Process Failed Work), otherwise it would lead to indeterminate results.**
>
> A Ledger Engine run will read the instruction entry and ledgerize that range into the ledger being rebuilt. If a partial rebuild was specified, then during the Ledger Engine instance, only those records within that range which also satisfy the FY or BFY partial rebuild criteria will be selected for processing.

The partial rebuild where not all years of a ledger are eventually rebuilt must be very carefully considered if the ledger's summarization options are modified on Journal Ledger Control (JLCTRL). A partial ledger rebuild instigated after JCTRL summarization option changes can result in the situation where some ledger records (for some years) were created with summarization options that are no longer configured on JLCTRL. If this occurs, the system can no longer interpret the ledger's Concatenated Key fields of those records ledgerized with the old JLCTRL options. Any attempt to retrieve the ledger records not rebuilt (summarized based on old JLCTRL summarization options) based on the Concatenated Keys may fail or produce incorrect results. Similarly, any attempt to ledgerize new journal records *into* those old ledger records may produce incorrect results. Each of these activities would be extremely rare if they even happen at all for older years that are closed and no longer being used.

Important Run Instructions

Several important facts should be known before scheduling journal rebuild:

1. When rebuilding a previously inactive ledger, after checking the Active flag on the Journal Ledger Control (JLCTRL) page, each VLS must be bounced for the change to be registered.
2. Rebuilding ledgers (full or partial), may involve restoring archived journal data in order to rebuild the ledger with the same breadth of data, else the ledger will be rebuilt with only un-archived journal data. This is not likely a concern when rebuilding the current year and possibly a prior year that has not been closed as journal archiving will only be allowed for closed fiscal years. If the archived journal activity is not rebuilt into the ledger it is effectively the same as a ledger archive without the storage of those ledger records. For this reason, a site may wish to archive a ledger up to the same point at the source journal before rebuilding the ledger.
3. If performing partial rebuilds based on BFY, do not forget to have a run for BFY = 9999 if that value appears on accounting transactions.
4. There will likely be at least one interaction with another batch program that uses the rebuilt ledger. When a full ledger rebuild is done, the application will see that the first new record created in the ledger is record #1. A partial rebuild cannot do this because of duplicate

record number problems. Note: none of the following applies to a ledger that is being activated and brought up-to-date.

a. In the case of a ledger rebuilt because of a rebuilt journal, a batch program reading the ledger would not have had the chance to select activity previously missing from the ledger, or would have selected activity that should not have been in the ledger, or would not be impacted because the incorrect or changed option in the journal did not apply to the batch job reading the ledger. No batch program can go back through the ledger to pick up activity initially missed or create transactions to remove activity previously selected that is no longer in the rebuilt ledger.

b. The number of batch jobs that read an input ledger to create transactions from is far fewer than those that read journals. Those that do (that is, Annual Close, Pre-Annual Close Sweep, and Cost Allocation) do not track progress through JLOG and have other mechanisms for selecting a ledger record only once. An analysis will have to be done manually and a decision made about manually creating transactions for missed activity. For activity selected that should not have been, that too will have to be a manual analysis with a decision made about creating transactions to correct such selected records.

Annual Close (no JLOG) – The only step required for this chain would be 7b above. For most sites, the Accounting Journal will not be rebuilt for any fiscal year that is closed.

System Assurance 1 (SA01LL, SA01IJL, & SA01JL) – A full run of SA1 is required if the input ledger (most likely LDGR_SA_BUD) has been rebuilt. The full run should have parameters of a normal full run to populate SA_BUD and produce a report. Before that full run, normal performance measures such as the Journal Posting Initiator and the Journal Engine should be performed.

System Assurance 2 (SA02LL, SA02IJL, & SA02JL) – A full run of SA2 is required after the rebuild of the input ledger (most likely LDGR_SA_BUD or LDGR_FYDAD). The full run should have parameters of a normal full run SA_NONBUD and produce a report. Before that full run, normal performance measures such as the Journal Posting Initiator and the Journal Engine should be performed.

System Assurance 7 (SA7) – When a journal and all ledgers from that journal have been rebuilt, it is best to delete all JRNL_LOG records with PROC_ID of SA07. The next SA7 should be run in the same manner as before the rebuild. If run incrementally against all ledgers, this next run will take longer than normal to complete.

System Assurance 8 (SA8) – When a journal is rebuilt that is the source to one or more ledgers, it is best to delete all JRNL_LOG records with PROC_ID of SA08. The next SA8 should be run in the same manner as before the rebuild.

Cost Allocation – The evaluation in 7b above is one step for this batch job.

Rebuild Ledger Process

1. **Parameter Validation**: The first step of the job is to validate input parameters.
2. **Purging**: The second step starts with marking any pre-existing JLOG records for rebuilding of this ledger as historical.

   The step continues with the purging of Journal Ledger Cross Reference records followed by the purging of ledger records. If either of the selection batch parameters (BFY or FY) is provided then a partial rebuild is implied. Existing ledger records from the ledger satisfying the batch parameter selection criteria are purged. Journal/ledger Cross-reference records for those ledger records are also purged.

If all the selection batch parameters are left blank then full rebuild is implied in which case all records from the ledger are purged. All Journal/ledger Cross-reference records for the ledger are also purged.

Some have even looked to database tools to do this purging more efficiently, but extreme care should be taken with this option.

3. **Adding Instruction to Journal Log**: The last step of the job is to add a record to Journal Log (JRNL_LOG) with a Process ID (PROC_ID) of LDGRINSTRC with the starting and ending Transaction Unique ID values.

Process Steps	Messages
1. Parameter Validation	• Run Started • Parameters are listed with values • Validating Batch Parameters • If the parameter is invalid, an error message will be displayed • Batch Parameter validation completed
2. Purging	• Processing started in Ledger Rebuild Mode • Marking previous JRNL_LOG entries as Historical for Ledger ID ## started • Marking previous JRNL_LOG entries as Historical for Ledger ID ## completed • Purging records from Journal Ledger Cross Reference Table for Ledger ID ## started • Purging records from Journal Ledger Cross Reference Table for Ledger ID ## completed • Purging records from Ledger ID ## started • Purging records from Ledger ID ## completed The ## above is replaced in the Job Log with the ID of the ledger rebuilt.
3. Adding Instruction Journal Log	• Rebuild Instruction for Ledger ID ## added to JRNL_LOG successfully • Run Ended The ## above is replaced in the Job Log with the ID of the ledger rebuilt.

For example, the JRNL_LOG has the following records before invoking the Rebuild Ledger with **Rebuild Ledger ID = 26**. Rebuild Ledger 26 summarizes journal 1.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC _UNID	END_DOC _UNID
1	LDGRPOST	1	Final	100	300
2	LDGRPOST	1	Gap	301	400
3	LDGRPOST	1	Failed	401	500

4	LDGRPOST	1	Final	501	600
6	LDGRPOST	2	Failed	901	1000

The job adds a rebuild instruction record, denoted by PROC_ID = LDGRINSTRC. The JRNL_ID contain the ledger's ID. The status is 'Intended' and retains this status until all rebuild work is completed. The range of DOC_UNIDs is the range of all already-processed DOC_UNIDs for the journal upon which the ledger summarizes. It is the range of DOC_UNIDs already processed in Normal mode. Transaction 600 is the last attempted transaction ledgerized for JRNL_ID = 1 and is set as END_DOC_UNID for the Instructions entry.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC _UNID	END_DOC _UNID
1	LDGRPOST	1	Final	100	300
2	LDGRPOST	1	Gap	301	400
3	LDGRPOST	1	Failed	401	500
4	LDGRPOST	1	Final	501	600
6	LDGRPOST	2	Failed	100	300
7	LDGRINSTRC	26	Intended	100	600

If the process has already been run for the specified Ledger Id then this process will mark the status as 'Historical' as explained below.

Starting a Rebuild Again

If the Rebuilding Ledger is invoked a second or a third or multiple times for the same ledger, all previous rebuild work is marked as 'Historical'.

For example, after the first set of rebuild invocations, the JRNL_LOG contains these records:

- UNID 7 was added by a previous "Rebuild Ledger" for Ledger Id 26.
- UNIDs 8 and 9 were added by Ledger Engine running in Continue Rebuild Ledger mode.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC _UNID	END_DOC _UNID
1	LDGRPOST	1	Final	100	300
2	LDGRPOST	1	Gap	301	400
3	LDGRPOST	1	Failed	401	500
4	LDGRPOST	1	Final	501	600
6	LDGRPOST	2	Failed	100	300
7	LDGRINSTRC	26	Intended	100	600
8	LDGRRBLD	26	Final	100	300
9	LDGRRBLD	26	Failed	301	400

Schedule the Rebuild Ledger process for Ledger 26 again. First, all previous rebuild work for ledger 26, (UNIDs 7, 8 and 9) is marked as historical. Then, a **new** instructions record is posted with UNID 10.

UNID	PROC_ID	JRNL_ID	ST_FL (Status)	BGN_DOC_UNID	END_DOC_UNID
1	LDGRPOST	1	Final	100	300
2	LDGRPOST	1	Gap	301	400
3	LDGRPOST	1	Failed	401	500
4	LDGRPOST	1	Final	501	600
6	LDGRPOST	2	Failed	100	300
7	LDGRINSTRC	26	Historical	100	600
8	LDGRRBLD	26	Historical	100	300
9	LDGRRBLD	26	Historical	301	400
10	LDGRINSTRC	26	Intended	100	600

Major Input

- Journal Log (JRNL_LOG)
- Source Journal (JRNL_XYZ)

Batch Parameters

Parameter	Description	Default Value
Rebuild Ledger ID	Required Ledger ID of the ledger to be rebuilt. See the Journal/Ledger Control table for valid values. If performing a partial rebuild, ledgers without the FY or BFY to match the selection parameter are not allowed.	(blank)
Start Rebuilding Ledger	Required parameter to signal the Ledger Engine to start a rebuild instead of continue one. Protected parameter that must be Y for the Rebuild Ledger job to work properly.	Y
BFY_PARM	Optional selection parameter for a partial rebuild. Enter a budget fiscal year as CCYY. Only 1 value is allowed and cannot be combined with Fiscal Year parameter.	(blank)

FY_PARM	Optional selection parameter for a partial rebuild. Enter a fiscal year as CCYY. Only 1 value is allowed and cannot be combined with Budget Fiscal Year parameter.	(blank)
Delete Block Size	Required. This parameter determines the number of records to be deleted in one block. The value should not be too low as it would degrade performance with multiple deletes of a small number of records. The value should not be too high as that can result in an error when deleting a high volume of records. The parameter value must be a positive integer. If the supplied value is not a positive non-zero integer value, all journal records will be deleted in one transaction as the program will consider such a value as no block size. If the parameter value is left blank, the system will default a value of 150000.	150000

Major Output

- Journal Log (JRNL_LOG)
- Journal Ledger Cross Reference (JRNL_LDGR_XREF)
- Ledger (LDGR_XYZ) where XYZ is decided by batch parameter Ledger ID

Job Return code

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

Return Code	Condition
Successful (1)	The job will be Successful under the following conditions: • Parameters are valid • Purging successful • The Instruction record is added successfully in the Journal Log
Warning (4)	The job does not end with this return code.
Non-Fatal Error (8)	The job does not end with 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 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

Not applicable

Selection Criteria

Historical Journal Logs:

1. Select records from JRNL_LOG where PROC_ID is LDGRINSTRC or LDGRRBLD and
2. JRNL_ID is the Ledger Rebuild parameter.

Ledger Records for Deletion:

1. Budget Fiscal Year is batch parameter supplied (BFY = CCYY) and
2. Fiscal Year is batch parameter supplied (FY_DC = CCYY)

   Note: criteria 1 and 2 are optional and cannot be used in conjunction with each other

Journal Ledger Cross Reference Records for Deletion:

- Ledger ID (LDGR_ID) equals that of batch parameter and
- Ledger Record Number (LDGR_REC_NO) equals a record number purged from ledger before rebuilding

Journal Log Selection for Instruction:

- Select the first and last records JRNL_LOG with a PROC_ID of LDGRPOST and
- Journal ID (JRNL_ID) is equal to the Source Journal for the Rebuild Ledger ID.
- The BGN_DOC_UNID for the first record along with the END_DOC_UNID from the last record becomes the range for the instruction record.

Problem Resolution

- It is a good practice to look at the log of the job even if the job has run successfully.
- If the job fails for any data setup reasons then submit a new instance of the job with correct setup.

Parameter Validation

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	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)	Beginning Run time is not in proper format. **Sample Message:** Invalid Beginning run time. Enter Beginning run time in mm/dd/yyyy hh:mm:ss format	Run a new instance of the engine with correct parameters.	N/A
Terminated (16)	Job is terminated manually by the user	Reason for the termination needs to be investigated. Reschedule a new job after reason for termination is resolved.	N/A

2.1.32 Receiver Accrual

Job Name	Receiver Accrual
Recommended Frequency	On Demand
Single Instance Required	Yes
Can be restarted?	N/A
Reports generated	Yes, the last job in the chain generates a report.

Overview

The Receiver Accrual is a chain job in CGI Advantage Financial and is available under the Financial/General Accounting/Chain Job folder.

The Receiver Accrual Chain job determines the Purchase Order (PO) Commody Lines (CL) that have Receivers against them but have not been paid. The process accrues the amount of the PO that is received but not paid and generates a Journal Voucher (JV) for the accrued amount.

To record accruals in the system, the process creates Journal Voucher (JV) type entries. JV transaction type entries are posted to record the expenditure accruals and encumbrance liquidation in the 13th period of the prior year as well as to record the expenditure accrual reversals and encumbrance establishment in the 1st period of the new fiscal year. The accrual amount, posting codes, and COA codes are automatically populated on the JV's accounting lines by the process. The JV transaction has an associated memo reference to the Purchase Order to support reporting.

PO type transactions (that are final with a Budget Fiscal Year (BFY) less than the beginning Fiscal Year (FY) value specified in the parameter) are selected by the chain process. The process extracts POs that have a greater amount of Receivers than Payments against it. For each PO, CL, and associated Accounting Lines (AL) that fit the criterion, a credit and debit accounting entry is created for each of the following: previous year payment, subsequent year payment reversed, previous year encumbrance liquidation, and subsequent year encumbrance establishment.

This chain has the following jobs:

- Receiver Accrual
- JV Load and Submit
- JV System Assurance

Sample postings are listed below (provided that the Accrual Amount is $500):

JV transaction		
BFY 2008; AFY 2008; APD 13	**DR**	**CR**
Cash Expense (D014)	$500	
Other Liability		$500
** Creating Expenditure Accrual.*		
Reserve for Encumbrance (P006)	$500	
Encumbrance		$500
** Liquidating original encumbrance for accrual amount*		
BFY 2008; AFY 2009; APD 1	**DR**	**CR**
Other Liability (A15X)	$500	

Cash Expense		$500
** Creating Reversal for Expenditure Accrual.*		
Encumbrance (P005)	$500	
Reserve for Encumbrance (P006)		$500
** Establishing Encumbrance in the new fiscal year.*		

Major Input

- Purchase Order Transaction Header Table (PO_DOC_HDR)
- Purchase Order Transaction Vendor Table (PO_DOC_VEND)
- Purchase Order Transaction Commodity Table (PO_DOC_COMM)
- Purchase Order Transaction Accounting Table (PO_DOC_ACTG)
- Receiver Transaction Header Table (RC_DOC_HDR)
- Receiver Transaction Vendor Table (RC_DOC_VEND)
- Receiver Transaction Commodity Table (RC_DOC_COMM)
- Receiver Transaction Accounting Table (RC_DOC_ACTG)
- Tax Profile Maintenance Line table
- Tax Type table (R_TAX_TYP_MAINT_Generic)
- Special Accounts

Major Output

- JV Transactions are submitted in Final Phase.
- System Assurance report is generated with rejected JV Transactions.

Chain Return Codes

The following table shows the potential return codes for the PO Batch Print chain job. Note that the Chain job ends with the highest return code across all of the jobs.

Return Code	Condition
Successful (1)	All of the jobs end successfully.
Warning (4)	One of the jobs in the chain ends with a return code of *Warning*.
Non-Fatal Error (8)	One of the jobs in the chain ends with a return code of *Non Fatal Error*.
Failed (12)	The job fails under the following conditions: • 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.

Problem Resolution

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

If the job fails for any batch parameter validation errors or data setup reasons, then correct the data setup or the batch parameter values and schedule a new job.

Receiver Accrual Chain: Receiver Accrual

Job Name	Receiver Accrual
Recommended Frequency	This process is run on demand as part of the Chain Job.
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	No

Overview

The Receiver Accrual Chain job reads the Purchase Order (PO) and Receiver (RC) transaction tables to identify accruals (open encumbrances with receivers that are not paid).This job step consists of following basic steps:

1. **Parameter Validation:** If the parameter validation is successful, the job goes to the next step. Otherwise, the job ends with a return code of *Failed*.
2. **Selection of Records:** Records from PO and RC are selected based on batch parameters.
3. **Processing of Records:** Selected PO and RC records are processed and JV transactions are created.
4. **Generate XML:** An XML file is generated with JV transactions.
5. **Files for Later Job Steps:** Files are created for later load and submit steps.

Process Steps	Messages
1. Parameter Validation	• Run Started • Each parameter is listed with value. • Any parameter errors are listed.
2. Selection of Records	• In this step, the job selects records based on the selection criteria specified by the user as batch parameters. • {Selecting eligible records • If the selection returns 0 records, then the following

	message is reissued: “No eligible record found”. • Number of records (count) selected are displayed. • At the end, the following message is issued: “Selection of records completed.”}
3. Processing of Records	• PO records processed: ## (where ## is the count of PO records found matching accrual criteria)
4. Generate XML	• No messages issued for step
5. Files for Later Job Steps	• No messages issued for step • Run Ended

Major Input

- Purchase Order Transaction Header Table (PO_DOC_HDR)
- Purchase Order Transaction Vendor Table (PO_DOC_VEND)
- Purchase Order Transaction Commodity Table (PO_DOC_COMM)
- Purchase Order Transaction Accounting Table (PO_DOC_ACTG)
- Receiver Transaction Header Table (RC_DOC_HDR)
- Receiver Transaction Vendor Table (RC_DOC_VEND)
- Receiver Transaction Commodity Table (RC_DOC_COMM)
- Receiver Transaction Accounting Table (RC_DOC_ACTG)
- Tax Profile Maintenance Line table
- Tax Type table (R_TAX_TYP_MAINT_Generic)
- Special Accounts

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
Closing Fiscal Year (CLOSING_FY)	A required parameter to specify the Closing Fiscal Year.	(No Default)
Beginning Fiscal Year (BEGIN_FY)	A required parameter to specify the New Fiscal Year.	(No Default)
Closing Period (CLOSING_PERIOD)	A required parameter to specify the Closing Period used for Accrual posting lines.	(No Default)

Beginning Period (BEGIN_PERIOD)	A required parameter to specify the Beginning period used for Accrual Reversal posting lines.	(No Default)
PO Accrual Transaction Code (ACCRUAL_DOC_CD)	A required parameter. Enter a Transaction Code to be assigned to the transaction created by the batch output.	(No Default)
Export/Import Location (AMSEXPORT)	The required location where the XML file is written.	$$AMSROOT$$/ExportImport
Log Location (AMSLOGS)	A required parameter to specify the location for the log file.	(No Default)
Parameter Location (AMSPARM)	A required parameter to specify the location for the parameter file.	(No Default)
File Name (PARM_GEN_FILE_NM)	A required parameter to specify the location for the file used in exporting the data to xml.	(No Default)
System Assurance File Name (EXCEP_FILE)	A required parameter to specify the name of the SA file.	(No Default)
Encumbrance Doc Codes (ENCUMB_DOC_CD)	A required parameter to specify the transaction code(s) used in the selection criteria.	(No Default)
Posting Code – Cash Expense (PSTNG_CD_ACCR_EXP)	A required parameter to provide the Posting Code for the Cash Expense PSCD field to allow a user to specify the PSCD.	(No Default)
Posting Code – Other Liability (PSTNG_CD_OTHER_LIAB)	A required parameter to provide the Posting Code for the Other Liability PSCD field to allow a user to specify the PSCD.	(No Default)
Posting Code – Reserve for Encumbrance (PSTNG_CD_RES_ENCUMB)	A required parameter to provide the Posting Code for the Reserve for Encumbrance PSCD field to allow a user to specify the PSCD.	(No Default)
Posting Code – Encumbrance (PSTNG_CD_ENCUMB)	A required parameter to provide the Posting	(No Default)

	Code for the Encumbrance PSCD field to allow a user to specify the PSCD.	
Commit Block Size (BLOCK_SIZE)	A required performance parameter that determines the number of transactions to be committed at a time.	(No Default)
Maximum Accounting Lines (MAX_ACTG_LINES)	A required parameter to provide the maximum accounting lines per transaction. This parameter specify the maximum number of accounting lines that are generated on the JV transaction.	(No Default)

Major Output

- Journal Voucher (JV) transactions XML file.

Sort Criteria

None

Selection Criteria

The basic selection criteria is as follows:

- PO Header table
  - Select records that meet the following criteria:
    a. Transaction Phase is equal to ‘Final’
    b. Transaction Function is not equal to ‘Cancellation’
    c. BFY equals Closing FY (as specified in parameter)
- Of those selected, read the PO transaction Commodity Line table
  - Select records that meet the following criteria:
    a. RC Qty is greater than Paid Qty
    b. Paid Final Flag is equal to ‘False’
- For each record found, look up the associated RC transactions
  - RC transactions must meet the following criteria:
    a. Transaction Phase is equal to ‘Final’
    b. Transaction Function is not equal to ‘Cancellation’

- If Received SC Amount (on RC Transaction Commodity table) is not equal to zero:
  a. Aggregate the Received SC Amounts for each PO CL where the receipt date (on RC Transaction Header table) is on or before June 30th.
     - Select those records where the aggregated Received SC Amount (from RC Transaction Commodity table) is greater than the Paid Amount (from PO Transaction Commodity table).
  b. Of those records selected, perform the following:
     - *Calculate Pre-Tax CL Accrual Amount*: Aggregated Received SC Amount minus Paid Amount
- If Received SC Amount (on RC Transaction Commodity table) is equal to zero:
  a. Aggregate the Received Quantity for each PO CL where the receipt date (on RC Transaction Header table) is on or before June 30th.
     - Select those records where the aggregated RC Quantity (from RC Transaction Commodity table) is greater than the Paid Quantity (from PO Transaction Commodity table).
  b. Of those records selected, perform the following:
     - *Calculate Accrual Quantity*: Aggregated RC Quantity minus Paid Quantity.
     - *Calculate Pre-Tax CL Accrual Amount*: Accrual Quantity multiplied by PO Unit Price.

- For each Pre-Tax CL Accrual Amount calculated, perform the following calculations:
  - Lookup Tax Type(s) (on Tax Type table) using the PO CL Tax Profile (from PO Transaction Commodity Line table).
  - *Calculate CL Accrual Tax Amount*: Pre-Tax Accrual Amount multiplied by Tax Percentage(s) (from Tax Type table).
  - *Calculate Total CL Accrual Amount*: Pre-Tax Accrual Amount plus Accrual Tax Amount.
- For each PO CL selected to accrue, look up the associated Accounting Lines (AL) on the PO Transaction Accounting Line table. The batch job calculates the proportion of each Accounting Line Open Amount in relation to its Commodity Line Open Amount. The Accrual Amount is then allocated to each Accounting Line according to this proportion.
  - *Calculate AL Percentage*: AL Open Amount (from AL table: AL Total Amount minus AL Closed Amount) divided by CL Open Amount (from CL table: CL Total Amount minus CL Closed Amount).
  - *Calculate Accrual Amount for each AL:* AL Percentage multiplied by Total CL Accrual Amount.
- Create one JV transaction to record the accrual accounting entries for all PO accounting lines for one PO transaction. The chart of account information for accrued accounting lines is brought forward to the JV transaction. The JV accounting lines have a memo reference to the PO accounting line that was accrued.

Problem Resolution

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

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	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:** The Transaction Code Parameter cannot be empty	Correct parameter error in a subsequent run of the chain.	N/A
	Failed because of runtime exceptions for an unexpected situation.	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before another instance of the chain is started.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the termination needs to be investigated before another instance of the chain is started.	N/A

Step 2: Selection of records

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Records selected are processed successfully	N/A	N/A
Warning (4)	No PO and RC records selected for processing	Verify selection criteria are correct and if not, then submit another instance. This may be a valid outcome if selection criteria were correct.	If selection criteria were wrong, the PO and RC progress tracking record need to be reset if incremented so selection can read that range of journal records again.

Non-Fatal Error (8)	This step does not issue this return code.	N/A	N/A
Failed (12)	In this step, the job can fail under the following conditions. 1) Encounters any runtime exceptions	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Step 3: Processing of records

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All records are successfully processed.	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)	In this step, the job can fail under the following condition. 1) Encounters any runtime exceptions	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Step 4: Generate XML

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	XML file was successfully generated.	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)	In this step, the job can fail under the following conditions. 3) Issue creating XML file, for example, Import/Export file location not found. 4) Encounters any runtime exceptions	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the failure needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Step 5: Files for Later Job Steps

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Files were successfully generated.	N/A	N/A
Warning (4)	This step does not issue this return code.	N/A	N/A
Non-Fatal Error (8)	This step doesn't issue this return code.	N/A	N/A
Failed (12)	In this step, the job can fail under the following conditions. 3) Issue creating parameter files, for	The reason for the failure needs to be investigated before starting another instance of the chain.	N/A

	example, parameter file location not found. 4) Encounters any runtime exceptions		
Terminated (16)	Job is terminated manually by the user.	The reason for the failure needs to be investigated before starting another instance of the chain.	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 starting another instance of the chain.	N/A

Receiver Accrual Chain: JV Load and Submit

Job Name	JV Load and Submit
Recommended Frequency	This process is run on demand as part of the Chain Job.
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	No

Overview

After the generation of the JV transaction xml file, the next step of the Receiver Accrual Process chain is the JV Load and Submit. This step uses the MultiProcessImport job with parameter as JV Transaction XML file name.

- The JV transaction processor updates the Accrual table.
- For each accrued PO accounting line, the JV transaction inserts one record to the Accrual table. Each record should be populated with the Current FY, Referenced PO Transaction Code, Transaction Department Code, Transaction ID, Vendor Line, Commodity Line, Accounting Line, Accrual Amount, and Accrual Date.
- The JV transactions are selected from the ACCA Transaction XML file and loaded/submitted into the application by using System Maintenance Utility (SMU).

Major Input

- JV Transaction.XML

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
Number of jobs to start (THREAD_COUNT)	Parameter defines the number of jobs to start processing.	(No Default)
Commit block size (BLOCK_SIZE)	(Required) Number of records to commit at a time.	(No Default)
Input XML File Name (FILE_LIST)	(Required) File name of the JV Transaction.xml	(No Default)
File Prefix for Parameter and Data Files (FILE_PREFIX)	(Required) Prefix used on the filenames for the output file segments.	(No Default)
Apply Overrides (I_SMU_APPLY_OVERRIDES)	(Required) If overrides are to be applied to transactions as they are loaded, this parameter should be true. A setting of false result in transactions rejecting for override errors.	(No Default)
Bypass Approvals (I_SMU_BYPASS_APPROVAL)	Required indication if approvals should be bypassed in the transaction loaded. (TRUE, FALSE).	(No Default)
SysManUtil input parameter – Bypass ADNT (I_SMU_BYPS_ADNT_FL)	(Required) Bypass ADNT Flag	(No Default)
Commit block size while importing (I_SMU_COMMIT_BLOCK_SIZE)	(Required) Number of records to commit at a time while importing.	(No Default)
SysManUtil input parameter – Doc Phase (I_SMU_DOC_PHASE_CD)	(Required) Transaction Phase for Loaded Transactions 1-Draft, 2-Final	(No Default)
SysManUtil input parameter – Doc Status (I_SMU_DOC_STA_CD)	(Required) Transaction Status for Loaded Transactions 1-Held, 2-Ready	(No Default)
Mode (MODE)	(Required) Mode of operation. (1=Import, 2=Import and Submit, 3=Import and Other Action)	(No Default)
Custom Action Code if Mode = 3 (OTHER_ACTION)	(Required) Custom Action code.	(No Default)
SysManUtil other parameter – Doc Phase (O_SMU_DOC_PHASE_CD)	(Required) Transaction Phase for Loaded Transactions 1-Draft, 2-Final	(No Default)

SysManUtil other parameter – Doc Status (O_SMU_DOC_STA_CD)	(Required) Transaction Status for Loaded Transactions 1-Held, 2-Ready	(No Default)
Stagger Time (in seconds) (STAGGER_TIME)	(Required) The lag time, in seconds, between the spawning of each child process.	(No Default)
SysManUtil submit parameter – Doc Phase (S_SMU_DOC_PHASE_CD)	(Required) Transaction Phase for Loaded Transactions 1-Draft, 2-Final	(No Default)
SysManUtil submit parameter – Doc Status (S_SMU_DOC_STA_CD)	(Required) Transaction Status for Loaded Transactions 1-Held, 2-Ready	(No Default)
Exception Report File Name (S_SMU_EXCEP_REP_FILE_NM)	(Required) Exception File Name	(No Default)
Exception Report Indicator (S_SMU_EXCEP_REP_IND)	(Required) Exception Detailing (1- Detailed, 2-Failed Transactions, 3-Processed Transactions, 4-Failed Transaction Lines, 5-Transaction Status)	(No Default)
Severity Flag (S_SMU_EXP_SEV_FL)	(Optional) System Maintenance Utility flag to indicate the Severity Flag in Submit Mode.	(No Default)

Major Output

- Journal Voucher (JV) transactions in Final version

Sort Criteria

None.

Selection Criteria

None

Problem Resolution

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All transactions were loaded and submitted.	N/A	N/A
Warning (4)	At least, 1 transaction didn't load.	The reason for the failure needs to be investigated before restarting the job. The transaction(s) that did not load, are in the Import/Export Error directory.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.
Non-Fatal Error (8)	No transactions were able to load	The reason for the failure needs to be investigated before restarting the job.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.
Failed (12)	In this step, the job can fail under the following conditions. 5) Issues in spawning jobs 6) Files not found 7) Runtime exceptions 8) Parameter Edits	The reason for the failure needs to be investigated before restarting the job.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.
Terminated (16)	Job is terminated manually by the user.	The reason for the failure needs to be investigated before restarting the job.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the system failure needs to be investigated before restarting the job.	If another instance of the chain has run, then the XML file has to be loaded with a separate SMU instance.

Receiver Accrual Process Chain: JV System Assurance

Job Name	JV System Assurance
Recommended Frequency	See chain
Single Instance Required	Yes
Can be restarted?	No
Report Generated	Yes

Overview

This job generates a System Assurance report for the rejected JV transactions. A system assurance (SA) text file is created by the PO Accrual Chain job if any of the JV transactions fail to submit. This file is available from the file location specified in the parameter.

The following data is included in each record:

- Transaction Code
- Transaction Department
- Transaction ID
- Status (Rejected)

The error message details are not provided in the SA file. This is consistent with how other jobs generate SA files and additionally, errors on JV transactions are very rare. If an error occurs, the user may use the transaction information provided in the file and search for the rejected JV in the Transaction Search page to review and fix errors.

Major Input

- JV_EXCP.txt
- JV Transactions

Batch Parameters

Parameter	Description	Default Value
Accrual Transaction Code (ACCRUAL_DOC_CD)	(Required) Accrual Transaction Code Example: JV	171
Client Name (PARM_CLIENT_NM)	(Optional) Client Name for the report header	(No Default)
Export/Import Location (AMSEXPORT)	(Required) AMS Export Import Directory	$$AMSROOT$$/ExportImport
Log Location (AMSLOGS)	Log Location at System Assurance Report	$$AMSROOT$$/Logs
Parameter Location (AMSPARM)	Parameter Location at JV System Assurance Job	$$AMSROOT$$/Parm

Parameter	Description	Default Value
System Assurance File Name (EXCEP_FILE)	(Required) System Assurance File Name that got generated by SMU	$$AMSEXPORT $$/ POJVSA.txt

Major Output

- JV System Assurance Report

Batch Return codes

The following table shows the potential job return codes for the individual JV System Assurance job.

Return Code	Condition
Successful (1)	System Assurance report was not generated. All the JV transactions were submitted successfully.
Warning (4)	Exception report was created with failed JV transactions.
Non-Fatal Error (8)	Not Applicable for this job.
Failed (12)	Input SMU log file was not found.
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. When this job ends with a return code of System Failure.

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All transactions are final	N/A	N/A
Warning (4)	One or more JV transactions rejected	Address the errors and submit the transactions either manually or with an SMU.	N/A

Non-Fatal Error (8)	The job does not end with this return code	N/A	N/A
Failed (12)	In this step, the job can fail under the following conditions. 4) System Assurance file not found 5) Runtime exceptions 6) Parameter Edits	The reason for the failure needs to be investigated before restarting the job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the failure needs to be investigated before restarting 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 before restarting the job.	N/A

2.1.33 Revolving Fund Replenishment

Chain or Job Name	Revolving Fund Replenishment
Recommended Frequency	On demand or regularly scheduled
Single Instance Required	Yes
Can be restarted?	See individual job steps
Reports generated	No

Overview

The Revolving Fund Replenishment process queries the Accounting Journal to create Appropriation and Allotment Budget (BGAA) transactions to replenish revolving funds for revenues recorded since the last run. The intended use of this job is when revenues are not tracked at Appropriation so that a budget line cannot have revenues contribute to available spending limits, those revenues can be identified and used to increase Current Budget for spending purposes. For if revenues were recorded at Appropriation, a line-level budget control could be used to add revenues to current budget.

The Revolving Fund Replenishment process consists of the following jobs (each of the jobs listed below, is described in the subsequent sections):

1. Revolving Fund
2. Load Transactions
3. Submit Transactions

Major Input

- Accounting Journal (JACTG - JRNL_ACTG)

Major Output

- XML file of BGAA transactions
- BGAA transactions

Problem Resolution

- The process has restart ability. If the job fails for any data setup reasons then correct the data setup and restart the job. If the first job has finished then set the Last Record Number back to the appropriate number.

Revolving Fund Replenishment: Revolving Fund

Job Name	Revolving Fund
Recommended Frequency	Run as part of the chain

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

Overview

The Revolving Fund Replenishment process searches the Accounting Journal for revenue transactions of revolving funds and generates Appropriation and Allotment Budget (BGAA) transaction to re-appropriate the money into the funds. The batch process generates one BGAA for each unique Fund and Department combination.

Major Input

- Accounting Journal (JRNL_ACTG)

Batch Parameters

Parameter	Description	Default Value
Export Location (AMSEXPORT)	A required location to write the XML file.	$$AMSEXPORT$$
Logs Location (AMSLOGS)	A required location to write the log file.	$$AMSLOGS$$
Parameter Location (AMSPARM)	A required location to write out a parameter file that is passed from the first job step to later job steps.	$$AMSPARM$$
Appropriation (APPR_CD)	A required output parameter for an appropriation for the budget transaction. Enter one value only.	(no default)
Bypass Approval (BYPASS_APP)	A required processing parameter that indicates the Bypass Approval action should be applied or not (*true* or *false*).	True
Department (DEPT_CD)	A required selection and output parameter for a department that is both the transaction department and budget line department. Enter one value only.	(no default)
Prefix (DOCID_PRFX)	A required output parameter for the creation of a structured transaction ID. Automatic numbering of generated budget transactions: Fiscal Year of System Date in YY format + REVFD + Fund parameter value + System Date in MMDDYY format. It should not be more than 5 characters long if individual funds passed are 3 characters long. It should not be more than 4 characters long If	REVFD

	any of the individual Funds passed are 4 characters long. Prefix + Fund should not exceed 8 characters.	
Transaction Code Filter (DOC_CD_LIST)	An optional parameter used to limit selection to one or more transaction codes. Separate multiple values with commas.	(no default)
Event Type (EVNT_TYP_CD)	The required output parameter of an event type for the budget update.	BG03
Fund (FUND_CD)	A required selection and output parameter for one or more funds. Separate multiple values with commas.	(no default)
Last Record Number (LAST_REC_NO)	Non-Editable field that is system generated at the end of each batch run to reflect the last record number on the accounting journal that was processed for the next instance to start after that record.	(no default)
Posting Code Closing Classification (PSCD_CLOS_CL_CD)	A required selection parameter of one or more posting code closing classifications that is collected earned revenue. Separate multiple values with commas.	14
Revenue Source (RSRC_CD)	A required selection parameter of one or more revenue sources. Separate multiple values with commas.	(no default)
Update Record Number (UPDATE_REC_NO)	A required parameter to indicate that the last record number should be updated in confirmation of completion of processing against a block of records. This parameter facilitates processing for sites that run this job multiple times with variations of selection parameters against the same block of journal records. The Update Last Record parameter should be set to *N*, if multiple jobs with variations in other selection criteria are being run against the same block of accounting records. The Update Last Record number parameter should be set to *Y*, after all records within the block have been processed. When run with this parameter set to *Y*, when the process completes, the LAST_REC_NO will be updated on the job for use with the next block of transactions when the process is run again.	Y

Major Output

- BGAA transactions (BG_DOC_HDR, BG_DOC_LN)
- If run with UPDATE_REC_NO = Y, LAST_REC_NO is updated in BATSETUP for the job to be used in the next run.

Job Return Codes

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

Return Code	Condition
Successful (1)	All parameters were valid and at least 1 journal record was selected.
Warning (4)	No journal records found for selection because no new records exist in the journal since the last run or no records met the selection criteria.
Non-Fatal Error (8)	Not Applicable for this job.
Failed (12)	The job fails under the following conditions: • Batch parameters are invalid When this job ends with a return code of Failed, subsequent jobs in the chain is 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 Sequence

None

Selection Criteria

Records are selected from JRNL_ACTG where:

- Record Number > Last Record Number Parameter
- Fund = Fund Parameter
- Department = Department Parameter
- Revenue = Revenue Source Parameter
- Posting Code's closing classification = Posting Code Closing Parameter

Problem Resolution

- Look into job log for errors.
- The process can be restarted after parameter validation has successfully completed.
- If the job fails for any data setup reasons then correct the data setup and restart the job.
- If the first job has finished, and was run with UPDATE_REC_NO = Y manual update to the BATSETUP of the job will be required to allow reprocessing. In the failed process, identify the UPDATE_REC_NO previously used and update the UPDATE_REC_NO via BATSETUP to reset to the previous last journal record processed.

Revolving Fund Replenishment: Load Transactions

Job Name	Load Transactions
Recommended Frequency	Run as part of the chain
Parallel processing enabled	No. Parallel processing is not supported for this job.
Can the job be restarted?	Optionally, based on the Save Restart Information parameter.
Exception report produced	No. All of the exceptions are only written to the job log.

Overview

The Revolving Fund Replenishment Load job loads the records from the Transaction XML file. The job first validates the batch parameters. If the parameters are valid, then it loads the records. Once the records are loaded into the Transaction Catalog, the summary information is written into the log to display the number of records that were in the input file and the number of records loaded successfully.

Major Input

- Import Parameter file (LoadRevolvingFundReplenishment.txt)

Batch Parameters

Parameter	Description	Default Value
Parameter file (PARM_FILE)	Specify the file to which Revolving Fund writes parameters for the Load Transactions job.	$$AMSPARM$$/LoadRevolvingFundReplenishment.txt

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

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

Major Output

- BGAA transactions in draft

Job Return Codes

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

Return Code	Condition
Successful (1)	All of the records are loaded into the Transaction Catalog successfully or the input file is empty.
Warning (4)	This return code is issued when some of the records failed to load whereas all other records were loaded successfully.
Non-Fatal Error (8)	This job does not issue this return code
Failed (12)	• Parameters are invalid • The input file is not found in the specified directory • Restart failed because another instance of the Clearing Maintenance chain has already been run successfully • Runtime exceptions encountered for any unexpected situations When the job ends with a return code of Failed, the subsequent jobs in the chain is set to inactive.
Terminated (16)	This return code is issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain are set to inactive.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain are set to inactive.

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Warning (4)	None or not all of the transactions successfully loaded from the XML file.	Corrections to that XML file or within the application should be made so that transactions can be loaded. Then the load	

		job can be restarted or the failed transactions can be loaded with a separate SMU, reading the Import/Export Error file created by the failed load.	
Failed (12)	The job failed while restarting the job since another instance of the job has already been run successfully. **Sample Message**: Cannot restart the job since another instance of this job has already been run successfully. Invalid job parameter	Wait until the first instance has been completed. Fix the invalid job parameter.	

Revolving Fund Replenishment: Submit Transactions

Job Name	Submit Transactions
Recommended Frequency	Run as part of the chain
Single Instance Required	No
Can be restarted?	Yes
Report Generated	No

Overview

This job submits the transactions listed in the input parameter file originally generated by the Load Transactions job.

Major Input

- SMU job parameter file (SubmitRevolvingFundReplenishment.txt)

Batch Parameters

Parameter	Description	Default Value

PARM_FILE	Non-Editable field used to specify the file to which Revolving Fund writes parameters for the Submit Transactions job.	$$AMSPARM$$/SubmitRevolvingFundReplenishment.txt

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

Major Output

BGAA transactions processed to final, pending, or rejected.

Job Return Codes

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

Return Code	Condition
Successful (1)	The job was able to find the parameter file and transactions were processed. This does not mean that all were accepted. Those not accepted are listed in the job log as well as in the exception report created by the next job step.
Warning (4)	Not Applicable for this job.
Non-Fatal Error (8)	Not Applicable for this job.
Failed (12)	Submit parameter file is not found.
Terminated (16)	This return code is issued when the job is terminated by the user.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues.

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	None or not all of the transactions successfully submitted to final.	Make corrections to the setup tables in the application and resubmit the transactions manually or through a separate SMU job.	
Failed (12)	Submit parameter file not found.	Correct where the submit step is looking for the file or the file name. The file could also be renamed or moved to where the submit job is looking.	Correct default parameter values on Job Setup to prevent this in the future.
Failed (12)	The job failed because of runtime exceptions for an unexpected situation.	The failure reason needs to be investigated before scheduling a new job.	

2.1.34 Treasurer Interface Process

Chain Name	Treasurer Interface Process
Recommended Frequency	Daily if not more frequently on demand
Single Instance Required	Yes
Can be restarted?	N/A for chain job.
Reports generated	N/A for chain job.

Overview

The Treasurer Interface Process reads from the specified input journal based on transaction codes, event types, and posting codes established in the Treasurer Interface Parameters (TREASPAR) page. Refer the "Advanced - Unique Features" section in the *CGI Advantage 4 - General Accounting User Guide* for details. Using these parameters, the process will create a journal voucher transaction for each transaction identified using the BSA and posting codes specified in the Treasurer Interface Parameters (TREASPAR) page and create entries to a special treasury fund.

The intent of these postings is not for financial reporting purposes but for bank reconciliation purposes. Any such reports are built in through a reporting tool and not part of this process.

The Treasury Interface Process has the following jobs in the chain.

- Treasurer Interface XML Creation - Creates the journal voucher XML file.
- Multi-threaded Transaction Loader - Breaks up the XML into smaller files and processes each with a spawned SMU job.

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

Important Run Instructions

1. The Treasurer Interface Process should be run after all nightly postings have occurred. Any postings that would be processed should not post while an instance of this chain is running or risk getting a Journal Log update that skipped records for processing.
2. The record written to the Journal Log (JLOG) will contain the following details. Such a record needs to be added to JRNL_LOG if during implementation it is determined that all historical records should not be processed and selection should begin at a recent point.
    - Run Date (PROC_RUN_DT) – Today's Date
    - Process ID (PROC_ID) – TREASINT
    - Journal/Ledger Name (JRNL_ID) – Value of the JRNL parameter in the Treasurer Interface XML Creation Job
    - Source Journal/Ledger ID (JRNL_LDGR_ID) – ID of the JRNL parameter as defined on Journal Ledger Control
    - Description (DSCR) – Treasurer Interface
    - Run Type (RUN_TYP) – INITIAL (1)

- Status (ST_FL) – FINAL (2)
- Begin Source Journal Record Number (BGN_JRNL_REC_NO) – Begin Record number from Cash/Input Journal
- End Source Journal Record Number (BGN_JRNL_REC_NO) – End Record number from Cash/Input Journal

Major Input

- Treasurer Interface Parameter (TREASPAR / TREAS_BAT_PARM)

Major Output

- Journal Vouchers

Chain Return Code

When an active job step in the chain ends with a Return Code of anything other than successful, any remaining job steps have a Run Status of *Inactive.* Please see documentation on individual job steps for possible Return Codes for those steps, as *Warning* and *Non-Fatal Error* are not always possible outcomes.

Problem Resolution

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

Treasurer Interface Process: Treasurer Interface XML Creation

Job Name	Treasurer Interface XML Creation
Recommended Frequency	N/A for job steps
Single Instance Required	N/A for job steps
Can be restarted?	Yes, in certain cases
Reports generated	No

Overview

After parameter validation, this job step selects records from the input journal and creates an XML output file of journal vouchers. The generated journal voucher transaction memo references each posting line identified in the input journal.

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

Process Steps	Messages

5. Parameter Validation	Run Started • Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log.
6. Journal Record Selection	• Records read ## • Records matched ## • Records written ## • Records suppressed ##
7. Create JV.XML	No messages

Input

- Treasurer Interface Parameter (TREASPAR / TREAS_BAT_PARM)
- Cash Journal (JCASH / JRNL_CASH) or Accounting Journal (JACTG / JRNL_ACTG)
- Journal Log (JLOG / JRNL_LOG)

Job Parameters

The parameters shown below are delivered values but may vary from what is implemented. Once this process is run, the selection parameter should not change for the Input Journal.

Parameter	Description	Default Value
AMSEXPORT	Export Location at Treasurer Interface XM Creation.	$$AMSROOT$$/ExportImport
List of BSA codes to exclude from processing (BSA_EXCLUDE_LIST)	An optional selection parameter of balance sheet account codes that should be excluded from selection. Multiple values are allowed, if comma-separated.	No default
Credit BSA (CREDIT_BSA_PARM	An optional output parameter for cash records selected that do not match TREASPAR to record cash events in a generic fashion. If one of the credit or debit parameters are used all must be.	No default
Credit Posting Code (CREDIT_PSCD_PARM	An optional output parameter for cash records selected that do not match TREASPAR to record cash events in a generic fashion. If one	No default

	of the credit or debit parameters are used all must be.	
Debit BSA (DEBIT_BSA_PARM)	An optional output parameter for cash records selected that do not match TREASPAR to record cash events in a generic fashion. If one of the credit or debit parameters are used all must be.	No default
Debit Posting Code (DEBIT_PSCD_PARM)	An optional output parameter for cash records selected that do not match TREASPAR to record cash events in a generic fashion. If one of the credit or debit parameters are used all must be.	No default
Treasurer Department (DEPT_CODE_PARM)	A required output parameter of a department that is used in journal voucher creation.	No default
Transaction Code (DOC_CODE_PARM)	A required output parameter of the transaction code in the Journal Voucher transaction type that is used in transaction creation.	JVA
Treasurer Fund (FUND_CODE_PARM)	The required output parameter of a unique non-operating fund code used for treasury reporting.	No default
Input Journal (JRNL)	The required input parameter of the journal used for record selection. The Cash Journal would be more efficient than the Accounting Journal unless all activity for selection is not also configured for the Cash Journal.	JRNL_CASH
Treasurer Parameter ID (TREAS_PARM_ID)	A required selection parameter used for selecting TREASPAR	* (Wildcard)

	records. When the * wildcard is entered, all records are selected. When one or more is listed here (comma-separated), then only those records are selected and there will not be a Journal Log (JLOG) tracking update.	

Output

- JVATreasurerInterface.xml
- Journal Log (JLOG / JRNL_LOG)

Job Return Codes

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

Return Code	Condition
Successful (1)	All parameter validation passed, journal records were read, and at least one transaction was created and successfully processed.
Warning (4)	No eligible records found. This could be because of the following reasons: • No new journal records were created since the last run. • No selected activity has been posted since the last run.
Non-Fatal Error (8)	This job step does not end with this return code.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid. • Unable to find the SOPT record for current fiscal year • 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 as records are created in the order found in the journal.

Selection Criteria

Using all Parameter IDs or only those specified:

- First run – The process creates a Journal Log record with the Process ID of XYZ, and starts at the first record and read until the last. As such, if this process is implemented long after going live, inserting a Journal Log record to bypass old activity is recommended. The first run should have the wildcard for the Treasurer Parameter ID.
- Subsequent runs – When the Parameter ID is the wildcard, the process starts journal record selection at the point the latest Journal Log record finished. After running, a new Journal Log is created with an updated end point. When the Parameter ID is used, there is no Journal Log processing and the input journal is read from the beginning.
- Journal records are matched against the TREASPAR records that have the Suppress Parameter set to *No* on Transaction Code, Event Type and Posting Code (if supplied).
- If no match is found to a TREASPAR record, the posting defaults to the Debit BSA, Debit Posting Code, Credit BSA, and Credit Posting Code values defined in the Job Parameters. If no match is found to a TREASPAR record and the Debit BSA, Debit Posting Code, Credit BSA, and Credit Posting Code values are not defined in the Job Parameters, the record will be skipped and the job will continue processing the next record. For example, if a new transaction code is configured but not entered on TREASPAR. If later it is determined that records should be picked up, then a record can be added to TREASPAR and the process run with the specified TREASPAR Parameter ID to process these specific records.
- Skipped records are recorded in the VLS Out log. If not using the 4 batch parameters as backup to post cash records that do not match a suppress or non-suppress record, then care should be taken when new posting code setup occurs. Such new setup can be selected if it was skipped by updating TREASPAR and running the job against that TREASPAR ID.
- The batch creates a journal voucher transaction, populated with accounting lines from each of the source transactions meeting the selection criteria.
- This job compares the value of ACTG_LN from R_GEN_SOPT with the property value of MAX_LINE_LIMIT for JV_DOC_ACTG and enforces the lesser value when creating the JV transactions to determine the maximum number of accounting lines. If the maximum number of accounting lines is exceeded, another JV transaction will be created.
- If Parameter ID is anything other than * (Wildcard) then a job log message is written stating no update was made to the journal (JLOG). Such a run is to find activity previously skipped for prior runs with the wildcard and there were suppressed records. If no suppressed records and the 2 posting code and 2 BSA parameters were used to record non-suppressed activity, running with a specific Parameter ID is not recommended as it will ‘double’ the update from the original transaction, unless that is acceptable because reports do not select activity recorded by those 4 batch parameters.

Problem Resolution

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)	Parameters were valid.	N/A	N/A
Warning (4)	This job step does not end with this return code.	N/A	N/A
Non-Fatal Error (8)	This job step does not end with this return code.	N/A	N/A
Failed (12)	The job failed due to a fatal condition such as invalid parameter, no ADNT match, or invalid accounting templates. **Sample Message:** "XYZ is invalid"	Submit another instance of the chain with correct and valid parameters.	N/A
Terminated (16)	The job was terminated manually by the user.	The reason for the termination needs to be addressed. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.
System Failure (20)	The job was terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.

Record Selection

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	At least one record was found and processed successfully into a journal voucher	N/A	N/A
Warning (4)	Job ended with a Warning because: • No new journal records were found. **Sample Message:** No journal records	If it is thought that records should have been selected, verify the input journal.	N/A

	found, no JV transactions created. • New journal records were found but did not match selection criteria. **Sample Message:** No error is issued but counts reflect condition.		
Non-Fatal Error (8)	This job step does not end with this return code.	N/A	N/A
Failed (12)	Job failed due to Fatal conditions from unexpected condition.	The reason for the failure needs to be addressed. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be addressed. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.
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. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.

XML File Creation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	XML file created and saved to the Export Location	N/A	N/A
Warning (4)	This job step does not end with this return code.	N/A	N/A
Non-Fatal Error (8)	This job step does not end with this return code.	N/A	N/A

Failed (12)	This job step does not end with this return code.	N/A	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be addressed. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.
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. Once addressed another instance of the chain should be submitted.	If another instance of the chain has already been scheduled and run successfully, then this job should not be restarted.

Treasurer Interface Process: Multi-threaded Transaction Loader

Job Name	Multi-threaded Transaction Loader
Recommended Frequency	See Chain
Single Instance Required	N/A for job steps
Can be restarted?	N/A for job steps
R99eports generated	No

Overview

This job takes the XML file created in the previous job step, breaks the file into smaller files, and then spawns multiple System Maintenance Utility jobs to load and submit the transactions. For example: If an input file has 31 transactions and the Block Size parameter is 5 and the Thread Count parameter is 3, then 3 split files are created and the first 5 transactions from the input file are written to the first split file, and the next 5 written to the next, and so forth. When complete, the first file contains 11 transactions. The second and third files contain 10 each. The interim processing was the first five records to file 1, the next five to file 2, the next five to file 3, the next five to file 1 again, and so forth. The final record was written to file 1 resulting in eleven.

The Multi-threaded Transaction Loader 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 as controlled by the Mode and Transaction Status parameters.

- Import – Split and import with a Phase of Draft and Status of Held.
- Import & Submit – Split, import, and submit. This is the intended mode for the Treasurer Interface Process.
- Import & Other Transaction Sub Action – This mode is not applicable to the Treasurer Interface Process.
- Import & Validate – Split, import, and validate resulting in a Status of Ready or Rejected.

The run sheet for “Multi-threaded Transaction Loader” in the *Utilities Run Sheets Guide* should be consulted for details including: Processing Steps, Return Codes, and Problem Resolutions. Each of these is common and not unique by use of this system feature with unclaimed property.

Input

- JVATreasurerInterface.xml

Job Parameters

Parameter	Description	Default Value
Block Size (BLOCK_SIZE)	Required number of transactions written to a split file before creating a new file.	100
Commit Block Size (COMMIT_BLOCK_SIZE)	Required block size to control the number of records committed in an instance.	1000
File Input Directory (FILE_INPUT_DIR)	Required directory where the XML is located.	$$AMSROOT$$/ExportImport
File List (FILE_LIST)	Required file name or names (comma separated) to be uploaded.	JVATreasurerInterface.xml
File Output Directory (FILE_OUTPUT_DIR)	Required directory to write the file segments created for processing.	$$AMSROOT$$/ExportImport
File Prefix (FILE_PREFIX)	Optional prefix used on the filenames of the output file segments.	
Bypass Approvals (I_SMU_BYPASS_APPROVAL)	Required indication if approvals should be bypassed in the transaction loaded. (TRUE, FALSE)	True
Log Status Interval (LOG_STATUS_INTERVAL)	Required logging frequency (in seconds) for controller thread reporting status of child threads to the system log.	300
Mode (MODE)	Required mode of operation. (1=Import, 2=Import and Submit, 3=Import and Other Action).	2
Other Transaction Action (OTHER_ACTION)	Conditionally required other action supplied to occur before submit. Parameter not applicable to this chain.	
Sleep Interval (SLEEP_INTERVAL)	Required sleep interval in seconds for internal controller thread for checking child processes.	5
SMU Catalog ID (SMU_CTLG_ID)	Required ID of the System Maintenance Utility job spawned as the child process.	3

Stagger Time (STAGGER_TIME)	Require lag time, in seconds, between the spawning of each child process.	30
Transaction Status (S_SMU_DOC_STA_CD)	Required status for loaded Transactions (2-Ready, 1-Held).	1
Exception Report File Name (S_SMU_EXCEP_REP_FILE_NM)	Required exception file name for recording submit failures.	
Exception Report Detail Level (S_SMU_EXCEP_REP_IND)	Required level of detail for exception reporting (1- Detailed, 2-Failed Transaction, 3- Processed Transaction, 4-Failed Transaction Lines, 5- Transaction Status).	2
Thread Count (THREAD_COUNT)	Required number of threads to use for processing.	6

Output

- Journal Voucher Transactions

Selection Criteria

Not applicable

Sort Sequence

Not applicable

2.1.35 Treasury Cash Balance

Chain Name	Treasury Cash Balance
Recommended Frequency	This job should be run daily after the Check Reconciliation job is completed. The job may be run more frequently to capture cash entries made during the day.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

The Treasury Cash Balance job is used to capture transactions from the Accounting Journal (JACTG) and Paid Checks (PDCHK) that update the cash accounts specified for selection on the Treasury Cash Balance Inquiry (TRCBAL) page to provide a centralized view of activity. Refer to the "Advanced - Unique Features" section in the *CGI Advantage 4 - General Accounting User Guide* for details related the TRCBAL page.

The following table shows the various steps of the process and the messages issued at each step.

Process Steps	Messages
1. Parameter Validation	Run Started • Parameters are valid or invalid depending on the validation. Validating batch parameters • If the parameter is invalid, the invalid value is displayed in the log.
2. Record Selection	• Processing records for JV, CR, IET, and ITA Transaction types. • Total number of records processed for JV, CR, IET, ITA Transaction types: # If no records found for JV, CR, IET, ITA that is stated. • Processing records for AD, EFT, and MD Transaction types. • Total number of records processed for AD, EFT, and MD Transaction types: # If no records found for AD, EFT, and MD that is stated. Total Records processed: ##
3. Record Processing	No messages

Process Details

The job should be executed daily after the Check Reconciliation job is run and successfully. When executing the Treasury Cash Balance job, following parameter validation, the process

cross-checks the Table Last Date (TBL_LAST_DT) for checks and the Cleared Date (CLR_DT) for EFTs in Paid Checks (PDCHK). It then checks the latest Record Number (REC_NO) in the Journal Log (JLOG) from the Accounting Journal (JACTG). These steps ensure the selection of only the records added to both input tables since the job's last execution. These selected records are then filtered further based upon the parameters in the job.

The record written to the Journal Log (JLOG) contains the following details. Such a record needs to be added to JRNL_LOG if during implementation it is determined that all historical records should not be processed and selection should begin at a recent point. This typically would require a restatement with a summary balance of all prior activity.

- Run Date (PROC_RUN_DT) – Today's Date
- Process ID (PROC_ID) – TBAL
- Journal/Ledger Name (JRNL_NM) – JRNL_ACTG
- Source Journal/Ledger ID (JRNL_ID) – 1
- Run Type (RUN_TYP) – INITIAL (1)
- Status (ST_FL) – FINAL (2)
- Begin Source Journal Record Number (BGN_JRNL_REC_NO) – Begin Record number from Accounting Journal
- End Source Journal Record Number (BGN_JRNL_REC_NO) – End Record number from Accounting Journal

Major Input

- Accounting Journal (JACTG / JRNL_ACTG)
- Paid Checks (PDCHK / AP_PD_CHK)

Major Output

- Treasury Cash Balance Inquiry (TRCBAL / TBAL)

Job Parameters

The parameters shown below are delivered values but may vary from what is implemented.

Parameter	Description	Default Value
Transaction Code	A required selection parameter for one or more transaction codes for selection. Ensures that once the process is run, that codes are never dropped unless no longer used and ones are added only before their use. (Required).	AD, EFT, CR, CACR, CRA, IET, ITA, JV, JVA, JVC, JVCAM, JVP, MD
Cash PSCD	A required selection criteria for at least one cash posting code. Multiple values are allowed when	A001

	comma separated.	
Cash BSA	Optional selection criteria for one or more cash balance sheet accounts. Multiple values are allowed when comma separated.	(blank)
Exclude Indication	Optional selection criteria to indicate a journal record that matches other criteria that should be omitted from selection. The value entered is compared to the Line Description. For example, if the parameter value is “XX”, then any transaction with an Accounting Line Description of "XX-bank adjustment" is skipped.	(blank)
Check Status	A required selection parameter to indicate AD and MD records on Paid Check have cleared. Multiple values are allowed when comma separated.	3 (Paid)
EFT Status	A required selection parameter to indicate EFT records on Paid Check have cleared. Multiple values are allowed when comma separated.	3,4 (Paid, Cancelled)
COMMIT_BLOCK_SIZE	A required performance parameter to control the number of records created at once. If left blank, 1000 is assumed.	1000

Job Return code

The following table shows the potential job return codes for the Treasury Cash Balance job.

Return Code	Condition
Successful (1)	Parameters were valid and at least one record was selected for processing.
Warning (4)	No records were selected for processing.
Non-Fatal Error (8)	This job does not use this return code.
Failed (12)	The job fails 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

N/A

Selection Criteria

Records are selected from Accounting Journal (JACTG) where:

- The transaction code (DOC_CD) matches one in the Transaction Code parameter.
- The posting code (PSTNG_CD_ID) matches the value entered in the Cash PSCD parameter.
- The balance sheet account (BSA_CD) matches the value entered in the Cash BSA parameter.
- The value in the line description does not begin with the value entered in the Exclude Indication parameter.
- The record has a run date (RUN_TMDT) after the last time the job was run and the Journal Log (JLOG) entry was created.

Records are selected from Paid Checks (PDCHK) using the same criteria above plus:

- If the transaction is a check (AD/MD), the Check Status (CHK_STA) matches the value entered in the Check Status parameter.
- If the transaction is an EFT, the Check Status (CHK)STA) matches the value entered in the EFT Status parameter.
- The record was created subsequent to the last time the job was run based on the Table Last Date (TBL_LAST_DT) for checks (AD/MD) and the Cleared Date (CLR_DT) for EFTs.

Problem Resolution

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

Parmeter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Parameters were valid.	N/A	N/A
Failed (12)	The job failed due to a fatal condition such as an invalid parameter. **Sample Message:** "XYZ is invalid"	Submit another instance of the job with correct and valid parameters.	If another instance of the job has already been run for a subsequent day, the date skipped cannot be recreated.
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	If another instance of the job has already been scheduled and ran successfully, then

Possible Return Codes	Condition	Recommendation	Other Instructions
		schedule a new job.	this job should not be restarted. Instead, 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. Instead, a new job should be scheduled.

Record Selection

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	At least 1 record was selected.	N/A	N/A
Warning (4)	No records were selected.	If parameters have not changed, then there is no new activity.	N/A
Failed (12)	The job failed due to a fatal condition such as an invalid parameter. **Sample Message:** "XYZ is invalid"	Submit another instance of the job with correct and valid parameters.	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.	If another instance of the job has already been scheduled and run successfully, then this job should not be restarted. Instead, 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. Instead, a new job should be scheduled.

Record Processing

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All records selected were updated	N/A	N/A
Failed (12)	Job failed due to unexpected fatal conditions.	If the job fails because of runtime exceptions, investigate the exception reported by the process, resolve the error, and run a new job.	If another instance of the job has already been run for a subsequent day, the date skipped cannot be recreated.
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. Instead, 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. Instead, a new job should be scheduled.

2.2 General Accounting Report Processes

The General Accounting report run sheets included in this section are:

- Automated Accrual/Accrual Clearing Mismatch Report
- Balance Sheet Report – Governmental Funds
- Budget vs. Actual Report for Expense
- Budget vs. Actual Report for Revenue
- Detail Transaction Listing
- Department Transaction Listing Report
- Encumbrance Activity
- Encumbrance Activity 2
- Open Activity Lapse Report
- Open Activity Roll Report
- Open Items Report
- Rebuild Balance Sheet Balance Information (Rebuild BBAL)
- Revenue and Expense by Fund Report
- Trial Balance Report
- Trial Balance with Prior Period Balances

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

2.2.1 Automated Accrual/Accrual Clearing Mismatch Report

Chain or Job Name	Automated Accrual/Accrual Clearing Mismatch Report
Recommended Frequency	On demand
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	Yes

Overview

Some scenarios exist where the amounts on the Automated Accrual (ACCA) and Automated Accrual Clearing (ACLA) transactions will not match. An example is when a current year Payment Request referencing a prior year encumbrance is modified downward after the Accounts Payable Period ends and the site had chosen procedurally not to run the Automated Accrual batch process. In this case, the Clearing would be for less than the Accrual. This report provides a way to investigate such discrepancies and manually update the transactions to ensure the Accrual and Clearing amounts match.

This report lists any transactions where the Accrued Amount (Line Amount) on the ACCA Accounting Line does not equal the Cleared Amount (Line Amount) on the matching ACLA Accounting Line. The match is made using the payment request information in the Special Reference section of the ACLA and the payment request in the regular Reference section of the ACCA transaction.

This report can be run as part of the normal cycle after the Automated Accrual Clearing process or on an ad hoc basis. The report should be reviewed and discrepancies carefully evaluated. Besides those modifications and cancellations to payment requests after the final running of the Automated Accrual chain, many mismatches will result from user actions:

- Modifying the ACCA but not the ACLA
- Modifying the ACCA and the ACLA differently
- Cancelling the ACCA but not the ACLA
- Cancelling the ALCA but not the ACCA

In most cases the discrepancy should be analyzed and corrected manually as needed.

Process Steps	Messages

1. Parameter Validation	• Run Started
	• Each parameter is listed with value
	• Any parameter errors are listed
	• If any validations fail the parameter and error message is displayed followed by - *Parameter Validation Failed*
	  • When AMSEXPORT is invalid then message displays “Error occurred while validating Export Path”
	  • When SELECT_BLOCK is invalid then message displays “Error occurred while validating Select Size”
	  • When PROG_CTR_SZ is invalid then message

	displays “Error occurred while validating Progression Counter” • When FROM_DT is invalid then message displays “Error occurred while validating From Date” • When TO_DATE is invalid then message displays “Error occurred while validating To Date” • When ACRL_DOC_CD is invalid then message displays “Error occurred while validating Transaction Code”
2. Selection of Records	• Reports output folder mapped • PDF and HTML output locations mapped • No accrual records found matching selected criteria. *If the selection returns 0 records* • Number of records (count) selected
3. Report Generation	• Rendering report started • Rendering report completed • Run Ended

Major Input

- CH_DOC_ACTG

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
ACRL_DOC_CD	(Required) Accrual Transaction Code	ACCA
AMSEXPORT	(Required) Export Location for Upload Job	$$AMSEXPORT$$
DOC_CD	(Required) Accrual Clearing Transaction Code	ACLA
FROM_DT	(Required) The Clearing Range From Date is used to limit the data for comparison. Please enter as CCYY-MM-DD.	(blank)
PROG_CTR_SZ	(Required) The Progression Counter Size defines the number of records compared that will trigger a progression message in the job log.	5000
SELECT_BLOCK	(Required) The Select Block Size defines the number of accounting lines selected in a query. It exists for performance tuning.	3000

TO_DATE	(Optional) The Clearing Range To Date is used to limit the data for comparison. Leaving blank will compare all records up to the most current.	(blank)

Major Output

- Mismatch exception report

Chain / Job Return code

Return Code	Condition
Successful (1)	All parameters were valid, clearing records were selected, and all clearings matched accruals.
Warning (4)	No records selected for batch parameters
Non-Fatal Error (8)	Clearing lines were found that did not equal accrual lines
Failed (12)	The job will fail under the following conditions: - Parameters are invalid - Any other exceptions
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 report is in ascending order by Accrual Clearing (ACLA)

- Transaction Code
- Transaction Department Code
- Transaction ID

Selection Criteria

Accrual Clearing (ACLA) records will be selected from CH_DOC_ACTG where:

- Record dates (DOC_REC_DT) are between the FROM_DT and TO_DATE specified in the job parameters. If TO_DATE is not specified in the parameter, selection will be based on records dated between the FROM_DT specified in the job parameters and the current system date;
- AND posting pair type (PSTNG_PR_TYP) is A (PY Clearing)

- AND reference transaction code (SPRED_DOC_CD) is equal to the transaction code for transaction type PR or ABS.

Accrual (ACCA) records will be selected from CH_DOC_ACTG where:

- Referenced transaction information (RFED_DOC_CD, RFED_DOC_DEPT_CD, RFED_DOC_ID, RFED_VEND_LN_NO, RFED_COMM_LN_NO, and RFED_ACTG_LN_NO) equals the special reference values of the selected ACLA records.

Further, each record selected by the above selections:

- System will sum the Line Amount of selected clearing accounting lines for a payment request reference
- System will sum the Line Amount of selected accrual accounting lines for the same payment request reference
    - If PY_ACRL_CLR_PC is 0% or the ACCA/ACLA is cancelled the Line Amount is considered to be $0.
- If the Line Amount of the selected clearing accounting lines and the Line Amount of the referenced accounting lines are not equal:
    - Information will be written in the report

Problem Resolution (All Process Steps)

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully and mismatch records selected for report message will be issued saying “Number of records selected for reports."	N/A	N/A
Warning (4)	The job ended with a Warning because there are no records selected for the mismatch report. A message will be issued saying “No Accrual records found matching selection criteria.”	Check parameter for valid records.	Alternatively, the job can be rescheduled with a different set of parameters. This return code can be acceptable outcome if the range of From and To Dates didn’t contain any ACLA transactions.
Non-Fatal Error (8)	This return code will be issued when records are selected that do not match.	Review report and take any manual actions necessary.	N/A
Failed (12)	Job failed due to Fatal conditions or invalid parameter.	In this step, the job can fail under the following two conditions:	If another instance of the job has already been scheduled and run successfully, then this job should not be

		1) Encounters any runtime exceptions 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.	restarted – only a new job should be scheduled.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. 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.

Report sample with mismatch records:

2.2.2 Balance Sheet Report – Governmental Funds

Description

The governmental funds statements include short–term information – the most readily available assets and presently due liabilities - and just the resources that flow into and out of a state government during the year or shortly thereafter.

The governmental funds financial statements show the Major Funds individually, and total the remaining governmental funds in a single column.

The report includes information used to:

Determine a state government's short-term financing needs and assess its capacity to meet them – for instance, by comparing cash and short-term investments with liabilities that are expected to be liquidated in the next year.

Assess a state government's short-run ability to balance inflows of resources with outflows to pay for services by contrasting revenues and expenditures.

When to Run

On request

Major Input

- Accounting Period Ledger (LDGR_APD_ACTG)
- Fiscal Year (R_FY)
- Balance Sheet Account (R_BSA)
- Fund (R_FUND)
- Posting Code (R_PSCD)

Output

Report (PDF and HTML)

Parameters

Job or Job # in a Chain	Parameter	Description	Default Values
BALANCE SHEET FOR GOVERNEMENTAL FUNDS	Client Name (CLIENT_NM)	Optional field for the name of client to be appear on report	No default
	Balance Sheet Account Selection (BSA_PSCD_CLOS_CL_CD)	Required posting code close classification for balance sheet selection. Multiple can be listed if separated by	No default

		commas.	
	Major Fund (FUND_CD)	Required to have one, but can have up to 3 separated by commas.	No default
	Fiscal Year (FY)	Optional field for a year to be used as selection criteria. Must be entered as CCYY. If one is not entered the today's date will be used to determine a fiscal year.	No default
	Nominal Account Selection (NOM_PSCD_CLOS_CL_CD)	Required posting code close classification for nominal activity to compute unreserved fund balance. Multiple can be listed if separated by commas.	No default
	Accounting Period (PER)	Optional field for an account period to be used as selection criteria. It will serve as a 'as of' period, which means activity in it and prior periods with the fiscal year are selected. If not entered, today's date is used to determine an accounting period.	No default

Sort Order and Selection Criteria

1. Select Records where:

   Fund = Fund Parameter (if blank include all funds – also if more than one is provided, the "Other Funds" and "Total" columns will be provided)

   Fiscal Year = Year Parameter

   Accounting period <= Accounting Period Parameter

   Posting Codes have values for the Balance Sheet parameter's Closing Classification Codes

   Posting Codes with Nominal Account parameter's closing classification codes

2. Sort by
   Fund_Cd
   Account Type of Balance Sheet Accounts
   BSA_Cd

3. Logic

Add Fund Column if Fund is defined as a Major Fund (NOTE: up to three can be handled). For records where the Posting Code value equals the Closing Classification Code parameter, they should be in an Unreserved bucket under Fund Balance.

Read Sorted Records

If Fund_Cd/Account Type/BSA_Cd does **not** change ← break and label depends on Roll-up parameter

Then

Add Pstng_Am to Balance Sheet Account Total

Else

Print Detail Line for that fund

If there is a change in Account Type (that is Asset vs Liability/Fund Balance) – Print Asset Totals

If there is a change in Fund, create new column for new fund and keep a running total for all funds

If more than one fund is reported on, include “Other Funds” and “Total” columns.

Format

Titles include fund names.

Balance Sheet Account is shown at each detail level.

Net totals are given for break of Balance sheet account type Indicator (Assets, Liabilities and Fund Balances)

Grand total for Liabilities and Fund Balances is shown.

Actual HTML Report Format

The output of Governmental Funds Balance Sheet Report in HTML format will look like the following

CGI Advantage FINANCIAL

RUN DATE: 04-13-2001 GOVERNMENTAL FUNDS BALANCE SHEET PAGE 1

As of 9/2001

	General Fund	Highway Fund	Other Funds	Total
ASSETS				
Cash – Bank South	200.00	-	200.00	400.00
Cash – AmSouth	-	400.00	100.00	500.00
Total assets	200.00	400.00	300.00	900.00
LIABILITIES				
Liabilities:				
Disbursements Payable	300.00	-	400.00	700.00
Account Payable	-	500.00	-	500.00
Total liabilities	300.00	500.00	400.00	1200.00
Fund Balances:				
Reserve for Encumbrance	-	300.00	100.00	400.00
Unreserved Fund Balances	(100.00)	(400.00)	(200.00)	(700.00)
Total fund balances	(100.00)	(100.00)	(100.00)	(300.00)
Total liabilities and Fund Balances	200.00	400.00	300.00	900.00

2.2.3 Monitoring and Tracking Budgets – Budget vs. Actual Report for Expense

The Budget vs. Actual Report for Expenses is a sample report, which is built to look at budget structure 29. The report compares the current budget amount against encumbered, accrued expenses, and cash expenses. The spending amounts are calculated for the selected accounting period and then again for an inception-to-date total up through that accounting period. Finally, there is a difference computed between the current budget amount and the total spending amounts. The report shows these details at the lowest level (Fund + Department + Appropriation + Unit + Object). Summary pages exist providing totals at these higher levels:

- Fund, Department and Appropriation
- Fund and Department
- Fund
- Budget Fiscal Year

Description

Budget amounts are retrieved from the Budget Data Input Source parameter, normally set to the Budget Journal (JRNL_BUD). The Current Budget Amount (Exp Budget on the report) is calculated by adding amounts for Adopted, Amended, Awarded, Carried Forward, Transferred In, and Transferred Out for all budget journal records within the selected Budget Fiscal Year, Accounting Fiscal Year, up through the selected Accounting Period.

Accounting amounts are retrieved from the Accounting Data Input Source parameter, normally set to the Systems Assurance Ledger (LDGR_SA_BUD). The Encumbered Amount is calculated by adding amounts on ledger records with posting codes set to update the budget bucket by that name (Bucket ID 13). The first instance of this amount in the Current section contains a total of those records within the selected Budget Fiscal Year, Accounting Fiscal Year, and Accounting Period. The second instance in the YTD or ITD section contains a total of those records within the selected Budget Fiscal Year, Accounting Fiscal Year, and through the selected Accounting Period. The Accrued Expenses Amount has the same selection criteria except ledger records for bucket ID 14 are selected. The Cash Expenses Amount has the same selection criteria except ledger records for bucket ID 15 are selected.

The reason for the second section having the YTD or ITD title is because those budget lines that are multi year (as indicated by a BFY of 9999), display an inception-to-date total, as the only BFY recorded for them is 9999. All other budget lines defined by a single BFY display a year-to-date total. The ‘Difference’ is calculated by subtracting the second total accounting amount from the total budgeting amount.

When summarizing details into higher-level totals, a value of ‘****’ is a way of signifying that what is being displayed is a summary line and that the key was summarized off.

Parameters in the report allow the selection of budget line(s) to be specific towards particular budget fiscal years. If the required parameters are not entered or any of the parameters entered are invalid then the return status for the job is set to “Failed”.

When the report is run:

- The Accounting Bucket Id’s (13, 14, & 15) and the Budget Bucket Id’s (1, 3, 4, 9, 10, & 21) are retrieved from the General Bucket Table (R_GN_BKT).
- The report takes the budget buckets specified and looks through the Posting Code table (R_PSCD) to find the posting codes that update each bucket. The report uses this information to aggregate activity for each bucket.

- The Accounting Source (that is, Journal or Ledger) will be used to get the Total Dollar Amount posted against each Budget Line for the Posting code associated with the Accounting Buckets up to the Accounting Period entered in the parameter. For example, the bucket 'Accrued Expenditures' is chosen then the records found on the Accounting Activity Source (LDGR_SA_BUD in our example) for those posting codes associated with the 'Accrued Expenditures' bucket would be summarized by the chart-of-accounts on each budget line.
- The report considers only the Accounting Source for which the associated Budget line exists in the Expense budget structure (BUD_STRU_29_LVL_2).
- The Budgeting Activity Source will be used to get the Total dollar amount posted against each Budget Line for the Posting code associated with the Budget Buckets and up to (inclusive of) the Accounting Period entered as a parameter for Budget Structure 29 level 2.
- When displaying amounts on the report if the bucket type is "Accounting", then keep the sign as found on the source. If the bucket type is "Budget" then use the increase/decrease field to determine the sign. All amounts stored in the budget journal are positive, and if combined with an increase, then the amount stays positive. When combined with a decrease, then the amount is made negative.

When to Run

The report can run on demand and when an accounting period is being closed, to produce an easy-to-read listing of budget lines.

The Ledger Posting Engine should be run to ledgerize all accounting journal records so that the report will have the most up-to-date information.

System Assurance1 should be run prior to this Report as the default Accounting Source (LGDR_SA_BUD) can be verified by the System Assurance 1 process.

Major Input

- Accounting Activity Source - the journal or ledger data object name that will provide accounting information.
- Budgeting Activity Source - the journal or ledger data object name that will provide budgeting information.
- Revenue Budget Structure (BUD_STRU_29_LVL_2) – to cross verify if the selected Accounting source has an associated budget line.
- General Bucket Table (R_GN_BKT) – to get the Accounting and Budget Bucket Id(s).
- Posting Code Table (R_PSCD) – to get the Posting Codes associated with the Accounting and Budget Bucket Id(s).

Output

- The Budget vs. Actual Expense Report

Parameters

Parameter	Description	Default Value	Required	For System Use (non-overridable by user)
Budget Fiscal Year	Specify the budget fiscal year for which the activity is to be found as CCYY	No Default	Yes	Yes
CLIENT_NM	Client Name appearing on the report	No Default	No	Yes
Accounting Activity Source	Specify the journal or ledger data object name that will provide accounting information. Use all capital letters separated by '_' where necessary.	LDGR_SA_BUD	Yes	Yes
Budgeting Activity Source	Specify the journal or ledger data object name that will provide budgeting information. Use all capital letters separated by '_' where necessary.	JRNL_BUD	Yes	Yes
Current Accounting Period	Specify the accounting period for which activity is to be labeled current.	No Default	Yes	Yes
Current Fiscal Year	Specify the fiscal year for which activity is to be found as CCYY.	No Default	Yes	Yes

Sort Sequence

The Budgets are tracked at various levels; hence, the records are sorted by Fund, Department, Appropriation, Unit, Object, Fiscal Year, Accounting Period and Posting Code.

Selection Criteria

Select criteria for selecting records from Accounting Activity Source (Ledger / Journal) is:

- Budget Fiscal Year = Budget Fiscal Year entered as Parameter
- Fiscal Year = Fiscal Year entered as Parameter
- Accounting Period = Accounting Period entered as Parameter
- Posting Code In (associated Posting Codes for the Accounting Buckets)

Select criteria for selecting records from Budgeting Activity Source (Ledger / Journal) is:

- Expense Budget Structure Id = 29
- Expense Budget Level Id = 2
- Budget Fiscal Year = Budget Fiscal Year entered as Parameter
- Fiscal Year = Fiscal Year entered as Parameter
- Accounting Period = Accounting Period entered as Parameter
- Posting Code In (associated Posting Codes for the Budget Buckets)

Problem Resolution

If the process was discontinued for any reason then the job has to be rescheduled. If the job failed because of invalid parameters, it can be restarted and new parameters used.

2.2.4 Monitoring and Tracking Budgets – Budget vs. Actual Report for Revenue

The Budget vs. Actual Report for Revenues is a sample report, which is built to look at budget structure 30. The report compares the current revenue budget amount against billed and collected earned revenues. Total billed and collected amounts are calculated for the selected accounting period and then again for an inception-to-date total up through that accounting period. Finally, there is a difference computed between the current revenue amount and the total billed and earned amount. The report shows these details at the lowest level (Fund + Department + Unit + Revenue Source). Summary pages exist providing totals at these higher levels:

- Fund and Department
- Fund
- Budget Fiscal Year

Description

Budget amounts are retrieved from the Budget Data Input Source parameter, normally set to the Budget Journal (JRNL_BUD). The Current Budget Amount (Rev Budget on the report) is calculated by adding amounts for Adopted, Amended, Carried Forward, and Transferred for all budget journal records within the selected Budget Fiscal Year, Accounting Fiscal Year, up through the selected Accounting Period.

Accounting amounts are retrieved from the Accounting Data Input Source parameter, normally set to the Systems Assurance Ledger (LDGR_SA_BUD). The Billed Earned Revenue Amount (Bill Earned Rev on the report) is calculated by adding amounts on ledger records with posting codes set to update the budget bucket by that name (Bucket ID 22). The first instance of this amount in the Current section contains a total of those records within the selected Budget Fiscal Year, Accounting Fiscal Year, and Accounting Period. The second instance in the YTD or ITD section contains a total of those records within the selected Budget Fiscal Year, Accounting Fiscal Year, and through the selected Accounting Period. The Collected Earned Revenue Amount (Coll Earned Rev on the report) has the same selection criteria except ledger records for bucket ID 24 are selected.

The reason for the second section having the YTD or ITD title is because those budget lines that are multi year (as indicated by a BFY of 9999) display an inception-to-date total, as the only BFY recorded for them is 9999. All other budget lines defined by a single BFY display a year-to-date total. The 'Difference' is calculated by subtracting the second total accounting amount from the total budgeting amount.

When summarizing details into higher-level totals, a value of '****' is a way of signifying that what is being displayed is a summary line and that the key was summarized off.

Parameters in the report allow the selection of budget line(s) to be specific towards particular budget fiscal years. If the required parameters are not entered or any of the parameters entered are invalid then the return status for the job is set to "Failed".

When the job is run:

- The Accounting Bucket Id's (22 & 24) and the Budget Bucket Id's (29, 31,32, & 37) are retrieved from the General Bucket Table (R_GN_BKT).
- The report takes the budget buckets specified and looks through the Posting Code table (R_PSCD) to find the posting codes that update each bucket. The report uses this information to aggregate activity for each bucket.

- The Accounting Source (that is, Journal or Ledger) is used to get total dollars posted against each Budget Line for the Posting code associated with the Accounting Buckets up to the Accounting Period entered as a parameter. For example, the bucket 'Billed Earned Revenue' is chosen then the records found on the Accounting Activity Source (LDGR_SA_BUD in our example) for those posting codes associated with the 'Billed Earned Revenue' bucket would be summarized by the chart-of-accounts on each budget line.
- The report considers only the Accounting Source for which the associated Budget line exists in Revenue budget structure (BUD_STRU_30_LVL_1).
- The Budgeting Activity Source is used to get the Total dollars posted against each Budget Line for the Posting code associated with the Budget Buckets up to the Accounting Period entered as a parameter for Budget Structure 30 level 1.
- When displaying amounts found from the sources on the report. If the bucket type is "Accounting", then keep the sign as found on the source. If the bucket type is "Budget" then use the increase/decrease field to determine the sign. All amounts stored in the budget journal are positive, and if combined with an increase, then the amount stays positive. When combined with a decrease, then the amount is made negative.

When to Run

The report can run on demand and when an accounting period is being closed, to produce an easy-to-read listing of budget lines.

The Ledger Posting Engine should be run to ledgerize all accounting journal records so that the report will have the most up-to-date information.

System Assurance1 should be run prior to this report as the default Accounting Source (LGDR_SA_BUD) can be verified by the System Assurance 1 process.

Major Input

- Accounting Activity Source - the journal or ledger data object name that will provide accounting information
- Budgeting Activity Source - the journal or ledger data object name that will provide budgeting information.
- Revenue Budget Structure (BUD_STRU_30_LVL_1) – to cross verify if the selected Accounting source has an associated budget line.
- General Bucket Table (R_GN_BKT) – to get the Accounting and Budget Bucket Id(s).
- Posting Code Table (R_PSCD) – to get the Posting Codes associated with the Accounting and Budget Bucket Id(s).

Output

- The Budget vs. Actual Revenue Report

Parameters

Parameter	Description	Default Value	Required	For System Use (non-overridea ble by user)
Budget Fiscal Year	Specify the budget fiscal year for which the activity is to be found as CCYY.	No Default	Yes	Yes
CLIENT_NM	Client Name appearing on the report.	No Default	No	Yes
Accounting Activity Source	Specify the journal or ledger data object name that will provide accounting information. Use all capital letters separated by '_' where necessary.	LDGR_SA_ BUD	Yes	Yes
Budgeting Activity Source	Specify the journal or ledger data object name that will provide budgeting information. Use all capital letters separated by '_' where necessary.	JRNL_BUD	Yes	Yes
Current Accounting Period	Specify the accounting period for which activity is to be labeled current.	No Default	Yes	Yes
Current Fiscal Year	Specify the fiscal year for which activity is to be found as CCYY.	No Default	Yes	Yes

Sort Sequence

The Budgets are tracked at various levels; hence, the budget records are sorted by Fund, Department, Unit, Revenue Source, Fiscal Year, Accounting Period, and Posting Code.

Selection Criteria

Select criteria for selecting records from Accounting Activity Source (Ledger / Journal) is:

- Budget Fiscal Year = Budget Fiscal Year entered as Parameter

- Fiscal Year = Fiscal Year entered as Parameter
- Accounting Period = Accounting Period entered as Parameter
- Posting Code In (associated Posting Codes for the Accounting Buckets)

Select criteria for selecting records from Budgeting Activity Source (Ledger / Journal) is:

- Revenue Budget Structure Id = 30
- Revenue Budget Level Id = 1
- Budget Fiscal Year = Budget Fiscal Year entered as Parameter
- Fiscal Year = Fiscal Year entered as Parameter
- Accounting Period = Accounting Period entered as Parameter
- Posting Code In (associated Posting Codes for the Budget Buckets)

Problem Resolution

If the process was discontinued for any reasons then the job has to be rescheduled. If the job failed because of invalid parameters, it can be restarted and new parameters used.

2.2.5 Detail Transaction Listing

Description

The report produces a detailed listing of specified transactions as they appear in the Journals. By specifying appropriate dates as parameters for the program, you can report on a single day or range of days.

When to Run

On request; usually nightly to verify the day's transactions, but often as part of the pre-monthly closing report series.

Major Input

- Journal Table depending on the Journal ID entered
- Vendor Customer Table

Output

- Report (PDF and HTML format)

Parameters

Job or Job # in a Chain	Parameter	Description	Default Values
TRANSACTION LISTING	Client Name (CLIENT_NM)	Optional field for the name of client to be appear on report	No default
	From Date (FROM_DT)	Required field for a date used as selection criteria. Must be entered as mm/dd/ccyy.	No default
	Journal ID (JRNL_LDGR_ID)	Required unique ID number for the journal to serve as input to the report. See the Journal/Ledger Control table for valid values. Note: Only choose a journal that contains accounting data.	No default
	To Date (TO_DT)	Required field for a date used as selection criteria. Must be entered as mm/dd/ccyy.	No default

	Transaction Type (TRAN_TYP)	Optional transaction type to produce a selective report. More than one can be entered if separated by commas. Leaving the field blank will result in all transaction types being selected.	No default

Sort Sequence and Selection Criteria

1. Select journal records where

   Run Date is between the entered from and to dates

   Transaction Type = Transaction Type parameter (if specified)

2. Sort Query Object

   Doc_Typ
   Doc_Dept_Cd
   Doc_Cd
   Doc_Id
   Doc_Vers_no
   Fund_Cd
   Dept_Cd
   DrCr_Ind

3. Logic for Detailed Report

   Read Sorted Records, if Transaction Id changes print the transaction header information:

   Transaction Typ
   Transaction Cd
   Transaction Dept Cd
   Transaction ID
   Run Tmdt
   Transaction Vers No

Format

Columns are:

- BFY
- Fund Cd
- Dept Cd
- Obj/Rsrc Cd
- Sobj/Srsrc Cd
- Pstng Cd
- Bsa Cd
- Vendor/Customer Name
- Pstng Am
- DrCr_Ind

In the Transaction header, the following is displayed:

- Caption Transaction Typ followed by the Transaction Type
- Caption Transaction Cd followed by the Transaction Code
- Caption Transaction Dept Cd followed by the Transaction Department Code
- Caption Transaction Id followed by the Transaction Id
- Caption Transaction Dt followed by the Run datetime
- Caption Transaction Vers No followed by the Transaction Version Number

Actual HTML Report Format

The output of Detail Transaction Listing Report in HTML format will look like the following

```
                              CGI Advantage FINANCIAL OPEN ITEM REPORT
RUN DATE: 3/3/2001                                                                    PAGE  1
-------------------------------------------------------------------------------------------
-----------------
                              Detail Transaction Listing

BFY    FUND   DEPT   OBJ/SUB      PSTNG  BSA    VENDOR/CUSTOMER                  AMOUNT
-------------------------------------------------------------------------------------------

                              From 1/1/2001 To 4/3/2001

DOC_TYP:PO     DOC_CD:DO    DOC_DEPT_CD:150     DOC_ID:PT    RUN_TMDT:03/31/2001        DOC_VERS_NO:1

2001   101    150    7770         P005   3120   Business Technologies Inc        1,000.00 C
2001   101    150    7770         P006   3120   Business Technologies Inc        1,000.00

DOC_TYP:PO     DOC_CD:DO    DOC_DEPT_CD:150     DOC_ID:PO1   RUN_TMDT:03/31/2001        DOC_VERS_NO:1

2001   101    150    7770         P005   3121   Business Technologies Inc        2,000.00 C
2001   101    150    7770         P006   3121   Business Technologies Inc        2,000.00

DOC_TYP:PO     DOC_CD:DO    DOC_DEPT_CD:150     DOC_ID:PO2   RUN_TMDT:03/31/2001        DOC_VERS_NO:1

2001   101    150    7770         P005   3120   Business Technologies Inc        3,000.00 C
2001   101    150    7770         P006   3120   Business Technologies Inc        3,000.00

DOC_TYP:PO     DOC_CD:DO    DOC_DEPT_CD:150     DOC_ID:P03   RUN_TMDT:03/31/2001        DOC_VERS_NO:1

2001   101    150    7770         P005   3120   Business Technologies Inc        4,000.00 C
2001   101    150    7770         P006   3120   Business Technologies Inc        4,000.00

DOC_TYP:PO     DOC_CD:DO    DOC_DEPT_CD:150     DOC_ID:P04   RUN_TMDT:03/31/2001        DOC_VERS_NO:1
```

```
2001   101    150    7770          P005   3120   Business Technologies Inc        5,000.00 C
2001   101    150    7770          P006   3120   Business Technologies Inc        5,000.00
```

2.2.6 Department Transaction Listing Report

Description

This report produces a detailed listing of specified transactions as they appear in the Journals. By specifying appropriate departments as parameters for the program, you can report on a single day or range of transactions.

Major Input

- Journal Table depending on the Journal ID entered
- Department depending on Department ID

Output

- Report (PDF and HTML format)

Parameters

Job or Job # in a Chain	Parameter	Description	Default Values
TRANSACTION LISTING	Client Name (CLIENT_NM)	Optional field for the name of client to be appear on report	No default
	Department Code (DEPT_CD)	Required field for a code used as selection criteria.	No default
	Journal ID (JRNL_LDGR_ID)	Required unique ID number for the journal to serve as input to the report. See the Journal/Ledger Control table for valid values. Note: Only choose a journal that contains accounting data.	No default
	Report Date (REPORT_DT)	Required field for a date used as selection criteria. Must be entered as mm/dd/ccyy.	No default
	Report ID (REPORT_ID)	Optional report ID to produce a selective report. More than one can be entered if separated by commas. Leaving the field	No default

		blank will result in all transaction types being selected.	

Format

Columns are

- BFY
- Fund Cd
- Dept Cd
- Unit
- Sunit
- Actv
- Function Obj
- Bs Acct
- PPC Phase
- Vendor/Customer Name
- Amount

Actual HTML Report Format

The output of Detail Transaction Listing Report in HTML format will look like the following

CGI Advantage FINANCIAL OPEN ITEM REPORT

RUN DATE: 3/3/2001 **PAGE 1**

Detail Transaction Listing

From 1/1/2001 To 4/3/2001

BFY	FUND	DEPT	OBJ/SUB	PSTNG	BSA	VENDOR/CUSTOMER	AMOUNT
DOC_TYP:PO	DOC_CD:DO	DOC_DEPT_CD:150		DOC_ID:PT	RUN_TMDT:03/31/2001		DOC_VERS_NO:1
2001	101	150	7770	P005	3120	Business Technologies Inc	1,000.00 C
2001	101	150	7770	P006	3120	Business Technologies Inc	1,000.00
DOC_TYP:PO	DOC_CD:DO	DOC_DEPT_CD:150		DOC_ID:PO1	RUN_TMDT:03/31/2001		DOC_VERS_NO:1
2001	101	150	7770	P005	3121	Business Technologies Inc	2,000.00 C
2001	101	150	7770	P006	3121	Business Technologies Inc	2,000.00
DOC_TYP:PO	DOC_CD:DO	DOC_DEPT_CD:150		DOC_ID:PO2	RUN_TMDT:03/31/2001		DOC_VERS_NO:1
2001	101	150	7770	P005	3120	Business Technologies Inc	3,000.00 C
2001	101	150	7770	P006	3120	Business Technologies Inc	3,000.00
DOC_TYP:PO	DOC_CD:DO	DOC_DEPT_CD:150		DOC_ID:PO3	RUN_TMDT:03/31/2001		DOC_VERS_NO:1
2001	101	150	7770	P005	3120	Business Technologies Inc	4,000.00 C
2001	101	150	7770	P006	3120	Business Technologies Inc	4,000.00
DOC_TYP:PO	DOC_CD:DO	DOC_DEPT_CD:150		DOC_ID:PO4	RUN_TMDT:03/31/2001		DOC_VERS_NO:1
2001	101	150	7770	P005	3120	Business Technologies Inc	5,000.00 C
2001	101	150	7770	P006	3120	Business Technologies Inc	5,000.00

2.2.7 Encumbrance Activity

Job Name	Encumbrance Activity
Recommended Frequency	On demand
Single Instance Required	No
Can be restarted?	No
Reports generated	Yes

Description

For each Department, Unit, Appropriation and Object Class, the Encumbrance Activity Report displays the encumbrance transaction accounting lines with their vendor and object along with columns for the line amount, closed amount, and outstanding amount (called encumbered, paid and outstanding). The report takes a Fiscal Year (FY) and Accounting Period (APD) as parameters, and shows encumbrance lines for given FY with APD's less than or equal to the APD parameter. The FY and APD parameters are optional; if left blank, the current application date is used to determine the current FY and APD.

The Encumbrance Activity Report can be run in the Open Encumbrances and All Encumbrances modes. In the Open Encumbrances mode, only encumbrances with non-zero outstanding amounts are shown. In the All Encumbrances mode, all encumbrances are shown, even those with a zero outstanding amount. In the All Encumbrances mode, the amounts listed for each encumbrance line are the Encumbered Amount, Expended Amount (known online as Referenced Amount), Closed Amount and Outstanding Amount. The Expended Amount and Closed Amount may or may not be equal; they are different when an encumbrance is closed by a request for payment that either closed the encumbrance line short or over referenced because tolerances allowed such. Since the discrepancy is only possible for closed encumbrances, the report run in the Open Encumbrances mode does not have the Expended Amount column; only the Encumbered Amount, Closed Amount and Outstanding Amount are listed. Expended in this case would always equal the Closed amount.

The Encumbrance Activity Report uses the Accounting Journal for input. The report reads Accounting Journal records for given/defaulted FY and all accounting periods less than or equal to the given/defaulted APD parameter.

- Journal lines with a Line Function Code of 1 (Standard) with a posting code that has a closing classification equal to the Encumbrance Closing Classification code parameter are used to display the transactions and calculate their Encumbered Amounts.
- Transactions that reference the encumbrance line posted to the journal with one Line Function Codes of 3 (Liquidation) with a posting code that has a closing classification equal to the Encumbrance Closing Classification parameter. These lines are used to calculate the Closed Amount.
- The Outstanding Amount is calculated as the difference between the Encumbered Amount ant the Closed Amount.
- There are often other journal records for transactions referencing the encumbrance that posted to the journal with a Line Function Code of 1. Posting codes in these types of lines that have a closing classification equal to the Accrued Expenditure Closing Classification code parameter are used to calculate the Expended Amount, if the report is run in the All Encumbrances mode.

No specific transaction types or codes are selected for encumbrance activity or the closing of that activity, so the report can show transactions from the ABS, RQ, and PO transaction types

because they have delivered accounting models that allow encumbrance activity. Other transaction types customized to have encumbrance activity will also appear on the report.

Major Input

- Accounting Journal (JACTG / JRNL_ACTG)
- Posting Code (PSCD / R_PSCD)

Output

- Encumbrance Activity Report

Parameters

Parameter	Description	Default Value
Client Name (CLIENT_NM)	An optional name to appear in the report header.	No default
Report ID (REPORT_ID)	An optional ID to appear in the report header for identification and routing.	No default
Fiscal Year (FY)	An optional selection parameter.	Current fiscal year if left blank
Accounting Period (APD)	An optional selection parameter.	Current accounting period if left blank
Encumbrances (ENCUM_CLSD_CLS)	A required selection parameter that identifies all encumbrance posting codes.	12
Accrued Expenditures (EXP_CLSD_CLS)	A required selection parameter that identifies all expenditure posting codes.	11
Report Type (REPORT_TYPE)	A required output parameter for the type of report to produce: 1 (Open Encumbrances Only) or 2 (All Encumbrances).	No default

Sort Sequence

Encumbrance records are sorted by Department, Unit, Appropriation, Object Class, and Transaction Code/Dept/ID/Vendor Line/Commodity Line/Accounting Line.

Selection Criteria

To select encumbrance transactions and calculate the Encumbered Amount, select Standard posting lines from the Accounting Journal for the selected Fiscal Year, with accounting periods less or equal to the selected Accounting Period. Limit selection to lines with posting codes

mapping to the closing classification code(s) provided in the Encumbrance Closing Classification parameter.

To calculate the Closed Amount for an individual transaction, select the sum of the Accounting Journal amounts for records with Ref Transaction Code/Dept/ID/Vendor Line/Commodity Line/Accounting Line and Posting Code matching the Transaction Code/Dept/ID/Vendor Line/Commodity Line/Accounting Line and Posting Code for given transaction.

To calculate the Expended Amount for an individual transaction, select the sum of the Accounting Journal amounts for records with Ref Transaction Code/Dept/ID/Vendor Line/Commodity Line/Accounting Line matching the Transaction Code/Dept/ID/Vendor Line/Commodity Line/Accounting Line for given transaction. Limit selection to records with posting codes mapping to the closing classification code(s) provided in the Accrued Expenditure Closing Classification parameter.

Problem Resolution

If a problem is encountered, reschedule the Encumbrance Activity report job. The report does not change any data, and thus can be re-run as many times as needed.

2.2.8 Encumbrance Activity 2

Job Name	Encumbrance Activity 2
Recommended Frequency	On demand
Single Instance Required	No
Can be restarted?	No
Reports generated	Yes

Description

This version of the report is more intended for the Cost Accounting review whereas the other version is just Fund Accounting. In this version, for each Department, Fund, Unit, Sub Unit, Object, Program, Program Period, and Phase, the report displays the encumbrance transaction accounting to the lines with their vendor code and vendor name along with columns for the line amount, closed amount, and outstanding amount (called encumbered, paid, and outstanding). The report takes a Fiscal Year (FY) and Accounting Period (APD) as parameters and shows encumbrance lines for the given FY with APDs less than or equal to the APD parameter. The FY and APD parameters are optional; if left blank, the current application date is used to determine the current FY and APD.

The Encumbrance Activity 2 Report can be run in Open Encumbrances or All Encumbrances mode. In the Open Encumbrances mode, only encumbrances with non-zero outstanding amounts are shown. In the All Encumbrances mode, all encumbrances are shown, even those with a zero outstanding amount. In the All Encumbrances mode, the amounts listed for each encumbrance line are Encumbered Amount, Closed Amount, and Outstanding Amount. Depending on which mode is selected, an O or an A is appended to the Report ID and Open or All will precede the Report Name in the application.

The Encumbrance Activity 2 Report uses the Accounting Journal for input. The report reads Accounting Journal records for the given/defaulted FY and all accounting periods less than or equal to the given/defaulted APD parameter.

- Journal lines with a Line Function Code of 1 (Standard) with a posting code that has an Expense Budget ID equal to the Encumbrance Expense Budget Bucket ID (EXP_BKT_UPDT_ID) parameter are used to display the transactions and calculate their Encumbered Amounts.
- Transactions that reference the encumbrance line posted to the journal with one Line Function Codes of 3 (Liquidation) with a posting code that has an Expense Budget ID equal to the Encumbrance Expense Budget Bucket ID (EXP_BKT_UPDT_ID) parameter are used to calculate the Closed Amount.
- The Outstanding Amount is calculated as the difference between the Encumbered Amount and the Closed Amount.

Encumbrance activity can be filtered by department using the Department parameter. When a Department is specified, that department is appended to the Report Name in the application.

No specific transaction types or codes are selected for encumbrance activity or the closing of that activity, so the report can show transactions from the ABS, RQ, and PO transaction types. Other transaction types customized to have encumbrance activity also appear on the report.

Major Input

- Accounting Journal (JACTG / JRNL_ACTG)
- Posting Code (PSCD / R_PSCD)

Output

- Encumbrance Activity 2 Report

Parameters

Parameter	Description	Default Value
Client Name (CLIENT_NM)	An optional name to appear in the report header.	No default
Report ID (REPORT_ID)	An optional ID to appear in the report header for identification and routing.	No default
Report Label (REPORT_LBL)	Report Label appearing at the end of the Report Name if a Department Code is not specified. Optional.	No default Value
Fiscal Year (FY)	An optional selection parameter.	Current fiscal year if left blank
Accounting Period (APD)	An optional selection parameter.	Current accounting period if left blank
Department (DEPT_CD_LIST)	An optional selection parameter for one or more departments. Separate multiple values with a comma.	No default
Encumbrance Expense Budget Bucket ID (EXP_BKT_UPDT_ID)	A required selection parameter that identifies all encumbrance posting codes by the Budget Bucket ID updated as seen on the Posting Code record.	13
Report Type (REPORT_TYPE)	A required output parameter for the type of report to produce: 1 (Open Encumbrances Only) or 2 (All Encumbrances).	No default

Sort Sequence

Encumbrance records are sorted by Department, Fund, Unit, Sub Unit, Object, Program, Program Period, Phase, and Transaction Code/Dept/ID/Vendor Line/Commodity Line/Accounting Line/Vendor Code.

Selection Criteria

To select encumbrance transactions and calculate the Encumbered Amount, select Standard posting lines from the Accounting Journal for the Fiscal Year with accounting periods less or equal to the Accounting Period. Limit the selection to lines with posting codes mapped to the Expense Budget ID provided in the Encumbrance Expense Budget Bucket ID (EXP_BKT_UPDT_ID) parameter.

To calculate the Closed Amount for an individual transaction, select the sum of the Accounting Journal amounts for records with Ref Transaction Code/Dept/ID/Vendor Line/Commodity Line/Accounting Line and Posting Code matching the Transaction Code/Dept/ID/Vendor Line/Commodity Line/Accounting Line and Posting Code for the given transaction.

Problem Resolution

If a problem is encountered, reschedule the Encumbrance Activity 2 report job. The report does not change any data so it can be run again as many times as needed.

2.2.9 Open Activity Lapse Report

Job Name	Open Actv Lapse
Recommended Frequency	To be run as the end of the Budget Fiscal Year approaches, is reached, or has passed for a given amount of time. The timing of when the report is run depends on client needs, which can even vary by the type of accounting activity so that one type is lapsed before another. One batch program should be run prior to the chain. The Matching chain job so that any purchase orders that have met matching rules for being received and/or invoiced are closed out with a payment request and not lapsed. Orders with only half of the matching requirements (that is received but not invoiced or the opposite) require manual attention or they will be lapsed.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes - Open Lines To Be Lapsed

Overview

When a year comes to a close, there are many transactions still un-referenced with accounting activity that has not yet reached its final state. The Open Activity Lapse (OAL) chain job lapses the activity in the old year, leaving it up to the sites to re-establish it in the new year manually. The lapse is done by a new transaction doing a zero-dollar reference that is a final reference. This way the transaction being lapsed remains unmodified, but has the open amount reduced to zero dollars.

The primary reason for lapsing open activity is that such activity is not desired in the fiscal year at the time of annual close. The alternative to lapsing is rolling the open activity. It is a policy decision for each site to determine what rolls and what lapses. The two processes can even act upon the same type of accounting activity, with one taking certain records and leaving the other with the remainder. One such example is where all pre-encumbrances over $100K are rolled using the Roll Minimum amount first, then a lapse is performed on all those pre-encumbrances that remain.

The OAL chain lapses several types of activity that are controlled by one of four action fields found on the Parameters for Lapse Process (LPPA) and FUND pages. This report job also reports on the same types of activity because it uses the same selection logic. The report exists to provide information without performing an actual lapse. The report also functions much like an open activity report, except without an 'As of Date' and any aging.

Sites **should** run the Open Activity Lapse report, setting up the Run Mode parameter to *Report Only* for the LPPA ID. The report provides an activity count to gauge run times by displaying the open accounting lines that will be lapsed. One of the fields on the report also shows if a pending version of any final transactions to be lapsed exists. Final transactions with Pending versions are skipped by the Lapse process when in the Run Mode is *Update*.

Depending on the parameter values provided in the job, a text file is generated that contains the list of transactions that are eligible for the Lapse process and have a pending version in the system. This pending text file can further be used as input to the System Maintenance Utility to discard all of the pending transactions listed.

Note: The System Maintenance Utility job needs to be scheduled manually to discard the pending transactions before running the Lapse program to generate transactions.

Action Field	Activity
Pre Enc	Event type with a posting code defined to update the Pre - Encumbrances (12) budget amount.
Enc	Event type with a posting code defined to update the Encumbrances (13) budget amount.
Recv	Event type with a posting code defined to update the Billed Earned Revenue (22), Unbilled Earned Revenue (23), or Billed Unearned/Deferred Revenue (25) budget amount.
Other	Event type with a posting code defined to update no budget amount, while not also defined to make a Disbursement Request update. Event type without posting codes.

Such activity recorded on the Requisition (RQ), Purchase Order (PO), Accounting Based Spending (ABS), Receivable (RE), Accrued Receivable (ARE), and Travel (TRVL) transaction types can be selected based on the activity having a Close Action of *Lapse* on LPPA. If the Allow Override Action for the type of accounting activity is checked, then the Close Action at the fund code level will be used, which is found on the Fund (FUND) page. Further selection of activity is then based on settings on the Parameters for Lapse Process (LPPA) table. Each instance of the report is associated with a single record on that table that provides selection criteria.

Lapsing Receivable transactions with the lapse program does not perform any aging such as over 365 days old. There is another process for that in the Accounts Receivable chain job folder called the Generate Write Off. The OAL lapses purely on budget fiscal year.

Notes:

1. The Lapse programs currently do not support lapsing Accrued Receivable (ARE) or Stock Requisition (SRQ) transactions. .
2. Travel Authorization (TRAUTH) is the only Travel Transaction Sub Type within the Travel (TRVL) Transaction Type that can be lapsed.

Pre-Processing Tasks:

1. If a site is to control lapsing the same for all funds, then they **should** ensure that all of the override flags on SOPT are unchecked to prevent Fund lookups that will return the same value or worse, different values that are not intended.
2. For sites where the selection criteria on LPPA is not sufficient, sites **should** run with the Pre selection mode by setting up the Run Mode parameter to Pre selection and run only the first job step of the OAL process. Running the report against these tables provides a mechanism for reviewing updates to the Selected flag.
3. Negative accounting lines that are open **will not be lapsed**. They will have to be dealt with manually.
4. To ensure fewer transaction lapsing failures, sites **must** ensure the transaction codes used to lapse are established on Transaction Control (DCTRL) with the following flags checked: Approval Override Allowed and Inactive COA Codes Allowed. COA Precedence and Rollup

Precedence are recommended to be set to *None* so that new inferences or changed rollups do not prevent a lapse.

5. This batch program produces transactions which are subject to the line limit functionality constraints defined on the Transaction Component Requirements (DCREQ) in the Administration application. Sites **must** ensure that limits have not been set to lower limits than existing transactions being lapsed contain. For reference, the CBDL transaction code is in the PO transaction type and the ABDL is in the ABS transaction type.
6. All transaction codes on the Parameters for Lapse Process should have valid Automatic Transaction Numbering entries.
7. Run first in a test environment that is a recent copy of production to shake out setup problems and other system configurations that can cause lapses to fail.

Common Run Instructions:

1. Run first in a test environment that is a recent copy of production to shake out setup problems and other system configurations that can cause lapses to fail.
2. Establish parameter ID on the LPPA table. There is the option of setting this up when scheduling the chain job by using the Setup Custom Parameters link found with job step 1. However, at sites the individual that schedules the report and the one that runs the report are often different. Therefore, establishing the table records outside of the report job is recommended unless the same individual does both. This step often requires only copying the prior year's record or making changes to that record, updating the Closing BFY, Fiscal Year, Accounting Period, and Transaction Record Date on LPPA.
3. When FUND settings are used for lapsing, double check that the controls on FUND are correct for the FY value equal to the Closing BFY on LPPA.
4. A good practice is to query for any remaining open activity that should have lapsed. This can be done from a variety of sources: ledgers, journals, and the BBAL Fiscal Year Details (BBALFY) page. If ledgers are to be used, then ensure the Ledger Engine has run to ledgerize all lapsing activity. It can also be done by running the report after running an update chain.

Process Steps	Messages
1. Parameter validation	• Batch input parameters are listed. If any is invalid, an error message is issued stating such.
2. Report generation	• Reports output folder mapped (followed by pdf and html report information) • Records processed: ## or • No records processed • Rendering report started • Rendering report completed

Major Input

- Transaction Accounting Line Catalog (DOC_ACTG)
- Lapse Parameters (LP_PROC_PARM)
- Roll/Lapse Pre-Selection Detail (RLPSD / RLLP_PRE_DET)

Peripheral Input

- Fund (FUND / R_FUND)
- Posting Code (PSCD / R_PSCD)
- Event Type (ETYP / R_EVNT_TYP)
- Transaction Control (DCTRL / R_GEN_DOC_CTRL)
- Department (DEPT / R_DEPT)
- Appropriation (APPR / R_APPR)
- Appropriation Type (APTYP / R_APTYP)
- Program (PROG / R_PROG)
- Transaction Accounting (*_DOC_ACTG)

Batch Parameters

For the entire chain process, a user has to enter parameters for Job 1 – Lapse. The process propagates required parameters down the chained jobs. Some of them are non-overrideable parameters, which will be provided with the Job Setup and need not be specified for each run. Many of these are just meant for system use and should not be changed. Those are noted in the Description column.

Parameter	Description	Default Values
CLIENT_NM	Client name for “Open Lines Lapsed Report” and “Lapse Transaction Detailed/Summarized Exception Report”.	
COMMIT_SIZE	Commit Block Size is a performance parameter that controls the number of records written to the report. A value too large will hold too much data in memory before writing to the report. A value too low compared to the number of open accounting lines will cause the program to run longer with smaller more frequent updates. If no value is specified, the report will default a value of 1000.	1000
LP_PROC_PARM_ID	If you are running the job in the Financial application, you can click on the **Custom Parameter** link and select a parameter record. If you are running the job in the Administration application, enter a valid Parameter ID from the Custom Parameter table from the Financial application.	
USE_PRETABLE	Use Detail Pre-Selection Table (1= Use Pre-Selection table as input, 2= do not use Detail Pre-Selection table but use transaction catalogs.)	1

CREATE_PEND_TEXT_ FILE	Create Pending Text File (1= Create file, 2= do not create) for a later SMU rejection from workflow as the Lapse Chain will skip such a record if a pending version exists.	2
PENDING_TEXT_FILE	Pending Text File for later SMU rejection from workflow.	LapsePendDocs.txt,
AMSEXPORT	Export Location where Pending Text File would be generated.	$$AMSROOT$$/ExportImport

Custom Batch Parameters (LPPA)

Please keep in mind that when using the COA selection fields of Department, Program, Appropriation, and Appropriation Type that these elements may not exist on all event types being selected. If an event type is selected and used without one of these elements used as selection criteria on LPPA, the accounting line will not be selected.

Also, if the selection criteria are not enough to identify what is to be lapsed, then the pre selection mode is available to load the Roll/Lapse Pre Selection Summary and Detail tables for manual selection. Reports can be run to take records from the detail table and join with other information to produce a very detailed listing of potential records for selection.

Not all parameters on LPPA apply to the report job. Only those that do are shown below.

Field	Online Description	Edits
Parameter ID	Unique Parameter ID	Error issued if no value is entered.
Closing BFY	Budget Fiscal Year for selection of records to lapse. Please enter as CCYY.	Error issued if the year does not exist on R_FY. Error issued if the field is null. Error issued if multiple years are entered. In each case, the job does not run successfully and an error is issued in the log.
Mode	CVL with values-Report Only, Update, and Pre Selection	Error issued if the value is blank. In each case, the job does not run successfully and an error is issued in the log.
Event Type Selection Criteria	Enter event type(s) for selection of open activity. Commas should separate multiple event types.	• Error issued if the event type does not exist on R_EVNT_TYP. • Error if the Expense Budget Bucket ID is not 12 (pre encumbrance) or 13 (encumbrance) and the Expense Budget Bucket Update flag is checked for either posting code in Posting Pair A of the event type, unless the Expense Budget Bucket ID is listed on the Allowed Open Activity Roll Amounts (ALW_ROLL_BKTS) Application

		Parameter (APPCTRL). • Error if the Revenue Budget Bucket ID is not 22 (billed earned revenue) and the Revenue Budget Bucket Update flag is checked, unless the Revenue Budget Bucket ID is listed on the Allowed Open Activity Roll Amounts (ALW_ROLL_BKTS) Application Parameter (APPCTRL). • Error is issued if event type entered is marked as Disbursement Request Update on Event Type. • Error issued if the field is null. • In each case, the job does not run successfully and an error is issued in the log.
Transaction Codes	Optional transaction code(s) for selection of open activity. Commas should separate multiple transaction codes. This parameter can enter text up to 255 characters in size.	Error issued if the transaction code does not exist on R_GEN_DOC_CTRL. Error is issued if the Transaction code entered is not of transaction type: ABS, RE, PO, RQ, or TRVL. In this case, the job does not run successfully and an error is issued in the log.
Fund Selection Criteria	Leave blank for selection of all funds. Individual funds can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any fund code entered is not valid for the closing BFY as FY on the Fund (R_FUND) table, and job does not run successfully.
Department Selection Criteria	Leave blank for selection of all departments. Individual departments can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any department code entered is not valid on the Department (R_DEPT) table, and job does not run successfully.
Appropriation Selection Criteria	Leave blank for selection of all appropriations. Individual appropriations can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any appropriation code entered is not valid for the closing BFY as FY on the Appropriation (R_APPR) table, and job does not run successfully.
Appropriation Type Selection	Leave blank for selection of all appropriation	Error issued in log if any appropriation code entered is not valid for the closing

Criteria	types. Individual appropriations can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	BFY as FY on the Appropriation (R_APTYP) table, and job does not run successfully.
Program Selection Criteria	Leave blank for selection of all programs. Individual programs can be listed, separated by commas if multiple. (Blank) means program is not part of the selection criteria. This parameter can enter text up to 255 characters in size.	Error issued if no department code is entered and field is not blank. Error issued if more than one department code is entered and more than one program is entered. Multiple values of each are not allowed as the batch job would not know what combinations to form. Multiple runs are required in this situation. Error issued if any program code entered is not valid on the Program (R_PROG) table for the department. In each case, the job does not run successfully and an error is issued in the log.
Allow Fund Pre Enc Enc Recv Other Items Close Actions	When set to YES, an individual fund code may supply a different close action.	N/A
Pre Enc Enc Recv Other Items Lapse Threshold	A minimum dollar amount which an open accounting line or transaction must be less than for selection. Set very high to ensure lapsing of all matching open activity.	N/A
Allow Fund Pre Enc Enc Recv Other Items Lapse Threshold	When set to YES, an individual fund code may supply a different threshold.	N/A
Pre Enc Enc Recv Other Items Lapse	The setting determines at which level the threshold is applied: transaction or accounting line.	N/A

Threshold - Transaction/Line		

Major Output

- Open Lines To Be Lapsed report

Job Return Code

The following table shows the potential return codes for the Open Activity Lapse report job.

Return Code	Condition
Successful (1)	All parameters were valid and the report was generated successfully.
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 • Run time exceptions for unexpected situations When this job ends with a return code of *Failed*, after a correction, a new report can be submitted.
Terminated (16)	This return code will be issued when the job is terminated by the user. After the reason for the termination has been addressed a new report can be submitted.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. After the reason for the failure has been addressed a new report can be submitted.

Sort Criteria

- Transaction Department Code (DOC_DEPT_CD)
- Transaction Code (DOC_CD)
- Transaction ID (DOC_ID)
- Transaction Vendor Line Number (DOC_VEND_LN_NO)
- Transaction Commodity Line Number (DOC_COMM_LN_NO)
- Transaction Accounting Line Number (DOC_ACTG_LN_NO)

Selection Criteria

1. Selection criteria for obtaining Transaction Accounting lines (DOC_ACTG) to be lapsed is – where the Accounting Line value below matches the LPPA value:
    a. BFY matches Closing BFY
    b. Event Type matches one of Event Type Selection Criteria

c. Transaction Code matches one of the optional Transaction Codes
d. Fund matches one of the optional Fund Selection Criteria
e. Department (accounting line one and not Transaction Department) matches one of the optional Department Selection Criteria
f. Appropriation matches on of the optional Appropriation Selection Criteria
g. Appropriation Type matches on of the optional Appropriation Type Selection Criteria
h. Program matches one of the optional Program Selection Criteria
i. Open Amount is > $0.00

2. Get respective Closing Action by lookup to SOPT and/or FUND using Closing BFY as FY. Please see the Chain Job overview for more information on matching action to type of open activity.
3. If Action is *Lapse* then select

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If no records are selected, review the selection criteria on the LPPA ID used. Also, review the SOPT table – General tab to verify that the appropriate Action field has a value of *Lapse*.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Parameter editing was successful and report was generated successfully.	If the report reads - No Open Lines to be Lapsed, then see prior Problem Resolution text.	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 conditions. 1) Encounters any runtime exceptions and 2) Invalid parameter found 3) report layout files could not be found If the job fails because of these, investigate the problem. Submit a new report after the reason for failure has been resolved.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Reschedule a new job after the reason for termination is	N/A

		resolved.	
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. Reschedule a new job after the reason for termination is resolved.	N/A

2.2.10 Open Activity Roll Report

Job Name	Open Actv Roll
Recommended Frequency	To be run as the end of the Budget Fiscal Year approaches, is reached, or has passed for a given amount of time. The timing of when the report is run depends on a site's needs, which can even vary by the type of accounting activity so that one type is rolled before another. Two batch programs should be run before this chain. The Matching chain job so that any purchase orders that have met matching rules for being received and/or invoiced are closed out with a payment request and not rolled. The last is the Systems Assurance 15 program to ensure the open amounts found on accounting lines is accurate.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Yes - Open Lines To Be Rolled

Overview

When a year comes to a close, there are many transactions still un-liquidated with activity that has not yet reached its final state. The Open Activity Roll (OAR) chain job takes the activity in the old budget fiscal year forward into the next year through a transaction modification, often referred to as 'rolling'. When activity is rolled, the program generates the necessary transaction modification, changing the existing BFY on selected accounting lines to the 'Target Year' BFY, which is specified in parameter setup on the Parameters for Roll Process (RLPA) page.

The Open Activity and Budget Roll (OABR) chain job also rolls the accounting activity but combines that functionality with the rolling of budget authority to cover the accounting activity. When budget authority is rolled along with the accounting activity, budget availability in both the old and new years will not change.

Both chain jobs roll several types of activity that are controlled by one of four action fields found on the Parameters for Roll Process (RLPA) and FUND pages. This report job also reports on the same types of activity because it uses the same selection logic. The report exists to provide information without performing an actual roll. The report also functions much like an open activity report, except without an 'As of Date' and any aging.

Sites **should** run the Open Activity Roll report, setting up the Run Mode parameter to *Report Only* for the RLPA ID. The report provides an activity count to gauge run times by displaying the open accounting lines that will be rolled. One of the fields on the report also shows if a pending version of any final transactions to be rolled exists. Final transactions with Pending versions are skipped by the Roll process when the Run Mode is *Update*.

Depending on the parameter values provided in the job, a text file is generated that contains the list of transactions that are eligible for the Roll process and have a pending version in the system. This pending text file can further be used as input to the System Maintenance Utility to discard all of the pending transactions listed.

Note: The System Maintenance Utility job needs to be scheduled manually to discard the pending transactions before running the Roll program to generate modifications.

Action Field	Activity
Pre Enc	Event type with a posting code defined to update the Pre Encumbrances (12) budget amount.
Enc	Event type with a posting code defined to update the Encumbrances (13) budget amount.
Recv	Event type with a posting code defined to update the Billed Earned Revenue (22), Unbilled Earned Revenue (23), or Billed Unearned/Deferred Revenue (25) budget amount.
Other	Event type with a posting code defined to update no budget amount, while not also defined to make a Disbursement Request update. Event type without posting codes.

Such activity recorded on the Requisition (RQ), Purchase Order (PO), Accounting Based Spending (ABS), Receivable (RE), Accrued Receivable (ARE), Stock Requisition (SRQ), and Travel (TRVL) transaction types can be selected based on the activity having a Close Action of *Roll* on the Parameters for Roll Process (RLPA) page. If the Allow Override Action for the type of accounting activity is checked, then the Close Action at the fund code level will be used, which is found on the Fund (FUND) page. Further selection is based on the Roll Minimum of the activity. If an accounting line with the open activity equals or exceeds that minimum, it will be selected. The same override is available for the minimum on the FUND page as well. Further selection of activity is then based on settings on the Parameters for Roll Process (RLPA) table. Each instance of the report is associated with a single record on that table that provides selection criteria.

Note that Travel Authorization (TRAUTH) is the only Travel Transaction Sub Type within the Travel (TRVL) Transaction Type that can be rolled.

Pre-Processing Tasks:

1. For maximum performance when specifying RLPA selection parameters, a non-unique index could be added to the DOC_ACTG table.
2. If a site is to control rolling the same for all funds, then they should ensure that all of the override flags on SOPT are unchecked to prevent Fund lookups that will return the same value, or worse, different values that are not intended.
3. For sites where the selection criteria on RLPA is not sufficient, sites should run with the Pre selection mode by setting up the Run Mode parameter to Pre selection and run only the first job step of the OABR process. Running the report against these tables provides a mechanism for reviewing updates to the Selected flag.
4. Negative and zero-dollar accounting lines that are open will not be rolled. They will have to be dealt with manually.
5. Accounting lines with the BFY 9999 value will not be selected by the report.

Common Run Instructions:

1. Run first in a test environment that is a recent copy of production to shake out setup problems and other system configurations that can cause modifications to fail.

2. Establish parameter ID on the RLPA table. There is the option of setting this up when scheduling the chain job by using the Setup Custom Parameters link found with job step 1. However, at sites the individual that schedules the report and the one that runs the report are often different. Therefore, establishing the table records outside of the report job is recommended unless the same individual does both. This step often requires only copying the prior year's record or making changes to that record, updating the Closing BFY, Modification Fiscal Year, Modification Accounting Period, and Transaction Record Date on RLPA.
3. Double check that the controls on SOPT and FUND are correct for the FY value equal to the Closing BFY on RLPA.
4. A good practice is to query for any remaining open activity that should have rolled. This can be done from a variety of sources: ledgers, journals, and the BBAL Fiscal Year Details (BBALFY) page. If ledgers are to be used, then ensure the Ledger Engine has run to ledgerize all rolling activity. It can also be done by running the report after running an update chain.

Process Steps	Messages
1. Parameter validation	• Batch input parameters are listed. If any is invalid, an error message is issued stating such.
2. Report generation	• Reports output folder mapped (followed by pdf and html report information) • Records processed: ## or • No records processed • Rendering report started • Rendering report completed

Major Input

- Transaction Accounting Line Catalog (DOC_ACTG)
- Roll/Lapse Pre Selection Detail (RLPSD / RLLP_PRE_DET)
- Roll Parameter (RLPA / RL_PROC_PARM)

Peripheral Input

- Fund (FUND / R_FUND)
- Posting Code (PSCD / R_PSCD)
- Event Type (ETYP / R_EVNT_TYP)
- Transaction Control (DCTRL / R_GEN_DOC_CTRL)
- Department (DEPT / R_DEPT)
- Appropriation (APPR / R_APPR)
- Appropriation Type (APTYP / R_APTYP)
- Program (PROG / R_PROG)
- Transaction Accounting (*_DOC_ACTG)

*Items above with a * denote that different values may be substituted in place of the *.*

Batch Parameters

For the entire chain process, a user has to enter parameters for Job 1 – Roll Update Preprocess and then on Job 8 - Budget Roll only. The process propagates required parameters down the chained jobs. Some of them are non-overrideable parameters, which will be provided with the Job Setup and need not be specified for each run. Many of these are just meant for system use and should not be changed. Those are so noted in the Description column.

Parameter Name	Description	Default Value
CLIENT_NM	Client name to be used for report headers produced from jobs in the chain.	(blank)
RL_PROC_PARM_ID	Parameter ID to identify the RLPA record used. The Setup Custom Parameters link opens the RLPA page for record selection, update, or addition.	(blank)
USE_PRETABLE	Use Detail Pre-Selection Table (1= Use Pre-Selection table as input, 2= do not use Detail Pre-Selection table but use transaction catalogs.") If pre selection is never used, change the default value of this to 2 on BATSETUP as the parameter is often overlooked.	1
CREATE_PEND_TEXT_FILE	Create Pending Text File (1= Create file, 2= do not create) for a later SMU rejection from workflow as the Roll Chain will skip such a record is a pending version exists.	2
PENDING_TEXT_FILE	Pending Text File for later SMU rejection from workflow.	RollPendDocs.txt
AMSEXPORT	Export Location where Pending Text File would be generated.	$$AMSROOT$$/Export Import

If you are running the job in the Financial application, you can click on the Custom Parameter link and select a parameter record. If you are running the job in the Administration application, enter a valid Parameter ID from the Custom Parameter table from the Financial application.

Custom Batch Parameters (RLPA)

Please keep in mind that when using the COA selection fields of Department, Program, Appropriation, and Appropriation Type that these elements may not exist on all event types being

selected. If an event type is selected and used without one of these elements used as selection criteria on RLPA, the accounting line will not be selected.

Also, if the selection criteria are not enough to identify what is to be rolled, then the pre selection mode is available to load the Roll/Lapse Pre Selection Summary and Detail tables for manual selection. Reports can be run to take records from the detail table and join with other information to produce a very detailed listing of potential records for selection.

Not all parameters on RLPA apply to the report job. Only those that do are shown below.

Field	Description	Edits
Parameter ID	Unique Parameter ID	Error issued if no value is entered.
Closing BFY	BFY for selection of record to roll. Please enter as CCYY.	Error issued if the year does not exist on the FY page. Error issued if the field is blank. Error issued if multiple years are entered. In each case, the job does not run successfully and an error is issued in the job log.
Mode	CVL with values of: Report Only, Update, and Pre Selection	Error issued if the value is blank. In each case, the job does not run successfully and an error is issued in the log.
Event Type Selection Criteria	Enter event type(s) for selection of open activity. Commas should separate multiple event types.	Error issued if the event type does not exist on R_EVNT_TYP. Error if the Expense Budget Bucket ID is not 12 (pre encumbrance) or 13 (encumbrance) and the Expense Budget Bucket Update flag is checked for either posting code in Posting Pair A of the event type. Error is the Revenue Budget Bucket ID is not 22 (billed earned revenue), 23 (unbilled earned revenue), or 25 (billed unearned/deferred revenue) and the Revenue Budget Bucket Update flag is checked. Error is issued if event type entered is marked as Disbursement Request Update on Event Type. Error issued if the field is null. In each case, the job does not run successfully and an error is issued in the log.
Transaction Codes	Optional transaction code(s) for selection of open activity. Commas should separate	Error issued if the transaction code does not exist on

	multiple transaction codes. This parameter can enter text up to 255 characters in size and is optional.	R_GEN_DOC_CTRL. Error is issued if the Transaction code entered are not of transaction type ABS, RE, ARE, PO, RQ, SRQ, or TRVL. In this case, the job does not run successfully and an error is issued in the log.
Fund Selection Criteria	Leave blank for selection of all funds. Individual funds can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any fund code entered is not valid for the closing BFY as FY on the Fund (R_FUND) table, and job does not run successfully.
Department Selection Criteria	Leave blank for selection of all departments. Individual departments can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any department code entered is not valid on the Department (R_DEPT) table, and job does not run successfully.
Appropriation Selection Criteria	Leave blank for selection of all appropriations. Individual appropriations can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any appropriation code entered is not valid for the closing BFY as FY on the Appropriation (R_APPR) table, and job does not run successfully.
Appropriation Type Selection Criteria	Leave blank for selection of all appropriation types. Individual appropriations can be listed, separated by commas if multiple. This parameter can enter text up to 255 characters in size.	Error issued in log if any appropriation code entered is not valid for the closing BFY as FY on the Appropriation (R_APTYP) table, and job does not run successfully.
Program Selection Criteria	Leave blank for selection of all programs. Individual programs can be listed, separated by commas if multiple. (Blank) means program is not part of the selection criteria. This parameter can enter text up to 255 characters in size.	Error issued if no department code is entered and field is not blank. Error issued if more than one department code is entered and more than one program is entered. Multiple values of each are not allowed as the batch job would not know what combinations to form. Multiple runs are required in this situation. Error issued if any program code entered is not valid on the Program (R_PROG) table for the department. In each case, the job does not run successfully and an error is

		issued in the log.
Roll or Accrual Processing	The Roll or Accrual Processing field on the Parameters for Roll Process (RLPA) page drives how the Open Activity Roll process will function. If set to Roll, OAR will increment BFY on open lines. If set to Accrue, OAR will close out selected lines and create a copied line with the Accrual Event Type.	.
Accrual Event Type	The Accrual Event Type field on the Parameters for Roll Process (RLPA) page will populate the Event Type in this field on each Accounting Line that will be inserted on modification transactions as a result of running the Open Activity Roll process.	N/A
Allow Fund Pre Enc Enc Recv Other Items Close Actions	When set to YES, an individual fund code may supply a different close action.	N/A
Pre Enc Enc Recv Other Items Roll Min	A minimum dollar amount that an open accounting line or transaction must be equal to or greater than for selection. Set to $0.00 if all matching open activity should be rolled.	N/A
Allow Fund Pre Enc Enc Recv Other Items Roll Min	When set to YES, an individual fund code may supply a different minimum.	N/A
Pre Enc Enc Recv Other Items Roll Min Transaction/Line	The setting determines at which level the roll minimum is applied: Transaction or Accounting Line.	N/A

Major Output

- Open Lines To Be Rolled report

Job Return Code

The following table shows the potential return codes for the Open Activity Roll report job.

Return Code	Condition
Successful (1)	All parameters were valid and report was generated successfully.
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 • Run time exceptions for unexpected situations When this job ends with a return code of *Failed*, after a correction a new report can be submitted.
Terminated (16)	This return code will be issued when the job is terminated by the user. After the reason for the termination has been addressed a new report can be submitted.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. After the reason for the failure has been addressed a new report can be submitted.

Sort Criteria

- Transaction Department Code (DOC_DEPT_CD)
- Transactions Code (DOC_CD)
- Transaction ID (DOC_ID)
- Transaction Version Number (DOC_VERS_NO)
- Transaction Vendor Line Number (DOC_VEND_LN_NO)
- Transaction Commodity Line Number (DOC_COMM_LN_NO)
- Transaction Accounting Line Number (DOC_ACTG_LN_NO)

Selection Criteria

1. Selection criteria for obtaining Transaction Accounting lines (DOC_ACTG) to be rolled is – where the Accounting Line value below matches the RLPA value:
    a. BFY matches Closing BFY
    b. Event Type matches one of Event Type Selection Criteria
    c. Transaction Code matches one of the optional Transaction Codes
    d. Fund matches one of the optional Fund Selection Criteria
    e. Department (accounting line one and not Transaction Department) matches one of the optional Department Selection Criteria
    f. Appropriation matches on of the optional Appropriation Selection Criteria

g. Appropriation Type matches on of the optional Appropriation Type Selection Criteria
h. Program matches one of the optional Program Selection Criteria
i. Open Amount is > $0.00
j. Transaction Type is not 'JV'

2. If the respective Fund Override flag is checked, get respective Closing Action and Minimum Roll Amount by lookup to FUND using Closing BFY as FY. Please see the Chain Job overview for more information on matching action to type of open activity.
    a. If that FUND action is Roll, then select.
    b. If that Fund action is other than Roll, do not select.

Problem Resolution

It is always a good practice to look at the log of each job for warnings and errors even if the job has run successfully.

If no records are selected, review the selection criteria on the RLPA ID used. Also, review the FUND page to verify that the appropriate Action field has a value of *Roll*.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Parameter editing was successful and report was generated successfully.	If the report reads - No Open Lines to be Rolled, then see prior Problem Resolution text.	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 conditions. 1) Encounters any runtime exceptions and 2) Invalid parameter found 3) report layout files could not be found If the job fails because of these, investigate the problem. Submit a new report after the reason for failure has been resolved.	N/A
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated. Reschedule a new job after the reason for termination is resolved.	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. Reschedule a new job after the reason for termination is resolved.	N/A

2.2.11 Open Items Report

Description

The Open Items Report selects open items for requisitions, purchase orders, receivables, and payment requests as of a specific accounting period.

When to Run

On request

Major Input

- Accounting Journal (JRNL_ACTG)
  Fund (R_FUND)
  Department (R_DEPT)
  Vendor Customer (R_VEND_CUST)

Output

- Report (PDF and HTML)

Parameters

Job or Job # in a Chain	Parameter	Description	Default Values
OPEN ITEMS	Beginning Fiscal Year (BGN_FY)	Records selected equal or greater than this Fiscal year, if blank select all records. If open items are rolled or lapsed then beginning Fiscal year should be the one greater than the Budget fiscal year rolled or lapsed. If open items do not roll or lapse then beginning Fiscal year should be blank or equal to oldest transaction. If there are transactions using multi year budgets then use the oldest Fiscal year on the multi Budget year transaction or do not fill out Beginning Fiscal year. Not filling out Beginning Fiscal year can impact performance of the report.	No default.

	Block Size (BLOCK_SIZE)	Number of records to be process in each Select block. Performance tuning parameter that should be configured based on server configuration	10000
	Client Name (CLIENT_NM)	Optional field for the name of client to be appear on report	No default
	Fiscal Year (FY)	Required field for a fiscal year used as selection criteria. Must be entered as ccyy. The Fiscal Year is used as an 'as of' fiscal year.	No default
	Accounting Period (PER)	Required field for an accounting period used as selection criteria. The period is used as an 'as of' period so activity for the period and all others before it within the fiscal year is selected.	No default
	Posting Code (PSCD_CD_ID)	Required field to be used as selection criteria for what type of open activity is to be in the report. Only 1 may be entered.	No default
	Report Type (RPT_TYP)	Required field that determines the type of report to be produced. (1) Detailed – will produce a report listing individual transactions that have open lines. (2) Summary – will produce a report with just totals outstanding for each fund and department combination.	1 (detailed)
	Type of Transaction (TRAN_TYP)	Required field for selecting specific open items. Valid values are RQ, PO, PR, and RE. Only 1 value may be entered.	No default

Sort Sequence and Selection Criteria

1. Select all journal records where

   Posting Code = Posting code parameter

   Fiscal Year (FY_DC) and Accounting Period (PER_DC) <= Accounting Period which uses the Fiscal year parameter as part of its key.

Journal Attributes Needed:

Detailed Report

Doc_Typ
Doc_Dept_Cd
Doc_Cd
Doc_Id
Doc_Vend_Ln_No
Doc_Comm_Ln_No
Doc_Actg_Ln_No
Rfed_Doc_Cd
Rfed_Doc_Id
Pstng_Am
Vend_Cust_Cd (join to get Vendor/Customer name)
Fund_Cd (join to get Fund name)
Dept_Cd (join to get Department name)
Obj_Cd (or Rsrc_Cd)

Summarized Report

Doc_Typ
Doc_Dept_Cd
Doc_Cd
Doc_Id
Run_Tmdt
Rfed_Doc_Cd
Rfed_Doc_Id
Pstng_Am
Vend_Cust_Cd (join to get Vendor/Customer name)

Calculated Fields

For Detailed Report

Po_Amount
Closed_Amount
Outstanding_Amount
Dept_Po_Amount
Dept_Cls_Amount
Dept_Outstanding_Amount
Fund_Po_Amount
Fund_Cls_Amount
Fund_Outstanding_Amount
Grand_Po_Amount
Grand_Cls_Amount
Grand_Outstanding_Amount

For Summarized Report

PO_Amount

Closed_Amount

Outstanding_Amount categorized under number of Aging days in slots of 0-30 days, 31-60 days, 61-120 days, over 120 days

Vendor_Po_Amount

Vendor_Cls_Amount

Vendor_Outstanding_Amount categorized under number of Aging days in slots of 0-30 days, 31-60 days, 61-120 days, over 120 days

Grand_Po_Amount

Grand_Cls_Amount

Grand_Outstanding_Amount categorized under number of Aging days in slots of 0-30 days, 31-60 days, 61-120 days, over 120 days

2. Sort Query Object

By Fund_Cd By Dept_Cd

By Doc_Cd
By Doc_Id
By Doc_Comm_Ln_No
By Doc_Actg_Ln_No
By Rec_No

For Summarized Report

By Vend_Cust_Cd

By Doc_Cd

By Doc_Dept_Cd

By Doc_Id

By Doc_Vers_No

1. Logic for Detailed Report

   Read Sorted Records

   If Doc_Code/Doc_Id/Doc_Actg_Ln_No does **not** change

   Find Closed amount for the Actg Ln No from the reverse encumbrance transaction records (for example, PR1) where Referenced Transaction details match the above transaction details (for example, PO1) – namely:

   Referenced Transaction Id of PR1 = Transaction Id of PO1 and

   Referenced Transaction Cd of PR1 = Transaction Cd of PO1 and

   Referenced Transaction Department Cd of PR1 = Transaction Department Cd of PO1 and

   Referenced Vendor Line No of PR1 = Vendor Line No of PO1 and

   Referenced Commodity Line No of PR1 = Commodity Line No of PO1 and

   Referenced Accounting Line No of PR1 = Accounting Line No of PO1 and

   Referenced Posting Line No of PR1 = Posting Line No of PO1 and

   Posting Code Id of PR1 = Posting Code Id of PO1

   Calculate Outstanding_Amount = Po_Amount – Closed_Amount

   If Outstanding_Amount > 0 (means there is an outstanding amount)

   Then

   Print Detail Line

   Update Totals for Department, Fund, and Grand for PO AMOUNTS, CLS AMOUNTS, and GRAND TOTAL AMOUNTS

   Else

   Skip to next Sort Record

   Update Totals for Department, Fund, and Grand for PO AMOUNTS, CLS AMOUNTS, and GRAND TOTAL AMOUNTS,

   If there is a change in Department – Print Department Totals (PO, Closed, Outstanding)

   If there is a change in Fund – Print Fund Totals (PO, Closed, Outstanding)

   If all records have been read – Print Grand Totals (PO, Closed, Outstanding)

2. Logic for Summarized Report

   Read Sorted Records

   If Doc_Code/Doc_Id does **not** change

   Find Closed amount for the Transaction Id from the reverse encumbrance transaction records (for example, PR1) where Referenced Transaction details match the above transaction details (for example, PO1) – namely:

   Referenced Transaction Id of PR1 = Transaction Id of PO1 and

   Referenced Transaction Cd of PR1 = Transaction Cd of PO1 and

   Referenced Transaction Department Cd of PR1 = Transaction Department Cd of PO1 and

Posting Code Id of PR1 = Posting Code Id of PO1

Calculate Outstanding_Amount = Po_Amount – Closed_Amount

If Outstanding_Amount > 0 (means there is an outstanding amount)

Then

Calculate the number of Aging Days as difference in the number of days between Transaction run(posted) date time and End date of the As of accounting period.

Print Detail Line.

Update Totals for Vendor and Grand for PO AMOUNTS, CLS AMOUNTS, and GRAND TOTAL AMOUNTS

If Number of Aging days is between 0 to 30

Update the Totals for Vendor and Grand Outstanding Amount for 0-30 days slot

If Number of Aging days is between 31 to 60

Update the Totals for Vendor and Grand Outstanding Amount for 31-60 days slot

If Number of Aging days is between 61 to 120

Update the Totals for Vendor and Grand Outstanding Amount for 61-120 days slot

If Number of Aging days is over 120

Update the Totals for Vendor and Grand Outstanding Amount over 120 days slot

If there is a change in Vendor – Print Vendor Totals (PO, Closed, Outstanding)

If all records have been read – Print Grand Totals (PO, Closed, Outstanding)

Please note that following applies for the amounts displayed on Detailed as well as Summarized Reports:

- PO/PR/RQ/RE Amount:

  The amounts shown here are in terms of debits and credits, not accounting line amounts. Therefore, for revenue accounting lines, a negative amount represents a credit (or increase) to revenue that would have resulted from a positive accounting line amount. The opposite is true for a negative accounting line amount in that it is a debit (decrease) to revenue. For spending transactions, positive accounting line amounts are generally debits (increases), with negative accounting line amounts generally crediting an account (decreasing it).

- Closed Amount:
  The amounts shown here are in terms of debits and credits, not accounting line amounts from referencing (liquidating) transactions. Therefore, for revenue accounting lines, a positive amount represents a debit (or decrease) to revenue that would have resulted from a positive, referencing accounting line amount. The opposite is true for a negative accounting line amount in that it is a credit (increase) to revenue. For spending transactions, positive, referencing accounting line amounts are generally credits (decreases), with negative accounting line amounts generally debiting an account (increasing it).
- Outstanding Amount:
  The amounts shown here are in terms of debits and credits, not accounting line amounts from referenced or referencing transactions. Therefore, for revenue accounting lines, a negative amount represents a credit balance (normal balance for revenue). The opposite is true for a

positive amount which would be a debit balance for revenue. For spending transactions, positive amounts are generally debit balances (normal balance for encumbrances, etc), with negative accounting line amounts are generally credit balances.

Format

Detailed Report

- Columns for this report are: Transaction Code, Transaction Department Code, Transaction ID, Commodity line no, Accounting line no, Vendor, Object, PO Amount, Closed Amount, and Outstanding Amount.
- In the header, the system displays the caption Fund followed by the Fund Code and Fund Description
- In the header, the system displays the caption Department followed by the Department Code and Department Description.
- Also, there is a subtotal number and a page break when there is a break in Fund and Department.

Summarized Report

- Columns for this report are: Transaction Code, Transaction Department Code, Transaction ID, Vendor, Run_Tmdt, PO Amount, Closed Amount, and Outstanding Amount.
- Summarized report will display $0 as outstanding amount for transactions recorded with future date until after that future date.
- In the header, the system displays the caption Vendor followed by the Vendor Code and Vendor Description.

Actual HTML Report Format

The output of Detailed Open Report in HTML format will look like the following

CGI Advantage FINANCIAL OPEN ITEM REPORT

RUN DATE: 3/3/2001 PAGE 1

Outstanding Purchase Orders Through Accounting Period as of 8/2001

FUND: 101 General Fund
DEPT: 150 Department of Transportation

Transaction Code	Transaction Dept Cd	Transaction Id	Comm Line	Acct Line	Vendor	Object	PO	Amount	Closed	Amount Outstanding
DO	150	PT1	1	1	Alpha	7020	500.00	0.00	500.00	
PO	150	2way 1	1	1	Alpha	7020	100.00	0.00	100.00	
PO	150	PT2	1	1	Alpha	7020	500.00	0.00	500.00	
PO	150	2way 2	1	1	Alpha	7020	100.00	50.00	50.00	
PO	150	PT3	1	1	Alpha	7020	500.00	50.00	450.00	
PO	150	2way 3	1	1	Alpha	7020	100.00	50.00	50.00	
PO	150	PT4	1	1	Alpha	7020	500.00	50.00	250.00	
PO	150	2way 4	1	1	Alpha	7020	100.00	50.00	50.00	
PO	150	PT5	1	1	Alpha	7020	500.00	50.00	450.00	
PO	150	2way 5	1	1	Alpha	7020	100.00	50.00	50.00	

TOTAL FOR DEPT	300.00	350.00	2650.00	
TOTAL FOR FUND	300.00	350.00	2650.00	
GRAND TOTAL FOR DEPT	300.00	350.00	2650.00	
GRAND TOTAL FOR FUND	300.00	350.00	2650.00	

American Management Systems

The output of Summarized Open Report in HTML format will look like the following

CGI Advantage FINANCIAL OPEN ITEM REPORT

RUN DATE: 3/3/2001 **PAGE 1**

Aged Purchase Orders Through Accounting Period as of 9/2001

VENDOR: 0300272163 Alpa Associates

Transaction Code	Transaction Dept Cd	Transaction Id	PO Date	PO Amount	Cleared Amount	Outstanding Amount		
					0-30Days	31-60Days	61-120Days	Over 120Days
DO	150	PT1	3-3-2001	200.00	90.00	110.00	0.00	0.00	0.00
PO	150	PO1	2-3-2001	200.00	90.00	0.00	110.00	0.00	0.00
PO	150	PO2	1-3-2001	200.00	90.00	0.00	0.00	110.00	0.00
PO	150	PO3	8-5-2000	200.00	90.00	0.00	110.00	0.00	110.00
PO	150	P04	3-3-2001	200.00	90.00	110.00	0.00	0.00	0.00
PO	150	PO5	2-3-2001	200.00	90.00	0.00	110.00	0.00	0.00
PO	150	PO6	1-3-2001	200.00	90.00	0.00	0.00	110.00	0.00
PO	150	PO7	8-5-2000	200.00	90.00	0.00	110.00	0.00	110.00
PO	150	P08	3-3-2001	200.00	90.00	110.00	0.00	0.00	0.00
PO	150	PO9	2-3-2001	200.00	90.00	0.00	110.00	0.00	0.00
PO	150	P11	1-3-2001	200.00	90.00	0.00	0.00	110.00	0.00
PO	150	P12	8-5-2000	200.00	90.00	0.00	110.00	0.00	110.00
Vendor Total				2400.00	120.00	330.00	330.00	330.00	330.00
Grand Total				1200.00	120.00	330.00	330.00	330.00	330.00

2.2.12 Rebuild Balance Sheet Balance Information

Chain or Job Name	Rebuild BBAL
Recommended Frequency	Only run when R_BBAL_ITD and R_BBAL_FY_DETAILS balance tables need to put back in sync after a System Assurance 2 reported condition.
Single Instance Required	Yes, and with no System Assurance 2 or Ledger Engine batch jobs running concurrently or any online or offline transaction processing happening when this job is running.
Can be restarted?	No
Reports generated	Yes – Balance Sheet Detail Records with No SA_NONBUD Entries

Overview

The Rebuild BBAL job is intended to provide a batch method for correcting data that has become out of sync with historical information recorded through the Accounting Journal for Balance Sheet Balance Detail, ITD Balance Sheet Detail (BBALD / R_BBAL_ITD), ITD Balance Sheet Summary (BBALS / R_BBAL_ITD_SMRY), and Fiscal Year Balance Sheet Detail (BBALFY / R_BBAL_FY_DETAILS). The program updates each independently and from different data sources. BBALFY is synchronized from an input ledger. BBALD is synchronized from the temporary table populated by the Systems Assurance 02 job. BBALS is synchronized from an updated BBALD.

Please note, when records are inserted into any table, certain common system edits are not invoked. One reason is that COA involved may not be valid in the current year. Another is that since historical information based on the Accounting Journal is being used for the rebuilds, all COA data was valid at some point. However, all updates invoke logic that pushes updates to parent tables.

The following actions are taken when rebuilding BBALD:

1. Reset the Debit Amount (DR_AM_TOT) and Credit Amount (CR_AM_TOT) of BBALD with values from SA_NONBUD.
2. Log BBALD records into the Balance Sheet Detail Records with 'No SA_NONBUD Entries' when a corresponding SA_NONBUD record is not found. If the Debit Amount (DR_AM_TOT) and Credit Amount (CR_AM_TOT) is not $0.00 on BBALD for such a record, the program will reset the amount to $0.00.
3. Add a new record for SA_NONBUD activity when one does not already exist. Save the newly inserted record to invoke certain business rules processing for inference and formula calculation of other fields on BBALD.
4. For any SA_NONBUD record found that does not exist on BBALD, the program will insert a BBALD record for that historical activity.
5. When an update happens to a BBALD record, the Balance (CURR_BAL) is recalculated to reflect the update.
6. When an update happens to a BBALD record, the CURR_BAL is pushed up to the corresponding field on any parent table that connects to the account. The full list of these optional tables in order of update are:

R_BSA_CBAL → R_CBAL → R_CBAL_SMRY → (if used) R_CBAL_POOL

R_BSA_FBAL → R_FBAL → R_FBAL_SMRY

The following actions are taken when rebuilding BBALS:

1. Reset the Debit Amount (DR_AM_TOT) and Credit Amount (CR_AM_TOT) with the sum of all children records on BBALD.
2. Set the Debit Amount (DR_AM_TOT) and Credit Amount (CR_AM_TOT) with $0 if there are no children records found on BBALD.
3. Add a new record if there are BBALD children records with no parent.
4. Save the newly inserted record to invoke certain business rules processing for inference and formula calculation of other fields on BBALS.

The following actions are taken when rebuilding BBALFY:

1. Purge BBALFY.
2. Rebuild the BBALFY from the ledger provided as input, setting the Debit Amount (DR_AM_TOT) amount to positive sum of the ledger records and the Credit Amount (CR_AM_TOT) set to negative sum of ledger records. As a result of this fix, rebuilt BBALFY records will not necessarily have the total debits and credits that occurred in the FY for an account, but the Balance will be correct.
3. For all records sort by FUND, BSA, Sub FUND, Sub BSA, and FY. Then set the Beginning Balance (BGN_BAL_TOT) on each record with the Balance (CURR_BAL) of the previous FY record. If there is a gap of a year, then a new record is inserted to account for the fiscal year in the gap.
4. Save the newly inserted record to invoke certain business rules processing for inference and formula calculation of other fields on R_BBAL_FY_DETAILS.

Important Run Instructions

It is important to run the SA02 in full mode before the Rebuild BBAL chain job and run these jobs in the following sequence:

1. Journal Posting Initiator
2. Journal Engine
3. Ledger Engine
4. SA02 Full Mode
5. Rebuild BBAL
6. SA02 Incremental Mode

Data contained in SA_NONBUD and the specified ledger is accurate. Accurate data in input sources will result in accurate data in the rebuilt balance tables. The input ledger to SA2 and the one for this conversion program should be completely up-to-date in order for a timely and accurate rebuild.

A. All posting lines of transactions with a Final and Historical Final Phase should be journalized.
B. Run Journal Posting Initiator.

C. Run Journal Engine.
D. Check for PSTNG_LN_CAT where JRNL_PSTNG_IND is 1 or 2 and DOC_PHASE is 3 or 5.
E. All journal records should be ledgerized.
F. Run Ledger Engine in Normal, Gap, and Failed Work Modes.
G. Check JRNL_LOG for PROC_ID = LDGRPOST where ST_FL = 4, 9, and 10.
H. Finally, run SA3 against the ledger which should show debits equal credits for FY, APD, and Fund level.

- In choosing the ledger parameter, the ledger must contain FY, Fund, Sub Fund, BSA, and Sub BSA at a minimum. It can contain other information, but other information on real accounts only slows the rebuild.
- When this batch process is in execution, there should be no System Assurance 2 or Ledger Engine batch jobs running concurrently. This is to make sure that no data changes are made concurrently on input data sources of SA_NONBUD and the specified Ledger while the batch process is in the process of reading its data. Further there should be no online or offline transaction processing occurring concurrently with the batch process execution.
- Updates to BBALFY will default names and other fields from the FUND, SFUND, BSA, and SBSA tables using the FY of that record. If the COA table record no longer exists, then the name field will be blank. Not finding a COA table record is more serious. There has to be a BSA record or the account type cannot be set. To find if there are any missing, run the following SQL's. Any records found missing will have to be added.

To find balance sheet accounts not valid in a given FY:
SELECT FY, BSA_CD FROM R_BBAL_FY_DETAILS A WHERE NOT EXISTS (SELECT 1 FROM R_BSA B WHERE A.FY=B.FY AND A.BSA_CD=B.BSA_CD)

To find funds not valid for a given FY:
SELECT DISTINCT FY, FUND_CD FROM R_BBAL_FY_DETAILS A WHERE NOT EXISTS (SELECT 1 FROM R_FUND B WHERE A.FY=B.FY AND A.FUND_CD=B.FUND_CD)

To find sub balance sheet accounts not valid in a given FY:
SELECT FY, BSA_CD, SBSA_CD FROM R_BBAL_FY_DETAILS A WHERE NOT EXISTS (SELECT 1 FROM R_SBSA B WHERE A.FY=B.FY AND A.BSA_CD = B.BSA_CD AND A.SBSA_CD= B.SBSA_CD) AND A.SBSA_CD != 'BLNK'

To find sub funds not valid in a given FY
SELECT DISTINCT FY, FUND_CD, SFUND_CD FROM R_BBAL_FY_DETAILS A WHERE NOT EXISTS (SELECT 1 FROM R_SFUND B WHERE A.FY=B.FY AND A.FUND_CD=B.FUND_CD AND A.SFUND_CD = B.SFUND_CD) AND A.SFUND_CD != 'BLNK'

- Updates to BBALD will default names and other fields from FUND, SFUND, BSA, and SBSA using the current FY (derived from Application Date). If a reference table record does not exist, then the name field will be blank. Not finding an R_BSA record is more serious. There has to be a record or the account type cannot be set. To find if there are any missing, then run the following SQL. Any records found missing in the current year will have to be added.

SELECT DISTINCT BSA_CD FROM LDGR_FYDAD A WHERE NOT EXISTS (SELECT 1 FROM R_BSA B WHERE A.BSA_CD=B.BSA_CD AND B.FY=####)

Substitute current year in place of the #### and change the ledger if another is used as the input source to SA02. Ignore the line returned with a blank BSA_CD. That is a single distinct record for all nominal accounts on the ledger.

- Save a copy of the following tables for reference before executing this data program for later comparison and verification:
  - R_BBAL_FY_DETAILS
  - R_BBAL_ITD
  - R_BBAL_ITD_SMRY
  - R_BSA_CBAL
  - R_CBAL
  - R_CBAL_SMRY
  - R_CBAL_POOL
  - R_BSA_FBAL
  - R_FBAL
  - R_FBAL_SMRY

Process Steps	Messages
Parameter Validation	• Parameters are listed • If the parameter is invalid, an error message will be displayed
Logging BBALD records to Report	• Reports output folder mapped (followed by pdf and html report information) • No records on R_BBAL_ITD whose amounts do not match amounts on corresponding SA_NONBUD or • Finished with logging R_BBAL_ITD records that do not have a corresponding entry in SA_NONBUD, Logged: ### • Rendering report started • Rendering report completed • Report Generated successfully
Update BBALD amount fields	• No records on R_BBAL_ITD whose amounts do not match amounts on corresponding SA_NONBUD or • Finished with all R_BBAL_ITD records whose amounts do not match amounts on corresponding SA_NONBUD, Total: ###
Insert new BBALD records	• No records on SA_NONBUD that does not have a corresponding R_BBAL_ITD record or • Finished with SA_NONBUD records that do not have a corresponding R_BBAL_ITD record, Total: ###

Purge BBALFY records	• Number of R_BBAL_FY_DETAILS entries deleted: ###
Insert BBALFY records	• Number of R_BBAL_FY_DETAILS entries inserted: ###
Update BBALFY records	• Number of R_BBAL_FY_DETAILS entries updated: ###
Insert Gap BBALFY records	• If an error occurs during the execution of business rules then a message is displayed that an unknown error has occurred, BSA_TYP_IND and NORM_BAL_IND not set properly during save. This error occurs because the Fiscal Year balance sheet detail record is not able to find its parent Balance Sheet Account (R_BSA) record for its Fiscal Year (FY) and BSA (BSA_CD).
Update BBALS records w/no children to $0	• Number of R_BBAL_ITD_SMRY records with no children is ###
Insert BBALS records	• Number of R_BBAL_ITD_SMRY records inserted is ###
Update BBALS records w/BBALD amount totals	• Number of R_BBAL_ITD_SMRY records updated is ###

In all update/insert/delete cases above, if failure occurs during commit then a message will be displayed to inform the cause of failure.

Major Input

- LDGR_FYDAD (or other chosen)
- SA_NONBUD

Peripheral Input

- R_BSA
- R_SBSA
- R_FUND
- R_SFUND
- IN_APP_CTRL
- R_CLDT

Batch Parameters

Parameter Name	Description	Default Value
COMMIT_SIZE	Required commit block size for the number of processed lines before a	1000

	commit is performed. This is a performance tuning parameter. The value can be tuned as per system configuration and its processing capabilities. A value too low can increase the number of database commits and a value too high can cause out of memory issues.	
INPUT_SRC	Required Ledger ID to rebuild BBALFY. Use a Ledger that summarizes on FY, Fund, Sub Fund, BSA and Sub BSA at a minimum. Apart from these, other values can also be used for summarization; however, it will cause the batch job to perform more summarization and take longer.	(blank)

Major Output

- Fiscal Year Balance Sheet Detail (BBALFY / R_BBAL_FY_DETAILS)
- ITD Balance Sheet Detail (BBALD / R_BBAL_ITD)
- ITD Balance Sheet Summary (BBALS / BBAL_ITD_SMRY)
- Cash Balance Detail Maintenance (CBALD lower half / R_BSA_CBAL
- Cash Balance Detail Query (CBALDQ / R_CBAL)
- Cash Balance Summary Query (R_CBAL_SMRY)
- Cash Balance Pool (CBALPQ / R_CBAL_POOL)
- Fund Balance Detail Maintenance (FBALD lower half / R_BSA_FBAL
- Fund Balance Detail Query (FBALDQ / R_FBAL)
- Fund Balance Summary Query (FBALSQ / R_FBAL_SMRY)
- Balance Sheet Detail Records with No SA_NONBUD Entries

Job Return Code

Return Code	Condition
Successful (1)	BBALD, BBALS, and BBALFY were updated as necessary.
Warning (4)	If no records are found from the input ledger to be inserted into the Fiscal Year Balance Sheet Detail table then a message will be displayed that no records were found.
Non-Fatal (8)	Job does not return this code.
Failed (12)	This return code will be issued under the following conditions: • Parameter validation failed.

	• Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

See Selection Criteria below

Selection Criteria

- Select R_BBAL_ITD records that have corresponding records on SA_NONBUD when grouped together by Fund, Sub Fund, BSA, and Sub BSA and the sum of debited and credited amount on these records does not equal the debited and credited amounts on R_BBAL_ITD records.
- Select records from the Ledger indicated in the "Source Ledger to build FY Balance Sheet detail from" batch parameter value and where BSA is not empty and group them by Fiscal Year, FUND, BSA, Sub FUND, and Sub BSA.
-

Problem Resolution

The report listing BBALD records that exist but have never been used is provided for one reason. Only one system functionality exists where a BBALD record would exist with no SA_NONBUD record – definition on Cash (CBALD) or Fund Balance Detail Maintenance (FBALD) of an account that is not already on BBALD. In such a case, the system creates a zero-dollar BBALD record. Another possible reason for such a situation would be a conversion routine that inserted a zero-dollar record. Outside of the first reason, such a record is something that would not be created by the application. These records can be deleted if there is not a corresponding record on R_BSA_CBAL or R_BSA_FBAL by a database utility. They can also be left as they do no harm.

The following table shows the possible return codes and recommendations.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the parameters are validated successfully. All updates necessary, occurred successfully.	N/A	N/A
Warning (4)	Number of records in the input ledger to be inserted into BBALFY is zero. The ledger is either empty or does not contain necessary COA elements or	Run again with the correct input ledger.	N/A

	FY.		
Non-Fatal (8)	Job does not return this code.	N/A	N/A
Failed (12)	Job failed due to parameter validation.	Run another job with the correct parameters.	N/A
	Job failed due to run time exceptions.	Investigate the exception reported by the process, resolve the error and schedule a new job.	N/A
Terminated (16)	Job was terminated by a user.	Resolve the reason for the termination and 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 and a new job scheduled.	N/A

2.2.13 Revenue and Expense by Fund Report

Job Name	Revenue and Expense by Fund
Recommended Frequency	Frequency depends on need, but often run before and after month and annual closes. Certain other jobs that should not be run while the report is running: Ledger Engine
Single Instance Required	Yes
Can be Restarted?	Yes
Reports Generated	Revenue and Expense by Fund Report

Overview

The Revenue and Expense by Fund reports the revenues and expenses by fund for the specified period along with YTD and net gain/loss information for each fund. ***Note: The report looks for specific COA and roll-up values in selecting and determining how to classify expense and revenues. This is likely to limit the use of the report.*** The specific values used are listed subsequently.

The Parameters for Revenue and Expense by Fund (RVEXFNPR) page provides the selection criteria like FY and APD for the report. The report can be run for a fund or other roll-ups of fund. However, the report will fail if the selection criteria for fund has a fund code that does not have any Fund Group specified.

For each fund, the revenues and expenses are grouped into Miscellaneous, Revenues, Expenses and Other Financial Sources and Uses sections. The Other Financial Sources and Uses section is further broken down into Transfers In and Out while other sections are broken down by Category and Class roll-ups. If a particular section does not have any records it is not displayed.

Important Run Instructions

Update of Input Sources

As with any report that uses a ledger as an input source, those input sources have to be up-to-date for the report to be accurate. Journal records not ledgerized are often because of a failed unit of work from a prior Ledger Engine job. In addition to running the Ledger Engine in ‘normal’ mode, it should also be run in ‘process failed work’ and ‘process logged gap’ modes to ensure all journal records have been ledgerized.

Input Ledger Match

As the report can be executed with various levels of Fund detail, the input ledger used must contain the Fund detail necessary to support the report. The input ledger must contain values in the Accounting Period column to support the report. When the input ledger contains a level of date, COA, or other detail not be used in the report, such details serve only to slow production of the report. Sometimes, this cannot be prevented as the number of ledgers and summarization levels are fixed and limited. When running the report, the ‘best match’ ledger should be used. For example, a Post Closing Trial Balance report would likely be at a very high level of detail and thus require the same ledger that was input into annual close.

Restartability Information

The job can be restarted however the parameters cannot be modified.

Progression Messaging

Whenever miscellaneous records are encountered the system logs the following entries:

- A Miscoded Transaction has been caught for fund: <fund code>; CurrAm: <Am>; YTDAm: <Am>
- A Revenue with No Revenue Source exists for fund: <fund code>

For more details on Miscellaneous records, please refer to the Selection and Sorting section.

Major Input

- Input Ledger – likely the APD Accounting Ledger (LDGR_APD_ACTG) but could be another to provide beginning balance amounts
- Posting Code (R_PSCD) – provides the connection from ledger and journal records to a posting code closing classification so that can be mapped to a closing classification in the report. Also provides names for posting codes in the report modes that display posting codes.
- Parameters for Revenue and Expense by Fund (R_RVEXP_CUSTOM_PARM) - supplies the selection and summarization criteria.

Minor Input

- Journal/Ledger Control Detail (R_JRNL_LDGR_CTRL) used to determine the source journal of the Input Ledger
- COA reference tables supply names for codes (for example, R_FUND, R_RSRC, R_OBJ, R_FCLS)

Batch Parameters

Parameter	Description	Default Value
Client Name	Optional Client Name for report	None
Parameter ID	Required Parameter ID from the Parameters for Revenue and Expense by Fund (RVEXFNPR) table	None
Report ID	Optional Report ID for report for the optional report distribution	None

Custom Parameters (RVEXFNPR page)

Parameter	Description	Default Value
General Information		
Parameter ID	A required unique alpha-numeric identification assigned to a set of	None

	parameters,	
Fiscal Year	An optional parameter used for journal and ledger record selection. If left blank, it defaults from any As of Date entered. If that date is left blank, the year is the default FY for the Application Date when the report is run.	None
Accounting Period	An optional parameter used for journal and ledger record selection of 'current period activity'. If left blank, the default is year and APD for the Application Date when the report is run.	None
As of Date	An optional parameter.	None
Summarization Level		
Fund Dimension	A required level of Fund used for summarization. The default is Fund. When choosing a level, the input ledger must contain that level for a meaningful report.	Fund
Fund Dimension Value	An optional parameter used for ledger record selection. Multiple values are allowed if comma-separated. Values entered here are used in conjunction with the Fund Dimension choice. If left blank then all values in the Fund Dimension are reported.	None
Report Source		
Journal ID	The required input journal to be used for the current period record selection. The Accounting Journal is the default and the most likely input source.	1
Ledger ID	The required input ledger to be used for current period record selection. Please refer the "Important Run Instructions" section in this run sheet for more information on input ledgers.	11

Major Output

- Revenue and Expense by Fund Report

Note: Every time the job is run a new folder gets created with Job ID for each instance of the report underneath the parent folder RvExpReport. Inside this new folder, another folder gets created with RvExpReport inside which resides the report with name RvExpReport.pdf.

Job Return Codes

The following table shows the potential job return codes for the Revenue and Expense by Fund job.

Return Code	Condition
Successful (1)	All parameter validations were successful and report was produced with records.
Warning (4)	Job does not end with this status.
Non-Fatal Error (8)	Job does not end with this status.
Failed (12)	The job fails under the following conditions: • Parameters are invalid • Report layout file could not be found • Run time exceptions for unexpected situations. • Fund Group not found for a Fund
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.

Selection Criteria & Sort Sequencing

- Fund and Fund Dimension Value match journal or ledger records
- Fiscal Year (FY_DC) = Fiscal Year parameter
- Accounting Period (PER_DC) <= Accounting Period parameter

Group By:

- Revenues

  Posting Code has a closing classification of 14 (Collected Revenue) or 15 (Billed Revenue) excluding the following Revenue Sources:4744, 4745, 4746, 4774, 4775, 4776, 4777, 4778, 4779, 4780, 4781, 4782, 4785.

- Expenditures

  Posting Code has a closing classification of 10 (Cash Expense) or 11 (Accrued Expense) excluding the Object Category: TR and Objects: 7312 and 7313.

- Proceeds

  Posting Code has a closing classification of 14 (Collected Revenue) or 15 (Billed Revenue) and Revenue Sources: 4775, 4776, 4777

- Transfers-In

  Posting Code has a closing classification of 14 (Collected Revenue) or 15 (Billed Revenue) and Revenue Source: 4744, 4745, 4746

- Transfers-Out

Posting Code has a closing classification of 10 (Cash Expense) or 11 (Accrued Expense) and Object Category: TR

- Other Uses

  Posting Code has a closing classification of 14 (Collected Revenue) or 15 (Billed Revenue) and Revenue Source: 4774, 4778, 4779, 4780, 4781, 4782, 4785 OR

  Posting Code has a closing classification of 10 (Cash Expense) or 11 (Accrued Expense) and Object: 7312, 7313

Miscellaneous

Remaining records are grouped under this heading. However, this section is displayed first.

Order By (ascending unless specified):

- Fund Dimension

Calculations

- Current Period

  Get sum (AM) from input ledger records where

  Fiscal Year (FY_DC) = Fiscal Year parameter and

  Accounting Period (PER_DC) = Accounting Period parameter

- Year-To-Date

  Get sum (AM) from input ledger where

  Fiscal Year (FY_DC) = Fiscal Year parameter and

  Accounting Period (PER_DC) <= Accounting Period parameter

Problem Resolution

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

Parameter Validation

Possible Return Codes	Condition	Recommendation	Other instructions
Successful (1)	Successful	N/A	N/A
Failed (12)	Required Parameters are not entered. **Sample Message:** Parameter ID required	Run a new instance of the engine with correct parameters or restart the current one and make parameter changes.	

Report Production

Possible Return Codes	Condition	Recommendation	Other instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	Failed because of runtime exceptions for unexpected situation	Failure reason needs to be investigated before scheduling a new report.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the system failure needs to be investigated before scheduling a new job.	

2.2.14 Trial Balance Report

Job Name	Trial Balance
Recommended Frequency	Frequency depends on need, but often run before and after month and annual closes. Certain other jobs should not be run while the report is running: Journal Engine or Ledger Engine
Single Instance Required	No
Can be Restarted?	In certain circumstances
Reports Generated	Trial Balance Report

Overview

The Trial Balance report lists all the account balances at many different levels of detail, as determined by parameter settings, from an input ledger and journal. To control that level of detail and provide selection criteria, there is a table - the Parameters for Trial Balance (TBPA) table – where multiple parameter ID records are established to produce different types of reports for different time periods. A later section in this run sheet will detail the options on that TBPA table and what functionality each provides.

The report can go by several different names depending on the level of detail and when it is run in the closing process. Before any adjustments are made, the report is simply referred to as a **Trial Balance** or **Unadjusted Trial Balance** when summarization levels are established at a low level. After month-end or year-end adjustments are made, the report is often called the **Adjusted Trial Balance**. Once an Annual Close has been performed, the report can be referred to as a **Post-Closing Trial Balance** when run against the fiscal year just closed. Any one of the previous names can then have **Detail** or **Summary** added to the beginning to create eight more names. The difference between these two is what is implied: how much detail is presented.

Important Run Instructions

Update of Input Sources

As with any report that uses a ledger or journal as an input source, those input sources have to be up-to-date for the report to be accurate. All posting lines should be journalized either by real-time journal posting or running the Journal Engine batch job. The most common occurrence of posting line not being journalized is one for a transaction that uses the Asynchronous Posting option found on the Transaction Control (DCTRL) table. These transactions require the Journal Posting Initiator job and the Journal Engine job to be journalized. Journal records not ledgerized are often because of a failed unit of work from a prior Ledger Engine job. In addition to running the Ledger Engine in ‘normal’ mode, it should also be run in ‘process failed work’ and ‘process logged gap’ modes to ensure all journal records have been ledgerized.

Input Ledger Match

As the report can be executed with various levels of chart of account (COA) detail, the input ledger used must contain the COA necessary to support the report. If not, the Beginning Balance column will not calculate properly. Most ledgers have the Real Account Level setting on the Journal/Ledger Control (JLCTRL) table set to ‘Fund and BSA Levels. Such a setting satisfies the desire to summarize real accounts to a much higher level than nominal accounts. The levels defined in the Fund and BS Account summarization setting fields will be

used for real accounts. However, such a ledger will not be able to provide any details for the Organizational Dimension setting on the TBPA page. If a report with organizational details is desired, the user must choose a ledger with the Real Account Level set to 'All Levels'.

The input ledger must contain values in the Accounting Period column to support the report. The report can also use the As of Date from the TBPA for ledger record selection. Running the report with an As of Date in the past may result in ledger record amounts which represent one or more journal records processed after that As of Date. If an input ledger contains Record Date, then selection will compare that date to compare to the As of Date from TBPA. If not, then ledger record selection will just be based on accounting period. Having Record Date on the input ledger provides a more accurate report when running for a past period of time because ledger records selection will use the As of Date just as journal record selection will.

When the input ledger contains a level of date, COA, or other detail that will not be used in the report, such details will serve only to slow production of the report. Sometimes this cannot be prevented as the number of ledgers and summarization levels are fixed and limited. When running the report, the 'best match' ledger should be used. For example, a Post Closing Trial Balance report would likely be at a very high level of detail and thus require the same ledger that was input into annual close.

- Prior Fiscal Year Not Closed

Calculation of the Beginning Balance amount on the report only selects records that match the TBPA selection Fiscal Year and have an accounting period less than the TBPA selection Accounting Period. **If the fiscal year prior to that of a Trial Balance report has not been closed, balances from that prior year will not have rolled into accounting period 0 of the report year.** Thus, the Beginning Balance amount will not reflect an inception-to-date balance for an account until a close is done on the prior year.

Descriptions of the Various Report Modes

The following section presents details on how each mode of the report will differ in details. To understand these levels, it is important to be familiar with the concepts of Posting Code and Posting Code Closing Classifications. Please see the General Accounting Users Guide for more details on these concepts. The TBPA table has the Closing Classifications section for definition of eight different Closing Classifications, which are at a higher level than Posting Code Closing Classifications. These Closing Classifications, and their assigned names, are used in all report modes as the primary grouping of report details within COA levels.

Within each report mode, additional levels of detail are available beneath the report mode values. Settings in the Object, Revenue Source, or BSA Dimension fields found in the Summarization Level section from the TBPA determine what that additional level is and if that additional level should exist. When one or more of those COA levels of detail are not desired, a setting of Do Not Keep will remove that detail.

- Summary: Amounts are summarized at the Closing Classifications group level with no details for transactions or posting codes.
- Summary with Transactions: Amounts from the Summary mode are augmented with transaction listings for the selected accounting period.
- Summary with Posting Codes: Amounts from the Summary mode are augmented with the various posting codes used for the posting code closing classifications defined to the closing classification.
- Full Detail: Amounts from the Summary with Posting Codes mode are augmented with transaction listings for the selected accounting period.

Restartability Information

Submitting the report without a parameter ID or an invalid ID will result in a failure that can be restarted and then edited to add or correct the parameter ID.

Progression Messaging

None

Major Input

- Input Journal – almost always the Accounting Journal (JRNL_ACTG) to provide current period amounts and transaction details.
- Input Ledger – likely the APD Accounting Ledger (LDGR_APD_ACTG) but could be another to provide beginning balance amounts
- Posting Code (R_PSCD) – provides the connection from ledger and journal records to a posting code closing classification so that can be mapped to a closing classification in the report. Also provides names for posting codes in the report modes that display posting codes.
- Parameters for Trial Balance (R_TRI_CUSTOM_PARM) table supplies all the selection and summarization criteria. The parameters are listed in the Batch Parameters topic given below.
- Trial Balance Temporary table (R_TRL_BAL_TMP_TBL) is loaded at the beginning of the report and used as input when classifying records found on the input ledger and input journal into one of the eight closing classification groupings.

Minor Input

- Journal/Ledger Control Detail (R_JRNL_LDGR_CTRL) table defines the list of valid journals and ledgers
- Many different COA reference tables supply names for codes (for example, R_FUND, R_OBJ, R_BSA, R_APPR, and so forth)

Batch Parameters

Parameter	Description	Default Value
Client Name	Optional Client Name for report	None
Date Format	Required. Valid format (1- MM-dd-yyyy, 2 - valid format: MM/dd/yyyy).	1
Parameter ID	Required Parameter ID from the Parameters for Trial Balance table (TBPA)	None
Report ID	Required Report ID for report for optional report distribution	**Increments by 1 with subsequent runs**
Run Date	Required field that specifies whether the report will print the transaction record date or run time	1

	on the report. (1 - Transaction Record Date, 2 - Run Time)	
Select Block Size	Optional. This is the number of records fetched at a time. The value of this parameter should be a positive integer. If not entered, it defaults to 100. Can be used for performance tuning.	100

Custom Parameters (TBPA page)

Parameter	Description	Default Value
General Information		
Parameter ID	A unique alpha-numeric identification assigned to a set of TBPA parameters	None
Fiscal Year	Optional field for specification of a Fiscal Year (FY) for record selection. Must be a valid FY. If left blank, it will default from an As of Date entered. If that date is left blank, the year will be the default FY for the Application Date when the report is run.	(See description)
Accounting Period	Optional field for specification of an Accounting Period (APD) for record selection. Must be a valid APD. Value will be used to select current period activity for Total Debit and Total Credit columns from the input journal. The value minus 1 will be used to select records from the input ledger for Beginning Balance. If left blank, it will default from an As of Date entered. If that date is left blank, the year will be the default APD for the Application Date when the report is run.	(See description)
As of Date	Optional field for specification of an As of Date for record selection. Must be a valid date. Value will be used to select current period activity for Total Debit and Total Credit columns from the input journal. The value can also be used to select records from the input ledger for Beginning Balance, if the ledger contains Record Date. In either case, record selection will be 'on or before' this date. If left blank, it will default to the Application Date when the report is run.	(See description)
Report Mode	Required field to specify the level of detail. Please see earlier section – Descriptions of the Various Report Modes – for more details.	Summary
Summarization Level		
Fund Dimension	Required field for a Fund summarization level. The default is Fund. Do Not Keep and Keep	Fund

	All are shown, but are not a valid choices. When choosing a level, the input ledger must contain that level for a meaningful report.	
Fund Dimension Value	An optional field for the specification of one or more values when a selective trial balance is desired. Values entered here will be used in conjunction with the Fund Dimension choice when selecting records. Multiple values must be separated by commas. Leaving the field blank, the default will result in a report for all Funds or other level choice made in the Fund Dimension field. Specification of a Sub Fund is possible. However, every Fund with that Sub Fund value will be selected.	None
Organizational Dimension	Required field for an Organization summarization level. The default is Do Not Keep. Other valid choices are Government Branch, Cabinet, Department, and Unit. Any other choice is not valid. When choosing a level, the input ledger must contain that level for a meaningful report.	Do Not Keep
Object Dimension	Required field for an Object summarization level. The default is Do Not Keep. Keep All is shown, but is not a valid choice. When choosing a level, the input ledger must contain that level for a meaningful report.	Do Not Keep
Revenue Source Dimension	Required field for a Revenue Source summarization level. The default is Do Not Keep. Keep All is shown, but is not a valid choice. When choosing a level, the input ledger must contain that level for a meaningful report.	Do Not Keep
BSA Dimension	Required field for a Balance Sheet Account summarization level. The default is Do Not Keep. Keep All is shown, but is not a valid choice. When choosing a level, the input ledger must contain that level for a meaningful report.	Do Not Keep
Appropriation Dimension	Required field for an Appropriation summarization level. The default is *Do Not Keep*. *Keep All* is shown, but is not a valid choice. When choosing a level, the input ledger must contain that level for a meaningful report.	Do Not Keep
Closing Classification		
Asset Closing Classification Values	A required field to define the closing classification values for assets. Default values are the delivered	1,4,5

	closing classifications for Assets (1), Contra Assets (4), and Cash (5). Any additional classifications created should be added in this list.	
Asset Closing Classification Name	A required field for the group name for the closing codes chosen in the Asset Closing Classification Values field.	Assets
Liability Closing Classification Values	A required field to define the closing classification values for liabilities. The default value is the delivered closing classification for Liabilities (2). Any additional classifications created should be added in this list.	2
Liability Closing Classification Name	A required field for the group name for the closing codes chosen in the Liability Closing Classification Values field.	Liabilities
Equity Closing Classification Values	A required field to define the closing classification values for equity. Default values are the delivered closing classifications for Equity (3) and Equity Offsets Closed to Net Assets (7). Any additional classifications created should be added in this list.	3,7
Equity Closing Classification Name	A required field for the group name for the closing codes chosen in the Equity Closing Classification Values field.	Equity
Pre Encumbrance Closing Classification Values	A required field to define the closing classification values for pre encumbrances. The default value is the delivered closing classification for Pre Encumbrances (13). Any additional classifications created should be added in this list.	13
Pre Encumbrance Closing Classification Name	A required field for the group name for the closing codes chosen in the Pre Encumbrance Closing Classification Values field.	Pre Encumbrances
Encumbrance Closing Classification Values	A required field to define the closing classification values for encumbrances. The default value is the delivered closing classification for Encumbrances (12). Any additional classifications created should be added in this list.	12
Encumbrance Closing Classification	A required field for the group name for the closing codes chosen in the Encumbrance Closing Classification	Encumbrances

Name	Values field.	
Expenditure Closing Classification Values	A required field to define the closing classification values for expenditures and expenses. Default values are the delivered closing classifications for Accrued Expenditures (11) and Cash Expenditures (10). Any additional classifications created should be added in this list.	10,11
Expenditure Closing Classification Name	A required field for the group name for the closing codes chosen in the Expenditure Closing Classification Values field.	Expenditure/Expenses
Revenue Closing Classification Values	A required field to define the closing classification values for revenues. Default values are the delivered closing classifications for Billed Revenue (15) and Collected Revenue (14). Any additional classifications created should be added in this list.	14,15
Revenue Closing Classification Name	A required field for the group name for the closing codes chosen in the Revenue Closing Classification Values field.	Revenues
Other Account Closing Classification Values	A required field to define the closing classification values for all other types of accounts not accounted for with the other Closing Classification parameters. The default value is the delivered closing classification for Accounts Left in Old Year (6). Any additional classifications created should be added in this list. To leave accounts off of a trial balance, those posting codes should belong to a Closing Classification not listed in any Closing Classification field. Therefore, if accounts with the Accounts Left in Old Year closing classification should not appear on a trial balance, then create a 'dummy' closing classification and put it in this parameter. To leave all such 'other' activity off the report, do not put a value of 6 in this field but instead enter a posting code closing classification value that has no associated posting codes.	6
Other Account Closing Classification Name	A required field for the group name for the closing codes chosen in the Encumbrance Closing Classification Values field.	Other Accounts

Report Source		
Journal ID	Required input journal to be used for current period record selection. The Accounting Journal is the default and the most likely input source.	1
Journal Name	Protected field that is the name inferred for the Journal ID from the Journal/Ledger Control table.	None
Ledger ID	Required input ledger to be used for current period record selection. The Accounting APD Ledger is the default and the most likely input source. However, please see the earlier section – Important Run Instructions – for more information on input ledgers.	11
Ledger Name	Protected field that is the name inferred for the Ledger ID from the Journal/Ledger Control table.	None

Major Output

- Trial Balance Temporary table (R_TRL_BAL_TMP_TBL) is loaded at the beginning of the report and used as input when classifying records found on the input ledger and input journal into one of the eight closing classification groupings.
- Trial Balance Report

Job Return Codes

The following table shows the potential job return codes for the CR XML Creation job.

Return Code	Condition
Successful (1)	All parameter validations were successful and report was produced with records.
Warning (4)	Job does not end with this status
Non-Fatal Error (8)	Job does not end with this status
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Report layout file could not be found • 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.

Selection Criteria & Sort Sequencing

Select records from input journal where:

Fund Dimension Value = corresponding journal field (if Fund Dimension Value is not blank)

Fiscal Year (FY_DC) = Fiscal Year parameter and

Accounting Period (PER_DC) <= Accounting Period parameter and

Transaction Record Date (DOC_REC_DT) <= As of Date parameter and

Posting Code in list created on the Trial Balance Temporary table

Page Breaking

Fund Dimension

Organizational Dimension (if Organizational dimension <> *Do Not Keep*)

Account Type* (based on Closing Classification of Posting Code)

Posting Code (if Report mode = *Summary with Posting Code* or *Full Detail*)

Balance Sheet Account Dimension (if Balance Sheet Account Dimension <> *Do Not Keep*)

Object Dimension (if Object Dimension <> *Do Not Keep*)

Revenue Source Dimension (if Revenue Source Dimension <> *Do Not Keep*)

Appropriation Dimension (if Appropriation Dimension <> *Do Not Keep*)

Posting Code (added after BSA, Object, Revenue Source grouping if Report mode = *Summary* or *Summary with Transactions*)

Debit/Credit Indicator

Transaction Code, Transaction Department, Transaction ID, Vendor/Customer, Transaction Record Date (if Report Mode = *Summary with Transactions* or *Full Detail*)

Accounting Period

Order By (ascending unless specified):

Fund Dimension

Organizational Dimension (if Organizational Dimension <> *Do Not Keep*)

Account Type* (based on Closing Classification of Posting code)

Posting Code (if Report Mode = *Summary with Posting Code* or *Full Detail*)

Balance Sheet Account Dimension (if Balance Sheet Account Dimension <> *Do Not Keep*)

Object Dimension (if Object Dimension <> *Do Not Keep*)

Revenue Source Dimension (if Revenue Source Dimension <> *Do Not Keep*)

Appropriation Dimension (if Appropriation Dimension <> *Do Not Keep*)

Posting Code (added after BSA, Object, Revenue Source grouping if Report Mode = *Summary* or *Summary with Transactions*)

Debit/Credit Indicator

Transaction Code, Transaction Department, Transaction ID, Vendor/Customer, Transaction Record Date (if Report mode = *Summary with Transactions* or *Full Detail*)

Accounting Period descending

* *Account Type corresponds to the eight major groupings of closing classification found on the TBPA page.*

Note: When a COA dimension is involved, appropriate fields are used on the selection query:

Object Dimension set to Object then OBJ_CD code is used

Object Dimension set to Sub Object then OBJ_CD and SOBJ_CD are used

Object Dimension set to Object Class then OCLS_CD is used

Object Dimension set to *Do Not Keep* then the Object element and rollups are not used on the selection query and no level of Object detail will be presented in the report

Calculations:

- Opening Balance is calculated as follows:

  Get sum (AM) from input ledger specified on TBPA parameter record where

  - Fiscal Year (FY_DC) = Fiscal Year parameter and
  - Accounting Period (PER_DC) < Accounting Period parameter and
  - (Transaction Record Date (DOC_REC_DT) is null OR <= As of Date parameter) and
  - Posting Code = Posting code on selected record (if Report mode = Summary with Posting Code or Full Detail)
  - Posting Code is in Account Type being processed (if Report mode = Summary or Summary with Transactions)
  - Fund, Organization, BSA, Object, Appropriation, and Revenue Source Dimension fields match with the corresponding field values on the selected record(s)
  - Note: Individual lines for transaction listings do not show a Beginning Balance but the total for the account does.

- Total Debits and Total Credits amount fields are calculated as follows:

  If report row is a transaction, sum of all debits or credits from that transaction where

  - Fiscal Year (FY_DC) = Fiscal Year parameter and
  - Accounting Period (PER_DC) = Accounting Period parameter and
  - (Transaction Record Date (DOC_REC_DT) is <= As of Date parameter) and
  - Posting Code = Posting code on selected record (if Report mode = Summary with Posting Code or Full Detail)
  - Posting Code is in Account Type being processed (if Report mode = Summary or Summary with Transactions)
  - Fund, Organization, BSA, Object, Appropriation, and Revenue Source Dimension fields match with the corresponding field values on the selected record(s)
  - For each unique vendor/customer code on transaction

  If report row is a single account listing, sum of all debits or credits where

  - Fiscal Year (FY_DC) = Fiscal Year parameter and
  - Accounting Period (PER_DC) = Accounting Period parameter and
  - (Transaction Record Date (DOC_REC_DT) is <= As of Date parameter) and
  - Posting Code = Posting code on selected record (if Report mode = Summary with Posting Code or Full Detail)

- Posting Code is in Account Type being processed (if Report mode = Summary or Summary with Transactions)
- Fund, Organization, BSA, Object, Appropriation, and Revenue Source Dimension fields match with the corresponding field values on the selected record(s)

- Ending Balance is calculated as follows:
  - Sum Beginning Balance + Total Debits + Total Credits

  Note: Individual lines for transaction listings do not show an Ending Balance but the total for the account does.

Page Breaking:

When the header COA dimensions change: Fund and Organizational

Sample Reports

The following are a series of reports starting with a very high level of summarization and then stepping down to a large amount of detail. The same account is being displayed in ever-increasing levels of detail. Not every combination of detail is presented below, as the samples are just to give an idea of the many different permutations of the report.

The samples start with a *Summary* Run Mode report with just Fund information for COA detail

Local

REPORTID: 21 PAGE : 1
RUN DATE: 01-10-2024 TRIAL BALANCE REPORT - SUMMARY FY 2018 THROUGH APD 12
RUN TIME: 13:12:29 AS OF 12-31-2018
FUND : 010 Payroll 010

APPR UNIT: ZZ02 ZZ02
DEPARTMENT : 010 Personnel
UNIT : 1001 FA test 3

ACCOUNT TYPE OBJ/REV/BSA	BEGINNING BALANCE	TOTAL DEBITS	TOTAL CREDITS	ENDING BALANCE
Asset				
ACCT TYP ASSET TOTAL	$0.00	$0.00	($100.00)	($100.00)

Now the above will be broken down by BSA and Sub BSA, but still in *Summary* mode. When no Sub BSA is used with a BSA, the report shows the NULL value and no name on the sub account row.

FUND : 010 Food Services Fund

ACCOUNT TYPE OBJ/REV/BSA	BEGINNING BALANCE	TOTAL DEBITS	TOTAL CREDITS	ENDING BALANCE
Asset				
0101 - CASH				
0001 - MAIN CASH	$0.00	$10.00	($121.00)	($111.00)
0002 - Alternate Cash	$0.00	$10.00	($10.00)	$0.00
NULL	$0.00	$10.00	($10.00)	$0.00
0101 TOTAL	$0.00	$30.00	($141.00)	($111.00)
ACCT TYP ASSET TOTAL	$0.00	$30.00	($141.00)	($111.00)

Now transaction activity is added by changing the Report Mode to *Summary with Transactions*.

ACCOUNT TYPE OBJ/REV/BSA DOCUMENT	RECORD DATE	VENDOR/CUSTOMER	BEGINNING BALANCE	TOTAL DEBITS	TOTAL CREDITS	ENDING BALANCE
Asset						
0101 - CASH						
0001 - MAIN CASH						
IET,010,07100700000000000004	2007-07-10				($10.00)	
MD,010,DSJFDS	2007-07-09	AMS-01			($111.00)	
IET,010,07100700000000000004	2007-07-10			$10.00		
0001 TOTAL			$0.00	$10.00	($121.00)	($111.00)
0002 - Alternate Cash						
IET,010,07100700000000000004	2007-07-10				($10.00)	
IET,010,07100700000000000004	2007-07-10			$10.00		
0002 TOTAL			$0.00	$10.00	($10.00)	$0.00
NULL						
IET,010,07100700000000000004	2007-07-10				($10.00)	
IET,010,07100700000000000004	2007-07-10			$10.00		
NULL TOTAL			$0.00	$10.00	($10.00)	$0.00
0101 TOTAL			$0.00	$30.00	($141.00)	($111.00)
ACCT TYP ASSET TOTAL			$0.00	$30.00	($141.00)	($111.00)

Next, instead of transaction activity, the Asset classification is broken down into Posting Code sections while retaining the BSA and Sub BSA information. Here the Report Mode is *Summary with Posting Codes.*

ACCOUNT TYPE OBJ/REV/BSA	BEGINNING BALANCE	TOTAL DEBITS	TOTAL CREDITS	ENDING BALANCE
Asset				
A001 - Cash				
0101 - CASH				
0001 - MAIN CASH	$0.00	$10.00	($121.00)	($111.00)
0002 - Alternate Cash	$0.00	$10.00	($10.00)	$0.00
NULL	$0.00	$10.00	($10.00)	$0.00
0101 TOTAL	$0.00	$30.00	($141.00)	($111.00)
POSTING CODE A001 TOTAL	$0.00	$30.00	($141.00)	($111.00)
ACCT TYP ASSET TOTAL	$0.00	$30.00	($141.00)	($111.00)

Finally, the *Full Detail* mode with BSA and Sub BSA details for COA presents the most granular level of the report.

ACCOUNT TYPE OBJ/REV/BSA DOCUMENT	RECORD DATE	VENDOR/CUSTOMER	BEGINNING BALANCE	TOTAL DEBITS	TOTAL CREDITS	ENDING BALANCE
Asset						
A001 - Cash						
0101 - CASH						
0001 - MAIN CASH						
IET,010,07100700000000000004	2007-07-10				($10.00)	
MD,010,DSJFDS	2007-07-09	AMS-01			($111.00)	
IET,010,07100700000000000004	2007-07-10			$10.00		
0001 TOTAL			$0.00	$10.00	($121.00)	($111.00)
0002 - Alternate Cash						
IET,010,07100700000000000004	2007-07-10				($10.00)	
IET,010,07100700000000000004	2007-07-10			$10.00		
0002 TOTAL			$0.00	$10.00	($10.00)	$0.00
NULL						
IET,010,07100700000000000004	2007-07-10				($10.00)	
IET,010,07100700000000000004	2007-07-10			$10.00		
NULL TOTAL			$0.00	$10.00	($10.00)	$0.00
0101 TOTAL			$0.00	$30.00	($141.00)	($111.00)
POSTING CODE A001 TOTAL			$0.00	$30.00	($141.00)	($111.00)
ACCT TYP ASSET TOTAL			$0.00	$30.00	($141.00)	($111.00)

Problem Resolution

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

Parameter Validation

Possible Return Codes	Condition	Recommendation	Other instructions
Successful (1)	Successful	N/A	N/A
Failed (12)	Required Parameters are not entered **Sample Message:** Parameter ID required	Run a new instance of the engine with correct parameters or restart the current one and make parameter changes.	

Report Production

Possible Return Codes	Condition	Recommendation	Other instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	Failed because of runtime exceptions for unexpected situation	Failure reason needs to be investigated before scheduling a new report.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	
System Failure (20)	When the job is terminated because of database server or network issues	Reason for the System Failure needs to be investigated before scheduling a new job.	

2.2.15 Trial Balance with Prior Period Balances

Job Name	Trial Balance with Prior Period Balances
Recommended Frequency	On demand
Single Instance Required	Yes
Can be Restarted?	In certain circumstances
Reports Generated	Trial Balance Report

Overview

The Trial Balance with Prior Period Balances is an extension of the Trial Balance report. This report follows the same process as the Trial Balance report. While there are minor differences in terms of labels and calculated amounts, the primary difference is that this report will look back into the prior fiscal year when it has not been closed to calculate beginning balances.

The report lists all the account balances at many different levels of detail, as determined by parameter settings, from an input ledger and journal. To control that level of detail and provide selection criteria, there is an online page: Parameters for Trial Balance (TBPA). Both reports share this page to define multiple parameter IDs to produce different types of reports for different time periods. As these reports are very similar with regards to using the TBPA page, please look to the Trail balance report for field-level details. However, this report only supports Summary mode and does not support other modes through the TBPA page. Furthermore, only Fund and Sub Fund choices of the Fund Dimension and BSA from the BSA Dimension are supported. The Object, Revenue, and Organizational Dimension fields are not read.

Please note that since the report will read a journal and a ledger, the Journal Engine and Ledger Engine should not be run while the report is running.

Restartability Information

Yes

Progression Messaging

None

Major Input

- Input Journal – almost always the Accounting Journal (JRNL_ACTG) to provide current period amounts and transaction details.
- Input Ledger – likely the APD Accounting Ledger (LDGR_APD_ACTG) but could be another to provide beginning balance amounts.
- Posting Code (R_PSCD) – provides the connection from ledger and journal records to a posting code closing classification so that can be mapped to a closing classification in the report. Also provides names for posting codes in the report modes that display posting codes.

- Parameters for Trial Balance (R_TRI_CUSTOM_PARM) table supplies all the selection and summarization criteria. Of note, the summarization fields for Organization, Object, and Revenue must be set to *Do not keep* for this report to run. Only the Fund and Balance Sheet summarization fields can be used.
- Trial Balance Temporary table (R_TRL_BAL_TMP_TBL) is loaded at the beginning of the report and used as input when classifying records found on the input ledger and input journal into one of the eight closing classification groupings.

Minor Input

- Journal/Ledger Control Detail (R_JRNL_LDGR_CTRL) table defines the list of valid journals and ledgers.
- Many different COA reference tables supply names for codes (for example, R_FUND, R_OBJ, R_BSA, and so on).
- Special Fund Accounts (SPECFUND) and Special Accounts (SPEC) supply accounts for the prior year.
- Fiscal Year (R_FY) supplies whether or not the prior FY has been closed by the Annual Close process or not.

Batch Parameters

Parameter	Description	Default Value
Client Name	Optional Client Name for report	None
Parameter ID	Required Parameter ID from the Parameters for Trial Balance (TBPA) table	None
Report ID	Required Report ID for report for optional report distribution	None

Major Output

- Trial Balance Temporary table (R_TRL_BAL_TMP_TBL) is loaded at the beginning of the report and used as input, when classifying records found on the input ledger and input journal into one of the eight closing classification groupings.
- Trial Balance Report

Job Return Codes

The following table shows the potential job return codes for the Trial Balance with prior period amounts job.

Return Code	Condition
Successful (1)	All parameter validations were successful and report was produced with records.
Warning (4)	The job does not end with this status.
Non-Fatal Error	The job does not end with this status.

(8)	
Failed (12)	The job fails under the following conditions: • Parameters are invalid. • Report layout file could not be found. • 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.

Selection Criteria & Sort Sequencing

Select records from input journal where:

Fund Dimension Value = corresponding journal field (if Fund Dimension Value is not blank)

Fiscal Year (FY_DC) = Fiscal Year parameter and

Accounting Period (PER_DC) <= Accounting Period parameter and

Run Time Date (RUN_TMDT) <= As of Date parameter and

Posting Code in list created on the Trial Balance Temporary table

Note: When previous Fiscal Year is not closed the selection criteria will additionally include the following:

Fiscal Year (FY_DC) = Fiscal Year parameter -1

Group By:

Fund Dimension

Account Type* (based on Closing Classification of Posting Code)

Balance Sheet Account Dimension (if Balance Sheet Account Dimension <> Do Not Keep)

Posting Code

Debit/Credit Indicator

Fiscal Year

Accounting Period

Order By (ascending unless specified):

Fund Dimension

Account Type* (based on Closing Classification of Posting code)

Balance Sheet Account Dimension (if Balance Sheet Account Dimension <> Do Not Keep)

Posting Code

Debit/Credit Indicator

Fiscal Year

Accounting Period descending

* *Account Type corresponds to the eight major groupings of closing classification found on the TBPA page.*

Calculations:

- Opening Balance is calculated as follows:

  Get sum (AM) from input ledger specified on TBPA parameter record where

  - Fiscal Year (FY_DC) = Fiscal Year parameter and
  - Accounting Period (PER_DC) < Accounting Period parameter and
  - (Run Time Date (RUN_TMDT) is null OR <= As of Date parameter) and
  - Posting Code is in Account Type being processed
  - Fund, and BSA, Dimension fields match with the corresponding field values on the selected record(s)

  For previous FY records:

  - 07 Equity Offset records are deducted from 03 Equity Roll Forward
  - 08 Other Accounts are excluded
  - Determine Encumbrance and Pre-Encumbrance for each fund and reduce them from Fund balance and include them under a separate section
- Total Debits and Total Credits amount fields are calculated as follows:

  If report row is a transaction, sum of all debits or credits from that transaction where

  - Fiscal Year (FY_DC) = Fiscal Year parameter and
  - Accounting Period (PER_DC) = Accounting Period parameter and
  - (Run Time Date (RUN_TMDT) is <= As of Date parameter) and
  - Posting Code = Posting code on selected record
  - Posting Code is in Account Type being processed
  - Fund, and BSA Dimension fields match with the corresponding field values on the selected record(s)
  - For each unique vendor/customer code on transaction

  If report row is a single account listing, sum of all debits or credits where

  - Fiscal Year (FY_DC) = Fiscal Year parameter and
  - Accounting Period (PER_DC) = Accounting Period parameter and
  - (Run Time Date (RUN_TMDT) is <= As of Date parameter) and
  - Posting Code = Posting code on selected record
  - Posting Code is in Account Type being processed
  - Fund, and BSA Dimension fields match with the corresponding field values on the selected record(s)
- Ending Balance is calculated as follows:
  - Sum Beginning Balance + Total Debits + Total Credits
  - Change in Fund Balance is calculated as follows:

    Revenues – Expenses
- Adjusted Total Fund Balance is calculated as follows:

- Fund Balance + Revenues – Expenses (where Fund Balance does not include the equity accounts for Encumbrance and Pre Encumbrance. The account values for Encumbrance and Pre Encumbrance will come from Miscellaneous section of Special Fund Accounts (SPECFUND) or Special Accounts (SPEC)).

- Total Liabilities & Fund Balance is calculated as follows:
  - Adjusted Total Fund Balance + Liabilities

  Note: Individual lines for transaction listings do not show an Ending Balance but the total for the account does.

Page Breaking:

When the header COA dimensions change: Fund

Problem Resolution

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

Parameter Validation

Possible Return Codes	Condition	Recommendation	Other instructions
Successful (1)	Successful	N/A	N/A
Failed (12)	Required Parameters are not entered **Sample Message:** Parameter ID required	Run a new instance of the job with correct parameters or restart the current one and make parameter changes.	

Report Production

Possible Return Codes	Condition	Recommendation	Other instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non-Fatal Error (8)	N/A	N/A	N/A
Failed (12)	The job failed because of runtime exceptions for	Failure reason needs to be investigated before	

	unexpected situation	scheduling a new report.	
Terminated (16)	The 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.	