CGI Advantage® 4

System Assurance Run Sheets Guide

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

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

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

Copyright © 2001, 2023, CGI Technologies and Solutions Inc. All Rights Reserved.

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:

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

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

1.1 Common terms and glossary used

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

2 Description of Processes

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

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

System Wide Batch Parameters:

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

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

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

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

Note:

Assumptions while implementing system wide batch parameters: It is assumed that wherever in the Job processes system wide batch parameter variables (that is, AMSEXPORT,

AMSIMPORT, AMSROOT, AMSLOGS, AMSPARM, AMSTEMP, AMSSPOOL) are declared as input parameters, care should be taken to set the overrideable flag for that variable to *true*, otherwise the process may fail.

Pivot Date/Year Validation:

Note:

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

2.1 System Assurance Batch and Chain Processes

CGI Advantage Financial has several systems assurance processes available to verify the integrity of the data posted to the system. Specifically, these processes verify that:

- The ledgers, journals, and positing line catalog are in sync with the budget tables and with certain non-budget tables
- Debits and credits are equal within a balanced journal/balanced ledger
- All records on the Disbursement Request (DISRQ) table are correct (which ensures that the amounts on transactions and tables that control outgoing payments are correct)
- All transaction component levels are in sync, and all posting lines within a transaction have the same journal posting indicator
- Records have been summarized from the journals into the ledger correctly
- User-specified ledger records sum up to the journal records of which they are comprised
- The Customer Account Information (CUSTA) table has been updated by all transactions with a billing profile correctly

All of these processes produce exception reports that list any data discrepancies found by the process.

Keep in mind that the implementation of systems assurance procedures is site-specific, so your organization may have additional systems assurance processes and procedures in place.

The following table lists the processes and the ID by which you run each process:

Process ID	Process Name
Systems Assurance 01	Budget Actuals Versus Ledgers, Journals and Posting Line Catalog
Systems Assurance 02	Non-Budgetary Actuals Versus Ledgers, Journals, and Posting Line Catalog
Systems Assurance 03	Debits Equal Credits
Systems Assurance 04	The Accounting Lines Requesting Payment vs. the Disbursement Request (DISRQ) Table
Systems Assurance 05	Disbursement Request (DISRQ) Table Versus the Accounting Lines in Payment Request (PR), Accounting-Based Spending (ABS), and Travel (TRVL) Transactions
Systems Assurance 06	Transaction Component Levels In-sync
Systems Assurance 07	Journal Ledger Cross Reference table In-sync
Systems Assurance 08	Ledger to Journal Diagnostic Process
Systems Assurance 09	Customer Account Information Versus Journal and Posting Line Catalog
System Assurance 10	Budget Linking Assurance
Systems Assurance 11	Budget Systems Assurance
Systems Assurance 12	Budgeting Incremental Updates Systems Assurance

Process ID	Process Name
System Assurance 13	Out of Sync differences between Financial and VSS
System Assurance 15	Open Amounts
System Assurance 16	Missing Posting Lines

Descriptions of these processes are organized in this section by their Process ID.

The System Assurance 01 (SA01) and Systems Assurance 02 (SA02) processes have been improvised to offer reduced runtimes. As a result they deserve a special mention. Discussed below are items that serve as a common background for the SA01 and SA02 processes.

Terminology: The terminology described below will be referred to frequently when discussing the enhanced SA01 and SA02 processes.

Values stored in the Status field of the Journal Log (JLOG) record

- FA – Failed
- U – Unprocessed
- FI – Final
- I – Intended

Definitions:

- **Critical parameters** are those parameters that when changed result in the Systems Assurance Process behaving differently. Typically for the Systems Assurance 01 and 02 processes the critical parameters are the BFY and Full or Incremental Run parameter values.
- **Major inputs** are the input tables that serve the materials for creation of the output record on major outputs
- **Major outputs** are the output from the process that serves as an interface between upstream and downstream modules/processes. In other words major output is the output that serves as the major input for the down stream steps.

Incremental Run

An Incremental Run is defined as a run where journal repost is resumed from the point (the record number) where the previously run, Full Run or an Incremental Run, ended. Records selected are from this point to the end of the journal. BFY is immaterial for this run.

Full Run

A Full Run is a run where eligible ledger and journal records are selected to be posted to the shadow tables.

At this point it is worthwhile to introduce the concept of a Full Run with BFY. This concept only applies to the SA01 process and not to the SA02 process since SA02 does not rely on the BFY parameter,

Full Run with BFY

A full run with BFY can be characterized as having two parts employing parts from the full and incremental runs:

1. A Full Run for the records having BFY equal to the BFY parameter
2. An incremental run for the set of eligible records having BFY **NOT EQUAL** to the BFY parameter.

POINT TO UNDERSTAND

It is imperative to understand that the way repost of ledger, journal and posting line records occurs to the shadow table has not been affected. The record selection and designation criteria have been leveraged to exploit the inherent data set independence that exists in the processed ledger, journal or posting line records.

Key Assumption

- Online transaction will not be conducted simultaneously when the System Assurance 1 or System Assurance 2 jobs are running.

Key Journal Log Process IDs to note:

JLOG (JRNL_LOG) processor IDs

The following are the JLOG processor Ids used for SA01 (substitute preceding SA01 with SA02 when inserting for SA02 jobs):

1. **SA01JL** – Full Run record range for unledgerized journal records. For a Full Run with BFY this will be unledgerized journal records having that BFY. For a Full Run without BFY it will be unledgerized journal records irrespective of BFY values.
2. **SA01IJL** – Record range for ***unprocessed*** journal records. This can be created by either an Incremental Run, or a Full Run with BFY. For a FULL Run with BFY, this range will consist of unprocessed journal records not belonging to the specified BFY set. For an 'Incremental Run', this range will consist of unprocessed journal records belonging to the specified BFY set. The distinction between whether the preprocessor is a Full Run or an Incremental Run can be made via the "Full or Incremental Run" parameter.
3. **SA01LL**- Full Run record range for workload sequence numbers. For a Full Run with BFY only ledger records for that BFY will be loaded to the workload. For a Full Run without BFY all ledger records will be loaded to the workload table.

The above can be extended for System Assurance 02 in a like manner by replacing the '01' with '02' and noting that there is no concern of BFY parameter value for SA02.

A note about Journal Log ranges

- Individual ranges are calculated using the minimum, maximum and the commit block size. Minimum and Maximum refer to Minimum and Maximum journal record numbers for SA01JL/SA01IJL entries, or the workload sequence number for SA01LL entries.

IMPORTANT CONSIDERATIONS AND SCENARIOS TO AVOID WHEN RUNNING SA01 or SA02:

1. The very first SA processing should be triggered off using a Full Run. Subsequent runs can be incremental. The very first run of the SA process cannot be an incremental run. The preprocessor is intelligent to throw an error if there are no SA01JL OR SA01IJL entries which are indicative of prior full run.
2. Pre-Processor (PP) is singleton which means only one instance can run. However separate PP entries for SA01 and SA02 were created since the PP for SA01 and SA02 are independent in their work.
3. User must exercise care not to run the online chain via batch catalog when the JIC script is running else data corruption of the shadow tables will occur. The ability to run the chain via the batch catalog is simply to preserve the usability of the SA process as it exists today.
4. While the PP is always singleton, the parallel repost step is not. Hence extreme caution must be exercised not to trigger off two independent JIC scripts in parallel at the same time. Violation of this rule will cause data corruption of the shadow tables.
5. The parallel repost step provides the capability to restart a failed job from the batch catalog. The parallel repost jobs not only complete failed work they also go about working on existing unprocessed records. When a restarted job is running, it should be the only step that runs along with other restarted jobs. In other words the chain job should not be run and nor should the independent standalone jobs associated with the SA01 and SA02 be run when restarted jobs are running. Violation of this rule will cause data corruption of the shadow tables.
6. Repost unless they are restarted jobs should always be run with an immediately preceding pre-processing step. In other words the pre-processing should always precede the repost steps except for restarted repost jobs.

2.1.1 Systems Assurance 01 – Budget Actuals Versus Ledgers, Journals and Posting Line Catalog

Chain Name	System Assurance 01
Recommended Frequency	Daily or On Demand. **Note:** With the exception of the report, System Assurance 01 jobs cannot run concurrently with any of the following activities. 1) Transaction processing 2) Journal posting 3) Ledgerization
Single Instance Required	Yes
Can be Restarted?	N/A
Reports Generated	Yes, SA01 out of sync report will be generated if the job is run in Run Mode Full or Incremental and finds any out of sync records.

Overview

The System Assurance 01 process verifies that records in a specified ledger, records in that ledger's source journal that are not yet ledgerized, and all posting lines not yet journalized to that journal are in sync with actual amounts on budget tables. An actual amount refers to the stand-alone budget buckets (such as encumbrances, cash expenditures, and collected earned revenue) that are updated by accounting transactions. When what is recorded in the three sources does not match a budget bucket, a line is written to the System Assurance 01 Report with the supporting information based on input parameters. Note that pending buckets and allotment lines are not verified as part of this job.

The System Assurance 01 process can be run in Full, Incremental, or Pre-Archive mode. When running in Full mode, the job will process all records, new and old. In Incremental mode, the job will process only records posted since the last time the SA01 job was run. In Pre-Archive mode, the job will process records for a single FY in preparation for running the Journal/Ledger Archiving chain job.

The following jobs are an integral part of the System Assurance 01 Process:

- SA01 Pre-Processor
- System Assurance 1
- SA01 Report

The SA01 Pre-Processor and SA01 Report jobs are singleton jobs, meaning that only one of their instances can be run. The System Assurance 1 job (repost step) has been designed such that multiple instances may run without re-doing each others work. If a site wants to exploit the multiple repost processing, they are advised to use a Job Interaction Client (JIC) script or manually schedule multiple repost steps to get triggered at the same time(possible via online access). Both of these options require running these component jobs outside of the System Assurance 01 Chain. To facilitate this, separate catalog entries have been provided for each

component System Assurance 01 job so that the Job Interaction Client script can be used to trigger multiple System Assurance 1 jobs (the repost step).

The first output from the process is an SA1 temporary table (SA_BUD) which is the input for the SA01 report job. This table is sometimes referred to as the SA01 shadow table.

The second output is a report that shows in order of Budget Structure ID, Budget level ID, and Budget Name; all those buckets that do not have the same balance for a budget line, as they should compared to the input ledger, journal, and posting line catalog. When the Report All parameter is set to “Y”, the report will show all of the budget lines. Each page header lists the details about what budget structure, level and line has a bucket that is out of sync. Following that section is a row for the budget bucket that is out of sync. The amounts for the bucket found in the ledger, journal, and posting line catalog are each listed separately in columns followed by the amount in the bucket itself. The last column is a calculated difference, which is arrived at by adding the amounts found in the ledger, journal and posting line catalog and then subtracting that amount from the budget bucket. The difference is shown as an absolute value.

Several of the SA01 Report job’s input parameters add information to the report. If any is set to Y, there will be additional Diagnostic Detail included in the report as explained below:

- The Show Budget Activity parameter will add a section that lists all records stored in the audit/activity budget level that records all update activity by non-budgetary transactions. Any transactions at this level with a net effect of zero will not appear. For example, a fully referenced transaction or a transaction that was accepted and then cancelled.
- The Show Unledgerized Journal Records parameter will add a section listing all journal records that updated the budget bucket but have not yet been ledgerized.
- The Show Unjournalized Journal Records parameter will add a section listing all posting line records that updated the budget bucket but have not yet been journalized.
- The Show Ledger Records parameter will add a section that lists all ledger record ID’s and the journal records that comprise them that updated the budget bucket.

Systems Assurance 1 will also assure those budget amounts updated by posting codes that are set to update a budget bucket, but also require a balance sheet account. The delivered posting codes with this setup are R101 - Billed Unearned Revenue, R102 - Collected Unearned Revenue, and R105 - Unearned Revenue for Pre Payment. If such posting codes are used, the Real Account Level option on Journal / Ledger Control (JLCTRL) should be set to “All Levels” for the ledger that is used as input into Systems Assurance 1 (LDGR_SA_BUD is the delivered input ledger). That way, these “real account” posting codes will retain the necessary COA level of detail needed in order for the activity to be assured against the various budget levels.

Please note: If the Real Account Level option is ever changed for the input ledger, then that ledger must be rebuilt.

The acceptable job return codes (configured in the Configure Chain Job section of the Job Setup in CGI Advantage) for the jobs in the System Assurance 01 Chain are delivered to be set to Successful. As with all CGI Advantage chain jobs, these acceptable return codes are configurable and may be changed to meet certain requirements. For simplicity, this run sheet assumes the chain is configured as delivered, including parameter settings.

In Pre-Archive mode, the job will empty all records from the Budget vs. Actual System Assurance Temporary Table ("shadow table") except for those with Posting Source “Archived” (4). The System Assurance 1 step will add/make updates to these “Archived” records, and the SA01 Report step will be made inactive when running as a chain job. The Journal/Ledger Archiving chain then deletes the ledger records for the Archive FY at the end of the process. A Full run of the Systems Assurance 1 process is expected at any time after archiving is complete. Please plan to allow time for a full SA1 run after the archive for all budget years remaining on the ledger.

Major Input

- LDGR_SA_BUD – 'Ledger SA Budgets' (or other appropriate ledger specified as a parameter). The record for the ledger must have System Assurance 01 checked in the Journal/Ledger Control (JLCTRL) table. The record must also contain all the COA elements and rollups used to define active budgets in the application.
- JRNL_ACTG – 'Accounting Journal' (or other journal which is the source journal to the input ledger)
- PSTNG_LN_CAT - 'Posting Line Catalog' – Records selected are those with a transaction phase of Final (3) or Historical (5) whose journal posting status is not equal to Posted (3).
- Batch parameters - These parameters are discussed in detail later.
- All active budget structure tables (BUD_STRU_##_LVL_##)
- Posting Code (R_PSCD)
- JRNL_LOG – Journal Log - In the Journal Log, the job looks for the records with the following process IDs:
  - SA01LL (identifies the ledger / work load records) a range specified that starts and ends with the SEQ_NO values on the corresponding SA*_WKLD table).
  - SA01JL (identifies the journal records that have not been ledgerized yet).
  - SA01IJL (identifies the unprocessed records)

Major Output

- JRNL_LOG – Journal Log – Records that need to be processed by System Assurance 01 will have their Status set to 'U' –Unprocessed – on the Journal Log.
- SA1_WRKLD - SA1 Workload Table
- SA_BUD - Budget vs. Actual System Assurance Temporary Table ("shadow table")
- System Assurance 1 Report

Chain Job Return Code

The following table shows the potential return codes for System Assurance 01 Chain. Note that the Chain job will end with the highest return code across all of the jobs.

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

Problem Resolution

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

Systems Assurance 01 Chain: SA01 Pre-Processor Job

Job Name	SA 01 Pre-processor
Recommended Frequency	Daily or On Demand. This job can be run as part of the System Assurance 01 Chain or it can be run independently through a Job Interaction Client (JIC) script or manually scheduled. **Note:** This job cannot run concurrently with any of the following activities. 1) Transaction processing 2) Journal posting 3) Ledgerization
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	No

Overview

As the name suggests the preprocessor does some preliminary work to facilitate parallel reposts (System Assurance 1 job) with minimal coupling (communication). The preprocessor job sets the status of all of the Journal Log records to Unprocessed (U) when the records are inserted. If by any chance in the System Assurance 1 Job (Repost job) where it fetches the Journal Log records to process it finds a record with the Process Id as empty, then it sets the status as Failed (F) for that record. Each Journal Log record's status is updated to Intended (I) after it is selected for processing. The status is still set as Intended even if the job fails due to any reason so it can be picked up in the next run. After the record is processed the status is updated to Final (F).

The preprocessor performs a wide array of functions the most important of which are listed below:

1. Ensure that if there are Failed (FA), Intended (I), or Unprocessed (U) Records on the journal log corresponding to the Process ID of ‘SA01JL’ or ‘SA01IJL’ or SA01LL’ on the Journal Log, then the preprocessor is not run with critical parameters (BFY and Run Mode) different from that of the earlier run.
2. If the above step does not cause a failure, then all FA or I records on the journal log are updated to the status of ‘U’.
3. Based on whether the run is a Full or Incremental Run a conditional purge and insertion is performed into peripheral tables (SA1 Workload Table, Journal Log table and SA Bud table) needed for the process. During the purge, the SA_BUD table retains all records with a posting source of ‘4’, archived.
4. Perform a complete repost of the unjournalized posting lines by disregarding the BFY.

5. Create 'U', unprocessed entries on the Journal Log, to indicate the range of records to process for the repost processes downstream. The range is calculated differently for different process IDs as shown below:

   - For SA01LL records the range is determined by the MIN and MAX Sequence Numbers (SEQ_NO) on the workload table (SA1_WRKLD).
   - For SA01IJL records the range is determined by the MIN (that is the maximum END_JRNL_REC_NO + 1 on the Journal Log table) and MAX (maximum REC_NO on the JRNL_ACTG table).
   - For SA01JL records the range is determined by the MIN and MAX Journal Accounting records that are not yet processed.

     Note: This step can be skipped by setting the Exclude Unledgerized Records (EXCL_UNLEDGR_REC) parameter to N (No). While this may improve performance, the budget lines updated with transactions that are yet to be ledgerized will show up as out of sync in the Report. Therefore, it is recommended to run the process by including the Unledgerized record on a periodic basis.

A note about the BFY parameter:

For an Incremental Run the BFY parameter is immaterial as all journal records since the last incremental or full run will be selected. For a Full Run, absence of a BFY causes the shadow table to be completely purged, with the exception of records with a posting source of '4', and then all eligible records in the ledger, its source journal, and the posting line catalog to be posted regardless of BFY.

For a Full Run, excluding a BFY parameter is not recommended because it causes the shadow table to be completely purged (with the exception of records with a posting source of '4') and all eligible ledger records and all eligible unledgerized journal records to be posted. For years that have been closed, this effort is unnecessary and should be avoided to save processing time. A full run with BFY specified signifies that a complete repost of the posting lines, journal entries and ledger records with that BFY will occur along with an incremental repost of any unledgerized journal records not having BFY equal to the Parameter BFY.

The following table shows the various steps that the SA01 Pre-Processor job goes through and the messages issued at each step.

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. Pre-processing	The following messages will be issued when the job runs in Full Mode. • Verification of the Journal Log table for any Intended, Failed or Unprocessed records started. • Verification of the Journal Log table for any Intended, Failed or Unprocessed records completed. • Purging records from the shadow table started. • Purging records from the shadow table completed. • Reposting of unjournalized posting lines to the

Process Steps	Messages
	shadow table started. • No of posting lines reposted xxx (xxx being the commit block size). • Reposting of unjournalized posting lines to the shadow table Completed. • Purging records from the SA1 Workload table started. • Purging records from the SA1 Workload table completed. • Inserting records in the workload table started. • Inserting records in the workload table completed. • Inserting workload journal records in the Journal Log table started. • Inserting workload journal records in the Journal Log table completed. • Inserting unledgerized journal records to the Journal Log table started. • Inserting unledgerized journal records to the Journal Log table completed. • Inserting unprocessed journal records to the Journal Log table started. • Inserting unprocessed journal records to the Journal Log table completed. The following messages will be issued when the job runs in Incremental Mode. • Verification of the Journal Log table for processed records started. • Verification of the Journal Log table for processed records completed. • Verification of the Journal Log table for any Intended, Failed or Unprocessed records started. • No of posting lines reposted xxx (xxx being the commit block size). • Verification of the Journal Log table for any Intended, Failed or Unprocessed records completed. • Reposting of unjournalized posting lines to the shadow table started. • Reposting of unjournalized posting lines to the shadow table completed. • Inserting workload journal records in the Journal Log table started. • Inserting unprocessed journal records to the Journal Log table started. • Inserting unprocessed journal records to the

Process Steps	Messages
	Journal Log table completed.

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

- LDGR_SA_BUD – 'Ledger SA Budgets' or other appropriate ledger specified as a parameter. The record in the Journal/Ledger Control table for the specified ledger must have System Assurance 1 checked.
- JRNL_ACTG – 'Accounting Journal' or other journal which is the source journal to the input ledger.
- PSTNG_LN_CAT - 'Posting Line Catalog' – Records selected are those with a transaction phase of Final (3) or Historical (5) whose journal posting status is not equal to Posted (3).
- JRNL_LOG – Journal Log – Records that need to be processed by System Assurance 01 will have their Status set to 'U' –Unprocessed – on the Journal Log.

Major Output

- SA_BUD - Budget Vs. Actual System Assurance Temporary Table
- JRNL_LOG – Journal Log – new records will be inserted with the appropriate process ID
- SA1_WRKLD – SA1 Workload Table

Batch Parameters

Parameter Name	Description	Default Value
BFY (Budget Fiscal Year)	Optional field. The BFY parameter helps in the formation of the 'U' entries on the Journal Log when used along with the incremental or full run parameter.	
Ledger Name (Ledger to use for assurance.	Required field. Data object name of the table to use as the input ledger. The default is the LDGR_SA_BUD, but a client may choose to use another, given it has sufficient detail. Any other value specified must be entered in the same 'all capitals' format. The ledger must summarize by at least all budget table rule elements. An invalid ledger specified will result in the job failing and an error being written to the log. There is a second part to this parameter that is not explicitly specified. The	LDGR_SA_BUD

Parameter Name	Description	Default Value
	journal used by the job is determined by finding the ledger specified on the Journal/Ledger Control table and retrieving its Source Journal.	
Run Mode (For Full Run enter 1, for Incremental Run enter 2, and for Pre-Archive Run enter 3)	Required field. Indicates if the run will be Full, Incremental or Pre-Archive. When running in Full mode, the job will process all eligible records, new and old. When running in Incremental mode, the job will process only posting line and/or journal records posted since the last time SA1 was run. When running in Pre-Archive mode, the job will process records to posting source "Archived" (4) for the FY specified in the Archive Year parameter.	1 (When running System Assurance 01 Chain)
Commit Block Size (Commit Block Size, should be a positive integer)	Required field. Commit Block Size indicates the size of the block of records written to the Journal Log (i.e. from 3001 to 4000, for a commit block size of 1000) that will be processed by each instance of the System Assurance 01 job.	
Archive FY	Pre-Archive mode only. This parameter will filter Ledger records for the FY to be archived.	
SA_RUN (System Assurance 01 Run = 1, System Assurance 02 Run = 2)	Required field. System Assurance 1/ System Assurance 2 parameter is needed because the SA Pre-Processor is the same for both SA01 and SA02. This parameter tells the pre-processor if it is being used for System Assurance 01 or System Assurance 02 processing. This parameter is preset within the chain and should not be overwritten.	1
Exclude Unledgerized Records (EXCL_UNLEDGR_REC)	Required field. Valid values are Yes (Y) and No (N) with a default of N. When set to Y, the process will skip any Journal Record that is not ledgerized.	N

Job Return Code

The following table shows the potential job return codes for the SA 01 Pre-processor job.

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	No eligible records found.
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: • Required Parameters are not entered. • Entered parameters are invalid. • If a previously run job failed and the Journal Log has records with the status of FA or I that don't match with the critical parameters (BFY and Run Mode). • If the SA_RUN is incremental and the Journal Log does not have records for the process ID SA01JL. • The Min /Max record number range is invalid. • Run time exceptions for unexpected situations. When this job ends with a return code of Failed, subsequent jobs in the chain are set to Inactive.
Terminated (16)	This return code 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 are set to Inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure subsequent jobs in the chain are set to Inactive.

Sort Criteria

None

Selection Criteria

N/A

Problem Resolution

If the job ends with a return code other than Successful after completing parameter validation, a new job should be scheduled (the failed job should not be restarted). Because a failed job's critical parameters are stored in the Journal Log table, the rescheduled job must have the same set of parameters as it was entered on the failed job with the exception of the following cases:

- If the run mode of the original job (the one which failed) was incremental, then the rescheduled job may be run in Full mode - and the other parameters on the rescheduled job do not need to be the same as on the failed job.
- If the BFY parameter value of the original job (the one which failed) was blank and the run mode was 'Full', then the rescheduled job does not need to have the same parameters of the failed job.

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

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters are valid.	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:** Ledger Name is required.	Schedule a new job after entering the Ledger Name.	
	Entered Parameters are not valid. **Sample Message:** BFY should be a valid year.	Schedule a new job after entering valid BFY.	
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated before scheduling a new job.	

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	When the Run mode is Full and BFY is not null OR Run mode is Incremental AND the critical parameter match fails. **Sample Message:** Parameter set for the UNID xyz does not match the parameter set from the previous failed run.	Schedule a new job with the same set of parameters as set on the failed job. The failed job's critical parameters are stored in the Journal Log table record's description field.	If the run mode is Incremental, alternatively, a new job can be scheduled with the Run Mode set to Full.
	Running the job with the Run Mode set to Incremental Mode before running the job in Full Mode. **Sample Message:** A Full run must be run before an Incremental run.	This scenario will arise when the job is scheduled with an Incremental mode before running the job in Full mode. Incremental run can be scheduled only after a Full run. The resolution for this error is to schedule a new job with the Run Mode set to Full.	
	Min / Max range is invalid. **Sample Message:** Journal Log record not recreated. Invalid range.	This situation only arises when the Journal Log data is corrupted. Schedule a new job with the Run Mode set to Full.	
	Job failed while performing Journal Posting / Ledger Posting. **Sample Message:** Journal Repost failed, during Journal posting. Ledger Repost failed during Ledger posting.	Analyze the reason for the failure and restart the job.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	Note: Ledger Repost message will be issued only when the Run Mode is Full.		
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated before scheduling a new job.	

Systems Assurance 01 Chain: System Assurance 1 Job

Job Name	System Assurance 1
Recommended Frequency	Daily or On Demand. This job can be run as part of the System Assurance 01 Chain or it can be run independently through a Job Interaction Client (JIC) script or manually scheduled. **Note:** This job cannot run concurrently with any of the following activities: 1) Transaction processing 2) Journal posting 3) Ledgerization
Single Instance Required	No
Can be Restarted?	Yes
Reports Generated	No

Overview

This job does the actual repost of ledger and journal records to the shadow table. Based on the process ID of the journal log record and the range contained between the begin and end journal record numbers of the Unprocessed (U) journal log record, the process determines which post it needs to perform: a ledger repost, or an incremental unprocessed journal record repost or unledgerized journal record repost.

The repost step supports check pointing and restarting. When restarted the process starts by checking if the check point contains journal log unique ids that are not 'Final' (FA) or 'Unprocessed' (U). Whenever some range of Journal Log record range is finalized it is marked as FINAL and removed from the checkpoint. Hence, the checkpoint will contain only 'FA' or 'Intended' (I) Journal Log range UNIDs.

Once the 'FA' or 'I' records contained in the checkpoint are processed completely, the repost process moves on to process existing 'U' records. It is important to note that restarted repost jobs behave in the same manner as new repost except that restarted jobs also process work designated in their checkpoint as 'FA' or "I".

The following table shows the various steps that the System Assurance 1 job goes through and the messages issued at each step.

Process Steps	Messages
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.
Reposting the records	• Processing of unprocessed Journal Log records started. • Number of Journal Log records processed: xxx (xxx being the progression counter size) • Processing of unprocessed Journal Log records completed. The following message will be issued when the job is restarted: ◦ "Restarting the processing of the Journal Log table from the previous process end point".

Restartability Information

This job can be restarted if it fails due to any reason. After restart, the job will continue to process the records based on the available check points. The job will fail if it does not find the correct check point (meaning check point records are either deleted or modified).

Major Input

- JRNL_LOG – Journal Log, which was updated by the Pre-Processor.
- SA1_WRKLD – SA1 Workload Table

- LDGR_SA_BUD – 'Ledger SA Budgets' or other appropriate ledgers specified as a parameter
- JRNL_ACTG – 'Accounting Journal' or other journal upon which the input ledger summarizes
- PSTNG_LN_CAT - 'Posting Line Catalog' – Records selected are those with a transaction phase of Final (3) or Historical (5) whose journal posting status is not equal to Posted (3)

Major Output

SA_BUD - Budget vs. Actual System Assurance Temporary Table

Batch Parameters

Description (Caption)	Parameter Name	Default Value
BFY (Budget Fiscal Year)	Optional field. Relevant for a full run only. A complete repost will occur for ledger and journal records that have the BFY or multiple BFY specified (comma-separated values). The incremental portion of a Full Run with a BFY causes the processing to resume from the largest of 'SA01IJL' and 'SA01JL' entries selecting all subsequent records **not contained in the BFY parameter**. The value must be the same as the SA01 Pre-Processor.	
Ledger Name (Ledger to use for assurance)	Required field. Data object name of the table to use as the input ledger. The default is the LDGR_SA_BUD, but a client may choose to use another, given it has sufficient detail. Any other value specified must be entered in the same 'all capitals' format. The ledger must summarize by at least all budget table rule elements. An invalid ledger specified will result in the job failing and an error being written to the log. There is a second part to this parameter that is not explicitly specified. The journal used by the job is determined by finding the ledger specified on the Journal/Ledger Control table and retrieving its Source Journal. The value must be the same as the SA01 Pre-Processor.	LDGR_SA_BUD
Run Mode (For Full Run enter 1, for	Required field. Indicates if the run will be Full, Incremental, or Pre-	

Description (Caption)	Parameter Name	Default Value
Incremental Run enter 2, and for Pre-Archive Run enter 3)	Archive. When running in Full mode, the job will process all records, new and old. Whereas when running in Incremental mode, the job will process only records posted since the last time SA1 was run. When running in Pre-Archive mode, the job will process records to posting source “Archived” (4) for the FY specified in the Archive Year parameter. The value must be the same as the SA01 Pre-Processor.	
Progression Counter Size	Optional field. During processing the job writes messages to the log to report on its progress based on the number of records already processed. When this parameter is specified, the value controls the interval at which these progression messages are written to the job log. A specified value should be a positive integer.	10
Archive FY	Pre-Archive mode only. This parameter will filter Ledger records for the FY to be archived.	

Job Return Code

The following table shows the potential job return codes for the System Assurance 1 job.

Return Code	Condition
Successful (1)	All of the selected payment records are processed successfully.
Warning (4)	N/A
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: • Required Parameters are not entered. • Parameters are invalid. • Batch parameters do not match with the Journal Log entries created by the pre-processor. • If it is a restarted job and critical parameters do not match. • If it is a restarted job and no Journal Log records are found for the UNID for the checkpoint. • If Journal posting fails. • If Ledger posting fails.

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

Sort Criteria

None.

Selection Criteria

See parameters.

Problem Resolution

No database restore is required. In case of failed jobs, restart the failed job to process work contained in its checkpoint that is not finalized. Once it successfully processes the records contained in its checkpoint range it will seek to process new unprocessed records. If the job ends with a return code other than 'Successful' after completing parameter validation, a new job should be scheduled (the failed job should not be restarted). Because a failed job's critical parameters are stored in the Journal Log table, the rescheduled job must have the same set of parameters as it was entered on the failed job with the exception of the following cases:

- If the run mode of the original job (the one which failed) was incremental, then the rescheduled job may be run in Full mode - and the other parameters on the rescheduled job do not need to be the same as on the failed job.
- If the BFY parameter value of the original job (the one which failed) was blank and the run mode was 'Full', then the rescheduled job does not need to have the same parameters of the failed job.

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

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters are valid	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
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:** Ledger Name is required.	Enter the correct Ledger name and restart the job.	Alternatively, the new job can be scheduled with the Ledger Name.
	Entered Parameters are not valid **Sample Message:** Full or Incremental Run value is not valid. The Valid values are 1 = Full Run, 2 = Incremental Run.	Enter the correct Run mode and restart the job.	Alternatively, the new job can be scheduled with the correct Run mode.
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	Investigate the reason for the termination, resolve it and restart the job. If the restart is not possible, then schedule a new job with the same set of parameters.	
System Failure (20)	When the job is terminated because of database server or network issues.	Investigate the reason for the system failure, resolve it and restart the job. If the restart is not possible, then schedule a new job with the same set of parameters.	

Step 2: Reposting the records. This step will be performed only if the parameter validation step is successful.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters are valid.	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
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)	This return code will be issued when the batch parameters do not match with the Journal Log entries created by the pre-processor. **Sample Message:** Batch parameters must be the same as xxx of Journal Log with UNID xxx.	Schedule a new job with the correct parameters as specified in the error message.	
	This return code will be issued if it is a restarted job and the critical parameter match fails. **Sample Message:** Parameter set for the UNID xyz does not match the parameter set from the previous failed run.	Schedule a new job with the same set of parameters as set on the failed job. The failed job's critical parameters are stored in the Journal Log record's description field.	
	This return code will be issued if it is a restarted job and no Journal Log entries are found for the failed job. **Sample Message:** Restart job JRNL LOG with UNID: xxx. UNID not found.	This scenario is possible when the Journal Log entries are not found (could have been deleted manually or the new chain would have been executed successfully with the run mode set to FULL) before restarting the job. If it occurs, then evaluate the scenario and schedule a new job if required.	
	Job failed while performing Journal Posting / Ledger Posting. **Sample Message:** Journal Repost failed, during Journal posting. Ledger Repost failed	Analyze the reason for the failure and restart the job.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	during Ledger posting. Note: Ledger Repost message will be issued only when the Run Mode is Full.		
	Failed because of runtime exceptions for an unexpected situation.	Investigate the reason for the Failure, resolve it and restart the job. If the job cannot be restarted immediately, then schedule a new job.	
Terminated (16)	Job is terminated manually by the user.	Investigate the reason for the termination, resolve it and restart the job. If the job cannot be restarted immediately, then schedule a new job.	
System Failure (20)	When the job is terminated because of database server or network issues.	Investigate the reason for the system failure, resolve it and restart the job. If the job cannot be restarted immediately, then schedule a new job.	

Systems Assurance 01 Chain: SA01 Report Job

Job Name	System Assurance 01 Report
Recommended Frequency	Daily or On Demand. This job can be run as part of the System Assurance 01 Chain or it can be run independently through a Job Interaction Client (JIC) script or manually scheduled.
Single Instance Required	Yes.
Can be Restarted?	No.
Reports Generated	SA01 Report is generated when out of sync records are found.

Overview

This step of the chain performs the comparisons of the entry made on the shadow table against the budget level tables. The report step is singleton. The report will indicate the records that are out of sync between the inputs being compared. It is recommended to run this job always without showing diagnostics and reporting only out of sync. If that returns a Non Fatal (meaning out of sync encountered), then run the job only for the structure that is reported as out of sync in the diagnostic mode. If the pre-archive mode has been selected, the report will not be generated. When an archive has been performed against the ledger used for the job, when records are displayed for the archived year(s), the header on the column will denote that record as being archived.

Processing Steps

Process Steps	Messages
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.
Creating SA01 Report	• Processing for Budget Structure: x and Level: y • Reviewing Budget Lines for Out of Sync Conditions. • Number of records reviewed: xxx (xxx being the progression counter size). • Review complete.

Restartability Information

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

Major Input

- Active budget level tables
- SA_BUD - Budget vs. Actual System Assurance Temporary Table

Major Output

- System Assurance 1 Report

Parameters

Batch Parameters

Description (Caption)	Parameter Name	Default Value
BFY (Budget Fiscal Year)	Optional field. Indicates the budget fiscal year for which the report should be executed. It limits the number of	

Description (Caption)	Parameter Name	Default Value
	budget records being verified. All budget records that do not have a BFY in their key are always selected. More than one BFY can be specified by separating the values with commas. The value must be the same as the SA01 Pre-Processor.	
CLIENT_NM (Client Name)	Optional field. Appears as the first line in the header of the report. The system wide default for Client Name will be used when established.	
Ledger Name (Ledger to use for assurance)	Required field. The Data object name of the table to use as the input ledger. The default is the LDGR_SA_BUD, but a client may choose to use another, given it has sufficient detail. Any other value specified must be entered in the same 'all capitals' format. The ledger must summarize by at least all budget table rule elements. An invalid ledger specified will result in the job failing and an error being written to the log. There is a second part to this parameter that is not explicitly specified. The journal used by the job is determined by finding the ledger specified on the Journal/Ledger Control table and retrieving its Source Journal. The value must be the same as the SA01 Pre-Processor.	LDGR_SA_BUD
Report All (Report all or Report out of sync only.)	Optional field. If set to **Y**, the report will list all budget records. The default for this field is **N**, which will cause the report to show only those buckets for budget records that are out of sync. Note - the report will take significantly longer to process if Report All is set to **Y**.	N
Show Budget Diagnostics (Show budget activity for buckets that are out of sync.)	Optional field. If set to **Y** the report will produce diagnostics from the activity level of a budget structure, showing all non-zero transaction activity against an out-of-sync bucket. The default is **N**, which will not show such details. Note - the report will take significantly longer to process if Show Budget Diagnostics is set to **Y**.	N
Show Identifiers	Optional field. If set to **Y** the report will	N

Description (Caption)	Parameter Name	Default Value
(Reveal miscellaneous identifiers in report for diagnostic purposes.)	reveal the internal identifiers for buckets and budget lines. These ID numbers are helpful in quickly locating budget records at the database level on the budget tables. The default is **N**, which will not show any such information.	
Show Journal Diagnostics (Show unledgerized journal records for buckets that are out of sync.)	Optional field. If set to **Y** the report will produce diagnostics for the not-yet-ledgerized journal records that should have been posted for an out of sync bucket under a separate heading of Unledgerized Journal Detail. The total of these records appears in the Recorded on Journal field of the report. Such detail will show the journal record ID, the transaction ID down to the posting line number, and the amount of the journal record. The default is **N**, which will not show any such information. Note - the report will take significantly longer to process if Show Journal Diagnostics is set to **Y**.	N
Show Ledger Diagnostics (Show ledger records (and their journal detail) for buckets that are out of sync.)	Optional field. If set to **Y** the report will produce diagnostics for the ledger records (and their summarized journal records) that should have been posted for an out of sync bucket under a separate heading of Ledger Detail. The total of these records appears in the Recorded on Ledger field of the report. Such detail will show for each journal record summarized into the ledger record matching the budget line, the journal record ID, the transaction ID down to the posting line number, and amount. The default is **N**, which will not show any such information. Note - the report will take significantly longer to process if Show Ledger Diagnostics is set to **Y**.	N
Show Posting Line Diagnostics (Show Unjournalized posting line records for buckets that are out of sync.)	Optional field. If set to **Y** the report will produce diagnostics for the not-yet-journalized posting line records that should have been posted for an out of sync bucket under a separate heading of Unjournalized Posting Line Detail. The total of these records appears in	N

Description (Caption)	Parameter Name	Default Value
	the Recorded on Posting Lines field of the report. Such detail will show the posting line ID, the transaction ID down to the posting line number, and the amount of the posting line. The default is **N**, which will not show any such information. Note - the report will take significantly longer to process if Show Posting Line Diagnostics is set to **Y**.	
Structure ID (Structure ID to report on)	Optional field. Reporting and out-of-sync checks will be performed on entered structures only. Multiple structures cannot be entered.	
Level IDs	Optional field. Can be entered only when a Structure ID is entered. Reporting and out-of-sync checks will be performed on entered levels only. Multiple structures can be entered as a comma separated list.	
Progression Counter Size	Optional field. During processing, the job writes messages to the log to report on its progress based on the number of records already processed. When this parameter is specified, the value controls the interval at which these progression messages are written to the job log. A specified value should be a positive integer.	10

Job Return Code

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

Return Code	Condition
Successful (1)	All the records are processed successfully and no out of sync condition is reported.
Warning (4)	N/A
Non Fatal Error (8)	Out of sync condition encountered.
Failed (12)	The job will fail under the following conditions: • Required parameters are not entered • Entered parameters are invalid • Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user. When this job ends with a return code of Terminated

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

Sort Criteria

The report is ordered by Budget Structure ID, Budget Level ID, and Budget Name.

Selection Criteria

See parameters.

Problem Resolution

Since the job cannot be restarted, if the job ends with any return code (Failed, Terminated or System Failure) a new job should only be scheduled with the same parameters. The first two jobs in the chain can be disabled before scheduling the new job. Alternatively, the stand alone report job can be run with the same set of parameters.

If the job ends with the return code of Non Fatal, then the out of sync condition needs to be investigated. Until the out of sync is investigated, it is advisable not to process any new transactions since that may impact the ability to complete the investigation. Run the SA01 chain again after resolving all of the issues. The out of sync condition can happen in the following scenarios:

- **Budget balances are incorrect.** This could be due to changing the budget settings after processing the transactions or incorrect settings on the REQBUD table. If the out of sync condition is because of the budget balances, then it is recommended to run the Budget Sync process to resolve the out of sync condition. The "Out of Sync Scenarios" section shows two sample scenarios of how to identify the out of sync conditions caused by the budget balances.
- **Historical balances are incorrect**. If the budget balances are correct, then the out of sync condition could be due to the historical balances. If the out of sync is because of the historical balances, then the records that caused the out of sync condition must be identified (by comparing the transaction lines against the budget activity level records) and appropriate accounting entries need to be passed through the Journal Voucher (JV) to resolve the out of sync condition.
- **Exclude Unledgerized Records Mode**. If there are budget lines updated with transactions that are yet to be ledgerized and the job is run by setting the Exclude Unledgerized Records (EXCL_UNLEDGR_REC) parameter to N (No), such budget lines will show up as out of sync in the Report. In such cases, it is recommended to run the job in include unledgerized record mode on a more periodic basis to identify the true out of sync lines.

Some of the possible reasons to investigate may be:

- Transactions with no posting lines but budget tables got updated.
- Journal entries are incorrectly ledgerized.
- Posting lines did not get journalized.

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

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Required Parameters are not entered. **Sample Message:** Ledger Name is required.	Schedule a new job after entering the Ledger Name.	
	Entered Parameters are not valid. **Sample Message:** The entered structure ID is not numeric.	Enter the correct structure ID and schedule a new job.	
	Failed because of runtime exceptions for an unexpected situation.	The Failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling a new job.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before scheduling a new job.	

Step 2: Creating SA01 Report -This step will be performed only when the parameters are valid.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	No out of sync condition encountered.	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	Out of sync condition encountered.	Verify the out of sync condition and take the appropriate action to resolve the out of sync condition.	Refer to the "Out of Sync Scenarios" section for recommended steps for analysis, verification, and resolution.
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The Failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling a new job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before scheduling a new job.	N/A

Out of Sync Scenarios

This section provides two sample scenarios of how to identify the out of sync conditions caused by the budget balances.

- Sample 1 – Accrued Expense Bucket is Out of Sync
- Sample 2 – Billed Earned Revenue Bucket is Out of Sync

Sample 1 – Accrued Expense Bucket is Out of Sync

```
REPORT ID:                                                                                                    PAGE :    8
RUN DATE:   11-28-2005                  BUDGET ACTUALS VS LEDGERS, JOURNALS, AND POSTING LINE CATALOG
RUN TIME:   18:57:10

FISCAL YEAR:   2005
PERIOD:
BUDGET TABLE:  Central Expense
BUDGET LEVEL:  2
BUDGET NAME:   BRENT'S DEMO
STRUCTURE:     BFY: 2005, Appropriation: BWB2, Fund: BWB2, Sub Fund: BWB2, Department: BWB2

-------------------------------------------------------------------------------------------------------------------------
                    RECORDED            RECORDED            RECORDED            RECORDED
                    ON LEDGER           ON JOURNAL          ON POSTING LINES    ON BUDGET TABLE     DIFFERENCE
-------------------------------------------------------------------------------------------------------------------------

Accrued Expenses
                                $1.00                                                 $500.00           $499.00
```

1) Determine the list of posting codes in the system that are set to update the budget bucket which is OOS. In this case, it's the Accrued Expense bucket. Here is the SQL:

   select * from r_pscd where exp_bud_fl = 1 and exp_bud_upd_bkt_id = 14

   SQL Result:

Tables | Execute SQLs | SQL_1

Pscd Id	Pscd Nm	Pscd Sh Nm
D021	Inventory Accrued Expenditure / Expense	Inventory AE
D024	Internal Accrued Expenditure / Expense	Internal AE
R007	Bad Debt Expense	Bad Debt Exp
A025	Accrued Operating Transfer Out	Acr Op Trf Out
D011	External Accrued Expenditure / Expense	External AE
D012	Accrued Bond Principle Expense	Acr Bnd Princ

2) Query the SA1 Input Ledger (LDGR_SA_BUD) to determine if the amount that is reported from the Ledger on the SA1 Report is correct. The query should include the posting codes from Step #1, as well as all of the Chart of Account (COA) elements that comprise the budget line. Here is the SQL:

   select sum(am) from ldgr_sa_bud where pstng_cd_id in (select pscd_id from r_pscd where exp_bud_fl = 1 and exp_bud_upd_bkt_id = 14) and fund_cd = 'BWB2' and sfund_cd = 'BWB2' and dept_cd = 'BWB2' and appr_cd = 'BWB2'

   SQL Result:

Tables | Execute SQLs | SQL_1

Sum(am)
1

3) Now query the input Ledger's (LDGR_SA_BUD) source journal (JRNL_ACTG) to determine if the amount that is reported from the Ledger matches what is recorded on the journal. Just as the previous query did, this query should include the posting codes from Step #1, as well as all of the Chart of Account (COA) elements that comprise the budget line. Here is the SQL:

   select sum(pstng_am) from jrnl_actg where pstng_cd_id in (select pscd_id from r_pscd where exp_bud_fl = 1 and exp_bud_upd_bkt_id = 14) and fund_cd = 'BWB2' and sfund_cd = 'BWB2' and dept_cd = 'BWB2' and appr_cd = 'BWB2'

   SQL Result:

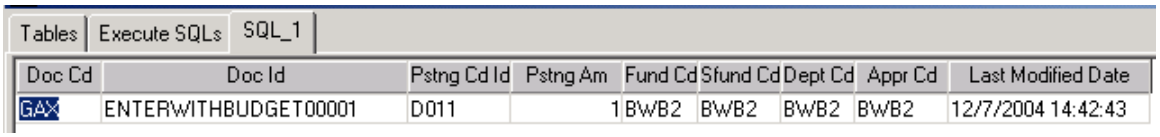
Tables | Execute SQLs | SQL_1

Sum(am)
1

4) At this point, all queries haven't revealed anything suspicious. The Ledger Amount matches the Ledger Amount reported by the SA1 Report, and the Ledger Amount matches its input Journal's total. The investigation should continue to the record level.

5) Query the source journal (JRNL_ACTG) to view the records that comprise the budget bucket amount for the budget line in question. Here is the SQL:

   select doc_cd,doc_id,pstng_cd_id,pstng_am,fund_cd,sfund_cd,dept_cd,appr_cd,last_modified_date from jrnl_actg where pstng_cd_id in (select pscd_id from r_pscd where exp_bud_fl = 1 and exp_bud_upd_bkt_id = 14) and fund_cd = 'BWB2' and sfund_cd = 'BWB2' and dept_cd = 'BWB2' and appr_cd = 'BWB2'
   SQL Result:



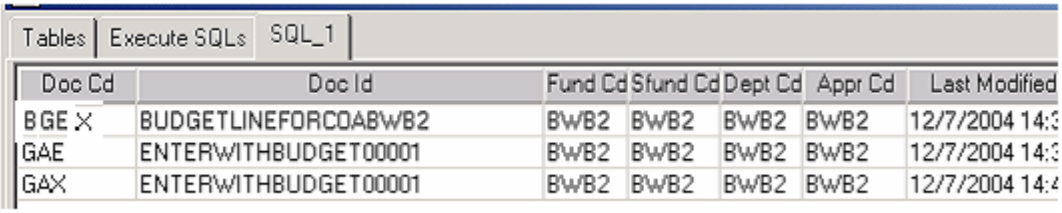
Tables | Execute SQLs | SQL_1

Doc Cd	Doc Id	Pstng Cd Id	Pstng Am	Fund Cd	Sfund Cd	Dept Cd	Appr Cd	Last Modified Date
GAX	ENTERWITHBUDGET00001	D011	1	BWB2	BWB2	BWB2	BWB2	12/7/2004 14:42:43

6) Query the budget structure's activity level to ensure the same record(s) returned from the Journal shows on budget structure. Here is the SQL:

   select doc_cd,doc_id,fund_cd,sfund_cd,dept_cd,appr_cd,last_modified_date from bud_stru_29_lvl_3 where fund_cd = 'BWB2' and sfund_cd = 'BWB2' and dept_cd = 'BWB2' and appr_cd = 'BWB2'

   SQL Result:



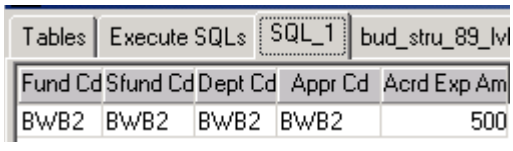
Tables | Execute SQLs | SQL_1

Doc Cd	Doc Id	Fund Cd	Sfund Cd	Dept Cd	Appr Cd	Last Modified
BGE	BUDGETLINEFORCOABWB2	BWB2	BWB2	BWB2	BWB2	12/7/2004 14:3
GAE	ENTERWITHBUDGET00001	BWB2	BWB2	BWB2	BWB2	12/7/2004 14:3
GAX	ENTERWITHBUDGET00001	BWB2	BWB2	BWB2	BWB2	12/7/2004 14:4

7) This time, the GAX Transaction found in the Journal record query appears on the budget structure activity level SQL. Also, the Last Modified Date between the two queries appear to be OK (i.e., the Budget Line was clearly established before the accounting transaction was processed).
8) Query the budget structure level for the OOS budget line and budget bucket. Here is the SQL:

   select fund_cd,sfund_cd,dept_cd,appr_cd,acrd_exp_am from bud_stru_89_lvl_2 where fund_cd = 'BWB2' and sfund_cd = 'BWB2' and dept_cd = 'BWB2' and appr_cd = 'BWB2'

   SQL Result:



Tables | Execute SQLs | SQL_1 | bud_stru_89_lvl

Fund Cd	Sfund Cd	Dept Cd	Appr Cd	Acrd Exp Am
BWB2	BWB2	BWB2	BWB2	500

9) Notice the Accrued Expense amount recorded on the budget line does not match the amount recorded on the Ledger / Journal.

Scenario #2 – Billed Earned Revenue Bucket is Out of Sync

```
REPORT ID:                                                                                    PAGE :
RUN DATE:   11-28-2005                    BUDGET ACTUALS VS LEDGERS, JOURNALS, AND POSTING LINE CATALOG
RUN TIME:   18:57:10

FISCAL YEAR:   2005
PERIOD:
BUDGET TABLE:  Central Revenue
BUDGET LEVEL:  3
BUDGET NAME:   2005-BWB1-BWB1-BWB1-BWB1
STRUCTURE:     BFY: 2005, Fund: BWB1, Sub Fund: BWB1, Department: BWB1, Revenue Source: BWB1

                     RECORDED          RECORDED          RECORDED          RECORDED
                     ON LEDGER         ON JOURNAL        ON POSTING LINES  ON BUDGET TABLE   DIFFERENCE

Billed Earned Revenue
                          $1.00                                                 $0.00            $1.00
```

1) Determine the list of posting codes in the system that are set to update the budget bucket which is OOS. In this case, it's the Billed Earned Revenue bucket. Here is the SQL:

   select * from r_pscd where rev_bud_fl = 1 and rev_bud_upd_bkt_id = 22

   SQL Result:

Tables | Execute SQLs | SQL_1

Pscd Id	Pscd Nm	Pscd Sh Nm
R002	External Billed Earned Revenue	External BER
R005	Billed NSF Revenue	Billed NSF Rev
R021	Inventory Accrued Revenue	Inventory AR
R023	Internal Accrued Revenue	Internal AR
A026	Accrued Operating Transfer In	Acr Op Trf In

2) Query the SA1 Input Ledger (LDGR_SA_BUD) to determine if the amount that is reported from the Ledger on the SA1 Report is correct. The query should include the posting codes from Step #1, as well as all of the Chart of Account (COA) elements that comprise the budget line. Here is the SQL:

   select sum(am) from ldgr_sa_bud where pstng_cd_id in (select pscd_id from r_pscd where rev_bud_fl = 1 and rev_bud_upd_bkt_id = 22) and fund_cd = 'BWB1' and sfund_cd = 'BWB1' and dept_cd = 'BWB1' and rsrc_cd = 'BWB1'

   SQL Result:

Tables | Execute SQLs | SQL_1 | SQL_1 | SQL_1

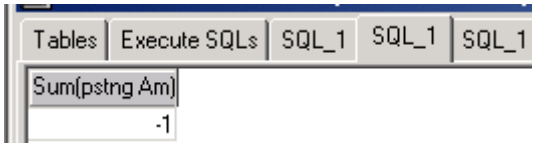
Sum(pstng Am)
-1

3) Now query the input Ledger's (LDGR_SA_BUD) source journal (JRNL_ACTG) to determine if the amount that is reported from the Ledger matches what is recorded on the journal. Just as

the previous query did, this query should include the posting codes from Step #1, as well as all of the Chart of Account (COA) elements that comprise the budget line. Here is the SQL:

select sum(pstng_am) from jrnl_actg where pstng_cd_id in (select pscd_id from r_pscd where rev_bud_fl = 1 and rev_bud_upd_bkt_id = 22) and fund_cd = 'BWB1' and sfund_cd = 'BWB1' and dept_cd = 'BWB1' and rsrc_cd = 'BWB1'

SQL Result:



4) At this point, all queries haven't revealed anything suspicious. The Ledger Amount matches the Ledger Amount reported by the SA1 Report, and the Ledger Amount matches its input Journal's total. The investigation should continue to the record level.

5) Query the source journal (JRNL_ACTG) to view the records that comprise the budget bucket amount for the budget line in question. Here is the SQL:

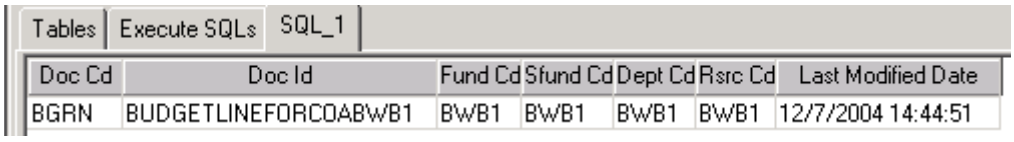
select doc_cd,doc_id,pstng_cd_id,pstng_am,fund_cd,sfund_cd,dept_cd,rsrc_cd,last_modified_date from jrnl_actg where pstng_cd_id in select pscd_id from r_pscd where rev_bud_fl = 1 and rev_bud_upd_bkt_id = 22) and fund_cd = 'BWB1' and sfund_cd = 'BWB1' and dept_cd = 'BWB1' and rsrc_cd = 'BWB1'

SQL Result:

6) Query the budget structure's activity level to ensure the same record(s) returned from the Journal show on budget structure. Here is the SQL:

select doc_cd,doc_id,fund_cd,sfund_cd,dept_cd,rsrc_cd,last_modified_date from bud_stru_30_lvl_4 where fund_cd = 'BWB1' and sfund_cd = 'BWB1' and dept_cd = 'BWB1' and rsrc_cd = 'BWB1'

SQL Result:



7) Notice the budget line SQL did not return the RE Transaction found by the Journal SQL. Also notice the Last Modified Date of the two SQL's.

2.1.2 Systems Assurance 02 – Accounting Control Tables Versus Ledgers, Journals, and Posting Line Catalog

Chain Name	System Assurance 02
Recommended Frequency	Daily or On Demand. **Note:** With the exception of the report, System Assurance 02 jobs cannot run concurrently with any of the following activities. 1) Transaction processing 2) Journal Engine 3) Ledger Engine
Single Instance Required	Yes
Can be Restarted?	N/A
Reports Generated	Yes, when the mode is not Pre-Archive, the SA02 out of sync report will be generated if the job finds any out of sync records.

Overview

The System Assurance 02 process compares entries on the **Inception-To-Date (ITD) Balance Sheet Detail** (BBALD) table, the **Cash Balance Detail** (CBALDQ), and the **Fund Balance Detail** (FBALDQ) tables and verifies them against the input ledger, journal records not yet ledgerized in the supporting journal of that ledger, and records in the posting line catalog not yet in the supporting journal. Not every bucket on these three tables is validated. For BBALD, the *Balance* bucket (the end result of the total debits and total credits) is checked. For CBALDQ, the *Non-Cash Increases* and *Non-Cash Decreases* are checked. The starting *Cash Balance* amount was checked when BBALD was verified. For FBALDQ, the *Revenue Account Balance and Expense Account Balance* amounts are checked. The stating *Fund Balance* amount was checked when BBALD was verified.

The process has 3 sources of information. The first source is the posting line catalog. All records with a journal posting status of 'ready' or 'not ready' are selected as long as the line has a transaction phase of 'Final' or 'Historical Final.' This group contains all of the posting lines that have not yet been posted to the journal, but could have been posted. The second source is the journal, which supports the input ledger. The **Journal Log** is read to determine the last record from that journal posted to the input ledger. When that record number is less than the current record number on that journal, all journal record numbers greater than the last posted record are selected. This gets all journal records not yet posted to the ledger, but could have been posted. The third source is the input ledger. All of its records are initially selected as input. In selection of records from all three sources, further selection of just those records that update BBALD, CBALDQ, or FBALDQ will occur.

The System Assurance 02 process can be run in Full mode, Incremental mode, or Pre-Archive mode. When running in Full mode, the job will process all records, new and old. In Incremental mode, the job will process only journal records posted since the last time the SA02 job was run. In Pre-Archive mode, the job will process records for a single FY in preparation for running the Journal/Ledger Archiving chain job.

The following jobs are an integral part of the System Assurance 02 Process:

1. SA02 Pre-processor

2. System Assurance 2
3. SA02 Report

The SA02 Pre-processor and SA02 Report jobs are singleton jobs, meaning that only one of their instances can be run. The System Assurance 2 job (repost step) has been designed such that multiple instances may run without re-doing each others work. If a site wants to exploit the multiple repost processing, they are advised to use a Job Interaction Client (JIC) script or manually schedule multiple repost steps to get triggered at the same time (possible via online access). Both of these options require running these component jobs outside of the System Assurance 02 Chain. To facilitate this, separate catalog entries have been provided for each component System Assurance 02 job so that the Job Interaction Client script can be used to trigger multiple System Assurance 2 jobs (the repost step).

The first output from the process is a SA2 temporary table that is used by the job. The second output is a report produced that shows those records on CBALDQ, FBALDQ, and BBALS (in that order) that are out-of-sync with the posting line catalog, journal, and ledger. Records for the CBALDQ and FBALDQ tables are sorted by fund. Those for BBALS are sorted by fund and balance sheet account. Within each of these three areas, those 'BAL' entries that had at least one record in one of the three sources of information are listed first. Then after that, records on a 'BAL' table that have no supporting source information are listed. Since the job cannot determine which records from the posting line catalog, journal, or ledger are responsible for any difference found, all records are listed if there parameter says to display them. The ledger records are listed first with the heading "Ledger Detail" then followed by the journal records as "Unledgerized Journal Detail" and finally the posting lines as "Unjournalized Posting Line Detail."

Each of the 'BAL' tables has a different format that is used to display the data that is out-of-sync:

The one for BBALS has five columns for summary information: Ledger Balance, Journal Balance, Posting Line Balance, BBALS (called 'Table') Balance, and Total Difference. The difference is calculated by subtracting the sum of the ledger, journal & posting line balances from the BBALS balance. The difference is specified in absolute value terms. For diagnostic detail, information from ledgers is displayed starting with the ledger record number. For that ledger record, each journal record that makes up that ledger record according to the **Journal Ledger XREF** table is displayed along with the transaction ID down to the posting line number. The total for that journal line is also displayed as stored, so negatives are credits and positives are debits. Information from journals is displayed just as that from ledgers except there is no ledger record number. Information from the posting line catalog is displayed with the posting line number in the same column that ledger record numbers appear, along with an amount and the transaction ID.

The one for CBALDQ has ten columns for summary information: Increase and Decrease amounts for the Ledger, Journal, Posting Line Catalog, CBALDQ (called 'Table'), and Difference calculations. The reason for the dual buckets is that Non-Cash Increases as well as Non-Cash Decreases are validated for CBALDQ. For diagnostic detail, information is displayed just as it is for the BBALS table. The difference is calculated in the same way, except that only increase buckets are used for the Increase Difference and vice versa for the Decrease Difference.

The one for FBALDQ has ten columns for summary information: an Expense Account Balance (EAB) and Revenue Account Balance (RAB) field for the Ledger, Journal, Posting Line Catalog, BBALS (called 'Table'), and Difference calculations. The difference is calculated by subtracting the sum of the ledger, journal & posting line balances from the BBALS balance. For diagnostic detail, information is displayed just as it is for the BBALS table.

The acceptable job return codes (configured in the Configure Chain Job section of the Job Setup in CGI Advantage) for the jobs in the System Assurance 02 Chain are delivered to be set to Successful. As with all CGI Advantage chain jobs, these acceptable return codes are configurable and may be changed to meet certain requirements. For simplicity, this run sheet assumes the chain is configured as delivered, including parameter settings.

In Pre-Archive mode, the job will empty all records from the System Assurance 02 temporary table (SA_NONBUD) except for those with Posting Source “Archived” (4). The System Assurance 2 step will add/make updates to these “Archived” records, and the SA02 Report step will be made inactive when running as a chain job. The Journal/Ledger Archiving chain then deletes the ledger records for the Archive FY at the end of the process. A Full run of the Systems Assurance 2 process is expected at any time after archiving is complete. Please plan to allow for time after an archive for this full SA02 run.

Major Input

- LDGR_FYDAD – (Or some other appropriate ledger specified as a parameter. The key is that the ledger contains FY, Fund, Sub Fund, BSA, and Sub BSA. The ledger used for Annual Close is such a ledger and would perform very well as the input.)
- JRNL_ACTG – ‘Accounting Journal’ (or some other journal which is the source journal to the input ledger).
- PSTNG_LN_CAT - ‘Posting Line Catalog’ – Records selected are those with a transaction phase of Final (3) or Historical (5) whose journal posting status is not equal to Posted (3).
- Batch parameters: These parameters are discussed in detail later.
- R_PSCD – Posting Code table to supply Cash Balance and Fund Balance settings
- R_BBAL_ITD – ITD Balance Sheet Detail (BBALD)
- R_CBAL – Cash Balance Detail (CBALDQ)
- R_FBAL – Fund Balance Detail (FBALDQ)
- JRNL_LOG – Journal Log. In the Journal Log, the job looks for the records with the following process IDs:
  - **SA02LL** (identifies the ledger / work load records) a range specified that starts and ends with the SEQ_NO values on the corresponding SA*_WKLD table).
  - **SA02JL** (identifies the journal records that have not been ledgerized yet)
  - **SA02IJL (**identifies the unprocessed records)

Major Output

- JRNL_LOG – Journal Log – Records that need to be processed by System Assurance 02 will have their Status set to ‘U’ –Unprocessed – on the Journal Log.
- SA2_WRKLD SA2 Workload Table
- SA_NONBUD – Non Budget Vs. Actual System Assurance Temporary Table
- System Assurance 2 Report

Chain Job Return code

The following table shows the potential return codes for System Assurance 02 Chain. Note that the Chain job will end with the highest return code across all of the jobs.

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

	Error”
Failed (12)	One of the jobs in the chain ends with a return code of “Failed”
Terminated (16)	One of the jobs in the chain ends with a return code of “Terminated”
System Failure (20)	One of the jobs in the chain ends with a return code of “System Failure”

Problem Resolution

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

System Assurance 02 Chain: SA02 Pre-Processor Job

Job Name	SA 02 Pre-processor
Recommended Frequency	Daily or On Demand. This job can be run as part of the System Assurance 02 Chain or it can be run independently through a Job Interaction Client (JIC) script or manually scheduled. **Note:** This job cannot run concurrently with any of the following activities. 1) Transaction processing 2) Journal Engine 3) Ledger Engine
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	No

Overview

As the name suggests, the preprocessor does some preliminary work to facilitate parallel reposts (System Assurance 2 job) with minimal coupling (communication). The preprocessor job sets the status of all of the Journal Log records to Unprocessed (U) when the records are inserted. If by any chance in the System Assurance 2 Job (Repost job) where it fetches the Journal Log records to process it finds records with the Process Id as empty then it sets the status as Failed (FA) for that record. Each Journal Log record's status is updated to Intended (I) after it is selected for processing. The status is still set as Intended even if the job fails due to any reason so it can be picked up in the next run. After the record is processed, the status is updated to Final (F).

The preprocessor performs a wide array of functions the most important of which are listed below:

1. Ensure that if there are Failed (FA) or Intended (I) or Unprocessed (U) Records on the Journal Log corresponding to a Process ID of ‘SA02JL’ or ‘SA02IJL’ or SA02LL’ on the Journal Log, then the preprocessor is not run with a critical parameter (run mode) different from that of the earlier run.
2. If the above step does not cause a failure, then all FA or I records on the Journal Log are updated to the status of ‘U’.

3. Based on whether the run is a Full or Incremental Run a conditional purge and insertion is performed into peripheral tables (that is, the SA2 Workload Table, Journal Log table and SA Non Bud table).
4. Perform a complete repost of the unjournalized posting lines.
5. Create 'U' - unprocessed entries on the Journal Log to indicate the range of records to process for the repost processes downstream. The range is calculated differently for different process IDs as shown below:
   - For SA02LL records the range is determined by the MIN and MAX Sequence Numbers (SEQ_NO) on the workload table (SA2_WRKLD).
   - For SA02IJL records the range is determined by the MIN (that is the maximum END_JRNL_REC_NO + 1 on the Journal Log table) and MAX (maximum REC_NO on the JRNL_ACTG table).
   - For SA02JL records the range is determined by the MIN and MAX Journal Accounting records that are not yet processed.

     Note: This step can be skipped by setting the Exclude Unledgerized Records (EXCL_UNLEDGR_REC) parameter to N (No). While this may improve performance, the budget lines updated with transactions that are yet to be ledgerized will show up as out of sync in the Report. Therefore, it is recommended to run the process by including the Unledgerized record on a periodic basis.

For a FULL run SA_NONBUD is completely purged, with the exception of records with a posting source of archived ('4'), and eligible ledger records along with all eligible unledgerized journal records are posted to the shadow table.

Performance Note:

- The input ledger into each of the job steps of the SA02 chain, as well as the individual job steps used with the Job Interaction Client (JIC) software is one of three critical performance components. That input ledger should be one that is streamlined to contain just information necessary for the assurance job. While the Systems Assurance ledger may be used for SA01 and SA02, the level of detail needed for each assurance is quite different. SA01 needs all COA that are used to define budget lines along with a BFY. SA02 only needs Fiscal Year, Fund, Sub Fund, BSA, and Sub BSA to perform assurances. It is suggested that such a ledger exist for each implementation for Systems Assurance 2 process. The same ledger would also be a likely input source for Annual Close as well.

  Note: SA01 and SA02 are delivered with the default ledger for both to be the LDGR_FYDAD as system resources to support two different ledgers cannot be assumed. It is recommended that if two ledgers are to be implemented so that SA02 has a different input ledger, that all Job Setup entries for the SA02 jobs be changed to have the new ledger as the default.
- The second performance component would be to ensure that all posting lines have been journalized by running the Journal Posting Initiator batch job as well as the Journal Posting initiator.
- The third performance component would be to ensure that all journal records have been ledgerized by running the Ledger Engine.
- Of special note for running in pre-archive mode is that the input ledger does not have Budget FY retained but only has Fiscal Year. To run in this mode where both years are in the ledger will result in SA_NONBUD being updated so that out of sync conditions will be falsely reported upon the next run in full mode.

The following table shows the various steps that the SA02 Pre-processor job goes through and the messages issued at each step.

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. Pre-processing	The following messages will be issued when the job runs in Full Mode. • Verification of the Journal Log table for any Intended, Failed or Unprocessed records started. • Verification of the Journal Log table for any Intended, Failed or Unprocessed records completed. • Purging records from the shadow table started. • Purging records from the shadow table completed. • Reposting of unjournalized posting lines to the shadow table started. • No of posting lines reposted xxx (xxx being the commit block size). • Reposting of unjournalized posting lines to the shadow table completed. • Purging records from the SA2 Workload table started. • Purging records from the SA2 Workload table completed. • Inserting records in the workload table started. • Inserting records in the workload table completed. • Inserting workload journal records in the Journal Log table started. • Inserting workload journal records in the Journal Log table completed. • Inserting unledgerized journal records to the Journal Log table started. • Inserting unledgerized journal records to the Journal Log table completed. • Inserting unprocessed journal records to the Journal Log table started. • Inserting unprocessed journal records to the Journal Log table completed. The following messages will be issued when the job runs in Incremental Mode.

Process Steps	Messages
	• Verification of the Journal Log table for processed records started. • Verification of the Journal Log table for processed records completed. • Verification of the Journal Log table for any Intended, Failed or Unprocessed records started. • No of posting lines reposted xxx (xxx being the commit block size). • Verification of the Journal Log table for any Intended, Failed or Unprocessed records completed. • Reposting of unjournalized posting lines to the shadow table started. • Reposting of unjournalized posting lines to the shadow table completed. • Inserting workload journal records in the Journal Log table started. • Inserting unprocessed journal records to the Journal Log table started. • Inserting unprocessed journal records to the Journal Log table completed.

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

- LDGR_FYDAD – (Or some other appropriate ledger specified as a parameter. The key is that the ledger contains FY, Fund, Sub Fund, BSA, and Sub BSA. The record in the Journal/Ledger Control (JLCTRL) table for the specified ledger must have System Assurance 02 checked.
- JRNL_ACTG – 'Accounting Journal' (Or some other journal which is the source journal to the input ledger.)
- PSTNG_LN_CAT - 'Posting Line Catalog' – Records selected are those with a transaction phase of Final (3) or Historical (5) whose journal posting status is not equal to Posted (3).
- JRNL_LOG – Journal Log – Records that need to be processed by System Assurance 02 will have their Status set to 'U' –Unprocessed – on the Journal Log.
- JLXREF – Journal Ledger Cross Reference table.

Batch Parameters

Parameter Name	Description	Default Value
Ledger Name	Required field. Data object name of the	LDGR_FYDAD

Parameter Name	Description	Default Value
(Ledger to use for assurance.)	table to use as the input ledger. The ledger must summarize by Fiscal Year, Fund and BSA. If used, Sub Fund and Sub BSA must also be present. The journal used by the job is determined by finding the ledger specified on the Journal/Ledger Control table and retrieving its Source Journal.	
Run Mode (For Full Run enter 1, for Incremental Run enter 2, and for Pre-Archive Run enter 3)	Required field. Indicates if the run will be Full, Incremental, or Pre-Archive. When running in Full mode, the job will process all eligible records, new and old. Whereas when running in Incremental mode, the job will process only posting line and/or journal records posted since the last time SA1 was run. When running in Pre-Archive mode, the job will process records to posting source “Archived” (4) for the FY specified in the Archive Year parameter.	1 (When running System Assurance 02 Chain)
Commit Block Size (Commit Block Size, should be a positive integer)	Required field. Commit block size indicates the size of the block of records written to the Journal Log (that is, from 3001 to 4000, for a commit block size of 1000) that will be processed by each instance of the System Assurance 02 job.	1000
Archive FY	Pre-Archive mode only. This parameter will filter Ledger records for the FY to be archived.	
SA_RUN (System Assurance 01 Run = 1, System Assurance 02 Run = 2)	Required field. System Assurance 1/ System Assurance 2 parameter is needed because the SA Pre-Processor is the same for both SA01 and SA02. This parameter tells the pre-processor if it is being used for System Assurance 01 or System Assurance 02 processing. This parameter is preset within the chain and should not be overwritten.	2
Exclude Unledgerized Records (EXCL_UNLEDGR_REC)	Required field. Valid values are Yes (Y) and No (N) with a default of N. When set to Y, the process will skip any Journal Record that is not ledgerized.	N

Major Output

- SA_NONBUD – Non Budget Vs. Actual System Assurance Temporary Table

- JRNL_LOG (Journal Log) – New records will be inserted with the appropriate process ID
- SA2_WRKLD – SA2 Workload Table

Job Return code

The following table shows the potential job return codes for the SA 02 Pre-processor job.

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	No eligible records found.
Non Fatal Error (8)	N/A.
Failed (12)	The job will fail under the following conditions: • Required Parameters are not entered • Entered parameters are invalid • If a previously run job failed and the Journal Log has records with the status of FA or I that don't match with the critical parameter (Run Mode). • If the SA_RUN is incremental and the Journal Log does not have records for the process ID SA02JL. • The Min /Max record number range is invalid. • Run time exceptions for unexpected situations. When this job ends with a return code of Failed, subsequent jobs in the chain are set to inactive.
Terminated (16)	This return code 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 are set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job ends with a return code of System Failure, subsequent jobs in the chain are set to inactive.

Sort Criteria

None.

Selection Criteria

N/A.

Problem Resolution

If the job ends with a return code other than ‘Successful’ after completing parameter validation, a new job should be scheduled (the failed job should not be restarted). Because a failed job’s

critical parameters are stored in the Journal Log table, the rescheduled job must have the same set of parameters as it was entered on the failed job with the exception of the following scenario:

If the run mode of the original job (the one which failed) was incremental, then the rescheduled job may be run in Full mode - and the other parameters on the rescheduled job need not be the same as on the failed job.

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

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters are valid.	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:** Ledger Name is required.	Schedule a new job after entering the Ledger Name.	
	Entered Parameters are not valid. **Sample Message:** SA_RUN is not valid. Valid values are 1 – SA01, 2 – SA02.	Schedule a new job after entering a valid SA_RUN.	
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated before scheduling a new job.	

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the selected records processed successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Critical parameter match fails. **Sample Message:** Parameter set for the UNID xyz does not match the parameter set from the previous failed run.	Schedule a new job with the same set of parameters as set on the failed job. The failed job's critical parameters are stored in the Journal Log Table record's description field.	If the run mode is Incremental, alternatively, a new job can be scheduled with the Run Mode set to Full.
	Running the job with the Run mode set to Incremental Mode before running the job in Full Mode. **Sample Message:** A Full run must be run before an Incremental run.	This scenario will arise when the job is scheduled with an Incremental mode before running the job in Full mode. Incremental run can be scheduled only after a Full run. The resolution for this error is to schedule a new job with the Run mode set to Full.	
	Min / Max range is invalid. **Sample Message:** Journal Log record not recreated. Invalid range.	This situation only arises when the Journal Log data is corrupted. Schedule a new job with the Run mode set to Full.	
	Job failed while performing Journal Posting / Ledger Posting. **Sample Message:** Journal Repost failed, during	Analyze the reason for the failure and restart the job.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	Journal posting. Ledger Repost failed during Ledger posting. Note: Ledger Repost message will be issued only when the Run Mode is Full.		
	Failed because of runtime exceptions for an unexpected situation.	Failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling a new job.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before scheduling a new job.	

Systems Assurance 02 Chain: System Assurance 2 Job

Job Name	System Assurance 2
Recommended Frequency	Daily or On Demand. This job can be run as part of the System Assurance 02 Chain or it can be run independently through a Job Interaction Client (JIC) script or manually scheduled. **Note:** This job cannot run concurrently with any of the following activities. 1) Transaction processing 2) Journal Engine 3) Ledger Engine.
Single Instance Required	No
Can be Restarted?	Yes

Reports Generated	No

Overview

This job does the actual repost of ledger and journal records to the shadow table. Based on the process ID of the journal log record and the range contained between the begin and end journal record numbers of the Unprocessed (U) journal log record, the process determines which post it needs to perform: a ledger repost, or an incremental unprocessed journal record repost or unledgerized journal record repost.

The repost step supports check pointing and restarting. When restarted, the process starts by checking if the check point contains journal log unique ids that are not 'Final' (FA) or 'Unprocessed' (U). Whenever some range of Journal Log record range is finalized it is marked as FINAL and removed from the checkpoint. Hence the checkpoint will contain only 'FA' or 'Intended' (I) Journal Log range UNIDs.

Once the 'FA' or 'I' records contained in the checkpoint are processed completely, the repost process moves on to process existing 'U' records. It is important to note that restarted repost jobs behave in the same manner as new repost except that restarted jobs also process work designated in their checkpoint as 'FA' or "I".

The following table shows the various steps that the System Assurance 2 job goes through and the messages issued at each step

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. Reposting the records	• Processing of unprocessed Journal Log records started. • Number of Journal Log records processed: xxx (xxx being the progression counter size) • Processing of unprocessed Journal Log records completed. The following message will be issued when the job is restarted: "Restarting the processing of Journal Log table from the previous process end point".

Restartability Information

This job can be restarted if it fails due to any reason. After restart, the job will continue to process the records based on the available check points. The job will fail if it does not find the correct check point (meaning check point records are either deleted or modified).

Major Input

- JRNL_LOG – Journal Log, which was updated by the Pre-Processor.
- SA2_WRKLD – SA2 Workload Table
- LDGR_FYDAD – or other appropriate ledger specified as a parameter.
- JRNL_ACTG – 'Accounting Journal' or other journal upon which the input ledger summarizes.
- PSTNG_LN_CAT - 'Posting Line Catalog' – Records selected are those with a transaction phase of Final (3) or Historical (5) whose journal posting status is not equal to Posted (3).
- Batch Parameters

Description (Caption)	Parameter Name	Default Value
Ledger Name (Ledger to use for assurance)	Required field. Data object name of the table to use as the input ledger. The default is the LDGR_FYDAD, but a client may choose to use another, given it has sufficient COA detail. Any other value specified must be entered in the same 'all capitals' format. The ledger must summarize by at least on Fiscal Year, Fund, and BSA. If used, Sub Fund and Sub BSA must also be present. An invalid ledger specified will result in the job failing and an error being written to the log. There is a second part to this parameter that is not explicitly specified. The journal used by the job is determined by finding the ledger specified on the Journal/Ledger Control table and retrieving its Source Journal. The value must be the same as the SA02 Pre-Processor.	LDGR_SA_BUD
Run Mode (For Full Run enter 1, for Incremental Run enter 2, and for Pre-Archive Run enter 3)	Required field. Indicates if the run will be Full or Incremental. When running in Full mode, the job will process all records, new and old. Whereas when running in Incremental mode, the job will process only records posted since the last time SA1 was run. When running in Pre-Archive mode, the job will process records to posting source "Archived" (4) for the FY specified in the Archive Year parameter. The value must be the same as the SA02 Pre-Processor.	
Progression Counter Size	Optional field. During processing the job writes messages to the log to report on its progress based on	500

Description (Caption)	Parameter Name	Default Value
	the number of records already processed. When this parameter is specified, the value controls the interval at which these progression messages are written to the job log. A specified value should be a positive integer.	
Archive FY	Pre-Archive mode only. This parameter will filter Ledger records for the FY to be archived.	

Major Output

SA_NONBUD – Non-Budget vs. Actual System Assurance Temporary Table

Job Return code

The following table shows the potential job return codes for the System Assurance 2 job.

Return Code	Condition
Successful (1)	All of the selected payment records are processed successfully.
Warning (4)	N/A
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: • Required Parameters are not entered • Parameters are invalid • Batch parameters do not match with the Journal Log entries created by the pre-processor • If it is a restarted job and critical parameters do not match • If it is a restarted job and no Journal Log records are found for the UNID for the checkpoint • If Journal posting fails • If Ledger posting fails • 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 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 are set to inactive.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues. When this job

Return Code	Condition
	ends with a return code of System Failure subsequent jobs in the chain are set to inactive.

Sort Criteria

None.

Selection Criteria

See parameters.

Problem Resolution

No database restore is required. In case of failed jobs, restart the failed job to process work contained in its checkpoint that is not finalized. Once it successfully processes the records contained in its checkpoint range it will seek to process new unprocessed records. If the job ends with a return code other than 'Successful' after completing parameter validation, a new job should be scheduled (the failed job should not be restarted). Because a failed job's critical parameters are stored in the Journal Log table, the rescheduled job must have the same set of parameters as it was entered on the failed job with the exception of the following scenario:

- If the run mode of the original job (the one which failed) was incremental, then the rescheduled job may be run in Full mode - and the other parameters on the rescheduled job do not have to be the same as on the failed job.

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

Step 1: Parameter Validation:

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters are valid.	N/A	N/A
Warning (4)	N/A	This step doesn't issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Required Parameters are not entered. **Sample Message:** Ledger Name is required.	Enter the correct Ledger name and restart the job.	Alternatively, a new job can be scheduled with the Ledger Name.
	Entered Parameters are not valid.	Enter the correct Run mode and restart the job.	Alternatively, a new job can be scheduled with

Possible Return Codes	Condition	Recommendation	Other Instructions
	Sample Message: Full or Incremental Run value is not valid. The Valid values are 1 = Full Run, 2 = Incremental Run		the correct Run mode.
	Failed because of runtime exceptions for an unexpected situation.	The failure reason needs to be investigated before scheduling a new job.	
Terminated (16)	The job is terminated manually by the user.	Investigate the reason for the termination, resolve it and restart the job. If the restart is not possible, then schedule a new job with the same set of parameters.	
System Failure (20)	When the job is terminated because of database server or network issues.	Investigate the reason for the system failure, resolve it and restart the job. If the restart is not possible, then schedule a new job with the same set of parameters.	

Step 2: Reposting the records. This step will be performed only if the parameter validation step is successful.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters are valid	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)	This return code will be issued when the batch parameters do not match with the Journal Log entries created by the pre-processor. **Sample Message:** Batch parameters must be the	Schedule a new job with the correct parameters as specified in the error message.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	same as xxx of Journal Log with UNID xxx.		
	This return code will be issued if it is a restarted job and the critical parameter match fails. **Sample Message:** Parameter set for the UNID xyz does not match the parameter set from the previous failed run.	Schedule a new job with the same set of parameters as set on the failed job. The failed job's critical parameters are stored in the Journal Log record's description field.	
	This return code will be issued if it is a restarted job and no Journal Log entries are found for the failed job. **Sample Message:** Restart job JRNL LOG with UNID: xxx. UNID not found.	This scenario is possible when the Journal Log entries are not found (could have been deleted manually or new chain would have been executed successfully with run mode set to FULL) before restarting the job. If it occurs, then evaluate the scenario and schedule a new job if required.	
	Job failed while performing Journal Posting / Ledger Posting. **Sample Message:** Journal Repost failed, during Journal posting. Ledger Repost failed during Ledger posting. Note: Ledger Repost message will be issued only when the Run Mode is Full.	Analyze the reason for the failure and restart the job.	
	Failed because of runtime exceptions for an unexpected situation.	Investigate the reason for the Failure, resolve it and restart the job. If the job cannot be restarted	

Possible Return Codes	Condition	Recommendation	Other Instructions
		immediately, then schedule a new job.	
Terminated (16)	Job is terminated manually by the user.	Investigate the reason for the termination, resolve it and restart the job. If the job cannot be restarted immediately, then schedule a new job.	
System Failure (20)	When the job is terminated because of database server or network issues.	Investigate the reason for the system failure, resolve it and restart the job. If the job cannot be restarted immediately, then schedule a new job.	

Systems Assurance 02 Chain: SA02 Report Job

Job Name	System Assurance 02 Report
Recommended Frequency	Daily or On Demand. This job can be run as part of the System Assurance 02 Chain or can be run independently through a Job Interaction Client (JIC) script or manually scheduled.
Single Instance Required	Yes
Can be Restarted?	No
Reports Generated	Yes, when run mode is not Pre-Archive.

Overview

This step of the chain performs the comparisons of the entry made on the shadow table against the **ITD Balance Sheet Summary** (BBALS) table, the **Cash Balance Detail** (CBALDQ), and the **Fund Balance Detail** (FBALDQ) tables. The report step is singleton. If nothing is out of sync, the report will be generated with the following text “No out of sync condition found”. If there is an out of sync condition, then the report will indicate the records that are out of sync between the inputs being compared. It is also possible to report the out of sync records for specific tables by setting the batch parameters appropriately. If either Fund Balance or Cash Balance features are not used because either posting code settings have been turned off or settings have been left on, but no errors are ever issued for transaction processing, then that table should not be included in the report to improve processing. For example, if the out of sync condition is reported only in CBALDQ table, then the job can be run only for the CBLADQ table to report all of the out of sync records. It is recommended to run this job always without showing diagnostics and reporting only out of sync. If the job returns a Non Fatal (meaning out of sync encountered), then run the job only for the balance that is reported as out of sync in the diagnostic mode. If the pre-archive

mode has been selected, this report will not be generated. When an archive has been performed against the ledger used for the job, when records are displayed for the archived year(s), the header on the column will denote that record as being archived.

Processing steps

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. Creating SA02 Report	Progression messages when the CBALDQ records are reviewed: • Selecting eligible records from Cash Balance Detail table. • Selection of records completed. • Reviewing Cash Balance Detail records for Out of Sync Conditions. • Number of records reviewed: xxx (xxx being the progression counter size). • Selecting Cash Balance from the SA_NONBUD table. • Selection of records completed. • Reviewing Cash Balance records on the SA_NONBUD table for Out of Sync Conditions. • Number of records reviewed: xxx (xxx being the progression counter size). • Review complete. Progression messages when the FBALDQ records are reviewed: • Selecting eligible records from the Fund Balance Detail table. • Selection of records completed. • Reviewing Fund Balance Detail records for Out of Sync Conditions. • Number of records reviewed: xxx (xxx being the progression counter size). • Selecting Fund Balance records from the SA_NONBUD table. • Selection of records completed. • Reviewing Fund Balance records on the

Process Steps	Messages
	SA_NONBUD table for Out of Sync Conditions. • Number of records reviewed: xxx (xxx being the progression counter size). • Review complete. Progression messages when the BBALS records are reviewed: • Selecting eligible records from ITD Balance Sheet table. • Selection of records completed. • Reviewing ITD Balance Sheet records for Out of Sync Conditions. • Number of records reviewed: xxx (xxx being the progression counter size). • Selecting ITD Balance Sheet records from the SA_NONBUD table. • Selection of records completed. • Reviewing ITD Balance Sheet records on the SA_NONBUD table for Out of Sync Conditions. • Number of records reviewed: xxx (xxx being the progression counter size). • Review complete.

Restartability Information

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

Major Input

- SA_NONBUD – Non-Budget vs. Actual System Assurance Temporary table
- R_CBALDQ – Cash Balance Detail table
- R_FBALDQ – Fund Balance Detail table
- R_BBAL – ITD Balance Sheet Detail table

Batch Parameters

Description (Caption)	Parameter Name	Default Value
CLIENT_NM (Client Name)	Optional field. Appears as the first line in the header of the report. The system wide default for Client Name	

Description (Caption)	Parameter Name	Default Value
	will be used when established.	
Ledger Name (Ledger to use for assurance)	Required field. Data object name of the table to use as the input ledger. The default is the LDGR_FYDAD, but a client may choose to use another, given it has sufficient detail. Any other value specified must be entered in the same 'all capitals' format. The ledger must summarize by at least Fiscal Year, Fund, and BSA. If used, Sub Fund and Sub BSA must also be present. An invalid ledger specified will result in the job failing and an error being written to the log. The value must be the same as the SA02 Pre-Processor.	LDGR_FYDAD
Report All (Report all or Report out of sync only)	Optional field. If set to **Y**, the report will list all budget records. The default for this field is **N**, which will cause the report to show only those buckets for budget records that are out of sync. Note - the report will take significantly longer to process if Report All is set to **Y**.	N
Show Journal Diagnostics (Show unledgerized journal records for buckets that are out of sync)	Optional field. If set to **Y** the report will produce diagnostics for the not-yet-ledgerized journal records that should have been posted for an out of sync bucket under a separate heading of Unledgerized Journal Detail. The total of these records appears in the Recorded on Journal field of the report. Such detail will show the journal record ID, the transaction ID down to the posting line number, and the amount of the journal record. The default is **N**, which will not show any such information. Note - the report will take significantly longer to process if Show Journal Diagnostics is set to **Y**.	N
Show Ledger Diagnostics (Show ledger records (and their journal detail) for buckets that are out of sync)	Optional field. If set to **Y** the report will produce diagnostics for the ledger records (and their summarized journal records) that should have been posted for an out of sync bucket under a separate heading of Ledger Detail. The total of these records appears in the Recorded on Ledger field of the report. Such detail will show for each journal record	N

Description (Caption)	Parameter Name	Default Value
	summarized into the ledger record matching the budget line, the journal record ID, the transaction ID down to the posting line number, and amount. The default is **N**, which will not show any such information. Note - the report will take significantly longer to process if Show Ledger Diagnostics is set to **Y**.	
Show Posting Line Diagnostics (Show Unjournalized posting line records for buckets that are out of sync)	Optional field. If set to **Y** the report will produce diagnostics for the not-yet-journalized posting line records that should have been posted for an out of sync bucket under a separate heading of Unjournalized Posting Line Detail. The total of these records appears in the Recorded on Posting Lines field of the report. Such detail will show the posting line ID, the transaction ID down to the posting line number, and the amount of the posting line. The default is **N**, which will not show any such information. Note - the report will take significantly longer to process if Show Posting Line Diagnostics is set to **Y**.	N
CBALDQ - Perform reports and out-of-sync and diagnostics	Report on CBALDQ indicates if reporting and out-of-sync condition checks should be performed for the table. If the entered value is a 'n' or 'N' then a check against that table is ignored; otherwise, reporting and out-of-sync check is performed against that table.	N
FBALDQ - Perform reports and out-of-sync and diagnostics	Report on FBALDQ indicates if reporting and out-of-sync condition checks should performed for the table. If the entered value is a 'n' or 'N' then a check against that table is ignored; otherwise, reporting and out-of-sync check is performed against that table.	N
BBAL - Perform reports and out-of-sync and diagnostics	Report on BBAL indicates if reporting and out-of-sync condition checks should be performed for the table. If the entered value is 'n' or 'N' then a check against that table is ignored; otherwise, reporting and out-of-sync check is performed against that table.	N
Progression Counter Size	Optional field. During processing, the job writes messages to the log to	10

Description (Caption)	Parameter Name	Default Value
	report on its progress based on the number of records already processed. When this parameter is specified, the value controls the interval at which these progression messages are written to the job log. A specified value should be a positive integer.	

Major Output

System Assurance 02 Report

Job Return code

The following table shows the potential job return codes for the System Assurance 02 Report job.

Return Code	Condition
Successful (1)	All of the records are processed successfully and no out of sync condition is reported.
Warning (4)	N/A
Non Fatal Error (8)	Out of sync condition encountered.
Failed (12)	The job will fail under the following conditions: • Required parameters are not entered • Entered parameters are invalid • Run time exceptions for unexpected situations
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

The report is ordered by Budget Structure ID, Budget Level ID, and Budget Name.

Selection Criteria

See parameters.

Problem Resolution

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

in the chain can be disabled before scheduling the new job. Alternatively, the stand alone report job can be run with the same set of parameters.

If the job ends with the return code of Non Fatal, then the out of sync condition needs to be investigated. Until the out of sync is investigated, it is advisable not to process any new transactions since that may impact the ability to complete the investigation. Run the SA02 chain again after resolving all of the issues. The out of sync condition can happen in the following scenarios:

- **BBALD, CBALDQ, or FBALDQ table balances are incorrect.** For CBALDQ and FBALDQ this could be due to changing the Cash and Fund Balance settings on the Posting Code table after processing transactions. If the Posting Code table is not the source of a CBALDQ or FBALDQ out of sync, then either a lack of update during transaction processing or database corruption might have caused the out of sync.
- For BBALD, the out of sync condition could arise either due to the lack of update during transaction processing or database corruption. Posting code setting scenarios won't arise in the case of BBALD since the BBAL table updates are not driven by the Posting codes.

  Historical balances are incorrect. If the BBALD, CBALDQ, or FBALDQ table balances are correct, then the out of sync condition is due to incorrect historical balances. If the out of sync is because of the historical balances, then the records that caused the out of sync condition must be identified (by comparing the transaction lines against the budget activity level records) and appropriate accounting entries need to be passed through the Journal Voucher (JV) to resolve the out of sync condition.

  Just as changes to the Posting Code table for CBAL and FBAL settings can make the tables not match the historical sources, changes to or the mistake of not selecting the Accounting Type Journal flag for any posting code that updates CBAL/FBAL or required a balance sheet with the Account Type setting, will cause the historical sources not to match the tables. When this occurs, the best course of action is to leave the Accounting Type Journal flag to unchecked and modify all accounting lines that use the posting code down to the Closed Amount of that line. Cancelling such lines is also a possibility. For journal vouchers, it will take a second transaction to back out the table updates. Then the Posting Code table should be corrected and the transactions modified back up or entered again. To change the Posting Code setting before corrective actions are taken will only compound the problem as there now will be some accounting lines that are correct and others that are not.

The "Out of Sync Scenarios" section later shows three sample scenarios of how to identify the out of sync conditions and how to put the application back in sync with SQL statements. Using a Journal Voucher with correction posting codes is also possible in each scenario when the volume is not great or an audit trail is desired. In all cases, updates should be done when there is no transaction processing being done in the application.

- **Exclude Unledgerized Records Mode**. If there are budget lines updated with transactions that are yet to be ledgerized and the job is run by setting the Exclude Unledgerized Records (EXCL_UNLEDGR_REC) parameter to N (No), such budget lines will show up as out of sync in the Report. In such cases, it is recommended to run the job in include unledgerized record mode on a more periodic basis to identify the true out of sync lines.

Some rare but possible reasons to investigate when it is not changes to the Posting Code table that caused the out of sync condition may be:

- Transactions that are final but balance tables were not updated
- Transactions not yet final but balance tables were updated
- Journal entries are incorrectly ledgerized
- Database corruption or manual manipulation

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

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Required Parameters are not entered. **Sample Message:** Ledger Name is required.	Schedule a new job after entering the Ledger Name.	
	Entered Parameters are not valid. **Sample Message:** The entered Ledger Name is not valid.	Schedule a new job after entering the correct Ledger Name.	
	Failed because of runtime exceptions for an unexpected situation.	The failure reason needs to be investigated before scheduling a new job	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling a new job.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before scheduling a new job.	

Step 2: Creating SA02 Report - This step will be performed only when the parameters are valid.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	No out of sync condition encountered.	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	Out of sync condition encountered.	Verify the out of sync condition and take the appropriate action to resolve the out of sync condition.	Refer to the "Out of Sync Scenarios" section for recommended steps for analysis, verification, and resolution.
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling a new job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues	The reason for the System Failure needs to be investigated before scheduling a new job.	N/A

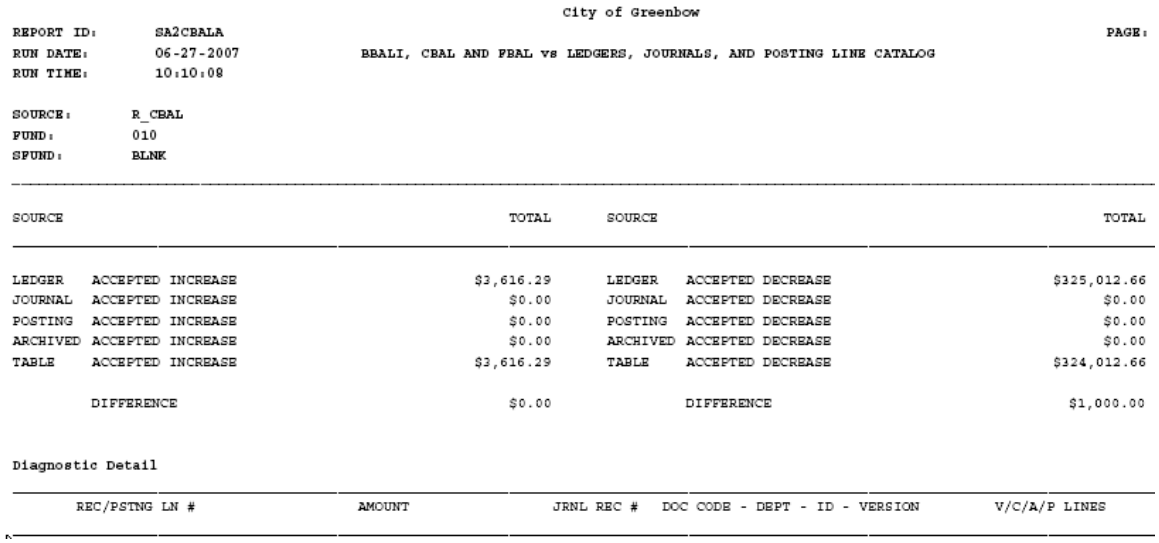
Out of Sync Scenarios

This section provides three samples of how to review the out of sync conditions for where the error was made with information on how to correct the error when it exists on Cash, Fund, or Balance Sheet Balance tables.

- Sample 1 – CBALDQ records are Out of Sync
- Sample 2 – FBALDQ records are Out of Sync
- Sample 3 – BBALD records are Out of Sync

This section is only to be used as a tool for how to start the investigation into an Out of Sync condition and should not be taken as the only steps and the only methods of correction. Certain circumstances can make the correction process more difficult. Methods are very similar between each of the three tables so after specific discussions of each are given, a common comparison rules section is presented.

Scenario 1 – CBALDQ records are Out of Sync



```
                                                City of Greenbow
REPORT ID:      SA2CBALA                                                                                    PAGE:
RUN DATE:       06-27-2007          BBALI, CBAL AND FBAL vs LEDGERS, JOURNALS, AND POSTING LINE CATALOG
RUN TIME:       10:10:08

SOURCE:       R_CBAL
FUND:         010
SFUND:        BLNK
```

SOURCE		TOTAL	SOURCE		TOTAL
LEDGER	ACCEPTED INCREASE	$3,616.29	LEDGER	ACCEPTED DECREASE	$325,012.66
JOURNAL	ACCEPTED INCREASE	$0.00	JOURNAL	ACCEPTED DECREASE	$0.00
POSTING	ACCEPTED INCREASE	$0.00	POSTING	ACCEPTED DECREASE	$0.00
ARCHIVED	ACCEPTED INCREASE	$0.00	ARCHIVED	ACCEPTED DECREASE	$0.00
TABLE	ACCEPTED INCREASE	$3,616.29	TABLE	ACCEPTED DECREASE	$324,012.66
	DIFFERENCE	$0.00		DIFFERENCE	$1,000.00

Diagnostic Detail

REC/PSTNG LN #	AMOUNT	JRNL REC #	DOC CODE - DEPT - ID - VERSION	V/C/A/P LINES

Here the out of sync (OOS) condition that exists for the Accepted Decrease amount only. The Difference amounts in the Increase and Decrease sections are calculated as follows:

Difference = Ledger + Journal + Posting Increase + Archived – Table

The following are a recommended set of steps to determine what system component(s) is causing the out of sync condition. It has almost always been the table that is wrong, but that should be verified before correcting the table to match what has been historically recorded in the ledger, journal, and posting line catalog. These steps are not the complete listing for every investigation, but are the main steps that should be done first before looking for the causes of why a component doesn't show the expected total. The 'why' step does not lend itself to run sheet documentation.

1. Determine the list of posting codes in the system that are set to update an OOS amount. Sample SQL results are shown below for baseline data. Posting Code table additions and modifications on site will make the results vary. Shown below is sample SQL for the Increase and Decrease columns respectively.

```
select * from R_PSCD where
CBAL_UPD_FL = 1 and
CBAL_BKT_IND = 2
```

Pscd Id	Pscd Nm
A025	Accrued Operating Transfer Out
D011	External Accrued Expenditure / Expense
D012	Accrued Bond Principle Expense
D021	Inventory Accrued Expenditure / Expense
D024	Internal Accrued Expenditure / Expense
P003	Pre Encumbrance
P005	Encumbrance
R007	Bad Debt Expense

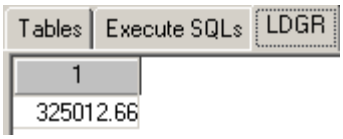
```
select * from R_PSCD where
CBAL_UPD_FL = 1 and
CBAL_BKT_IND = 1
```

Pscd Id	Pscd Nm
A026	Accrued Operating Transfer In
R002	External Billed Earned Revenue
R005	Billed NSF Revenue
R010	Earned Revenue
R021	Inventory Accrued Revenue
R023	Internal Accrued Revenue
R301	Expenditure Refund Holding Account

2. Query the SA2 input ledger to determine if the amount that is reported from the ledger on the SA2 Report is correct. The query should include the corresponding set of posting codes from the previous step, as well as the Fund and Sub Fund values that are listed on

the OOS line. Here is a sample SQL from the LDGR_FYDAD. If another ledger is input into the SA2 process, then it should be used.

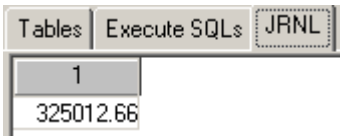
```
select sum (AM) from
LDGR_FYDAD where
PSTNG_CD_ID in (select
PSCD_ID from R_PSCD where
CBAL_UPD_FL = 1 and
CBAL_BKT_IND = 2) and
FUND_CD = '010' and
SFUND_CD is NULL
```



3. Now query the input ledger's source journal to determine if the amount that is reported from the ledger matches what is recorded on the journal. Just as the previous query did, this query should include the posting codes from Step #1, as well as the Fund and Sub Fund values that are listed on the OOS line. Even if the report does not list any amount for journal activity, this step should still be done to verify the ledger amount. Here is a sample SQL:

```
select sum (PSTNG_AM) from
JRNL_ACTG where PSTNG_CD_ID
in (select PSCD_ID from R_PSCD
where CBAL_UPD_FL= 1 and
CBAL_BKT_IND= 2) and FUND_CD
= '010' and SFUND_CD is NULL
```

Sample SQL Result:



4. At this point, the two queries have not revealed anything suspicious. The ledger query matches the Ledger Accepted Decrease amount reported by the SA2 Report and that matches the query for the same amount from the input journal. If the journal query resulted in a different amount from the ledger, that difference should be explained by the Journal Accepted Decrease amount.
5. Now query the Posting Line Catalog to determine if the amount that is reported from the journal query matches what is recorded on the catalog. Just as the previous query did, this query should include the posting codes from Step #1, as well as the Fund and Sub Fund values that are listed on the OOS line. Here is a sample SQL. The transaction phase code is a critical component to this query because the catalog contains draft and pending transactions that should not have updated the Cash Balance table.

```
select sum (PSTNG_AM) from
PSTNG_LN_CAT where PSTNG_CD_ID
in (select PSCD_ID from R_PSCD where
CBAL_UPD_FL= 1 and CBAL_BKT_IND=
2) and FUND_CD = '010' and SFUND_CD
is NULL and DOC_PHASE_CD in (3, 5)
```

Sample SQL Result:

6. At this point, all three queries have revealed the same amount in all three components: posting line catalog, journal, and ledger. Each is in sync with the other and the table is now the suspect component. If the posting line query resulted in a different amount from the journal, that difference should be explained by the Posting Accepted Decrease amount.If the same amount was not revealed in all 3 queries, please see the common comparison section after the BBAL report section.
7. Now that the table is found to be incorrect, a corrective action is needed. Database tools can be used to make table corrections, but the recommended approach is to use the Cash and Fund Balance Sync Process. The job and the related run sheet can be found in

the General Accounting / Batch Jobs folder. The primary reason for using the batch job is that it ensures that all tables are corrected: Cash Balance Detail (CBALDQ/R_CBAL), Cash Balance Summary (CBALSQ/R_CBAL_SMRY), and Cash Balance Pool (CBALPQ/R_CBAL_POOL).

An alternative approach for adjusting table amounts is a Journal Voucher transaction. That process starts with the creation of a new posting code that is setup to update CBAL in the Non Cash Decrease bucket, but is not indicated as one that goes to the Accounting Journal. The posting code should belong to the Closing Classification of 6 and have no value in the Account Type field. As a Journal Voucher will have to balance, another posting code should be created. This one should have no value in the Account Type field or in any of the CBAL fields. It should also be one that does not go to the Accounting Journal and should have a Closing Classification of 6. The reason for the classification of 6 is just to group the adjustment posting code with others that should not be selected for accounting reports.

Now a journal voucher would be created using both of the new posting codes. The one with the Non Cash Decrease setting would be debited for $1,000.00 for Fund 010 with no BSA used. A credit line to the other posting code will then be created for $1,000.00 to the same Fund with no BSA. After that journal voucher is submitted, the CBALDQ and CBALSQ tables will be updated as will the CBALPQ table, if that feature is used. The Journal Engine should be run if posting is not real time. Both posting lines will be marked as posted, but no journal records will actually be created because of the posting code settings. Now a new run of the report will show the table balance in sync with the historical activity. If the Non Cash Decrease amount on the table had been over of what was recorded in the historical sources, then the debit and credit would have been reversed. When correcting the Accepted Increases amount, a credit would increase the table amount and a debit would decrease it.

The advantage of the journal voucher approach is that an audit trail is created and the adjustment can happen during the day instead of at night when no activity is happening because the sync process requires no concurrent transaction processing. The drawback of the JV approach is when there are many records to be adjusted.

Sample 2 – FBALDQ records are Out of Sync

City of Greenbow

REPORT ID: SA2FBALO PAGE:
RUN DATE: 07-06-2007 BBALI, CBAL AND FBAL vs LEDGERS, JOURNALS, AND POSTING LINE CATALOG
RUN TIME: 16:17:51

SOURCE: R_FBAL
FUND: 100
SFUND: 01

SOURCE		TOTAL	SOURCE		TOTAL
LEDGER	EXPENSE ACCOUNT BALANCE	$3,220.50	LEDGER	REVENUE ACCOUNT BALANCE	$0.00
JOURNAL	EXPENSE ACCOUNT BALANCE	$0.00	JOURNAL	REVENUE ACCOUNT BALANCE	$0.00
POSTING	EXPENSE ACCOUNT BALANCE	$0.00	POSTING	REVENUE ACCOUNT BALANCE	$0.00
ARCHIVED	EXPENSE ACCOUNT BALANCE	$0.00	ARCHIVED	REVENUE ACCOUNT BALANCE	$0.00
TABLE	EXPENSE ACCOUNT BALANCE	$3,370.50	TABLE	REVENUE ACCOUNT BALANCE	$0.00
	DIFFERENCE	$150.00		DIFFERENCE	$0.00

Here the out of sync (OOS) condition that exists for the Expense Account Balance only. The Difference amounts in the Increase and Decrease sections are calculated as follows:

Difference = Ledger + Journal + Posting Increase + Archived – Table

The following are a recommended set of steps to determine what system component(s) is causing the out of sync condition. It has almost always been the table that is wrong, but that should be verified before correcting the table to match what has been historically recorded in the

ledger, journal, and posting line catalog. These steps are not the complete listing for every investigation, but are the main steps that should be done first before looking for the causes of why a component doesn't show the expected total. The 'why' step does not lend itself to run sheet documentation.

1. Determine the list of posting codes in the system that are set to update an OOS amount. Sample SQL results are shown below for baseline data. Posting Code table additions and modifications on site will make the results vary. Shown below is sample SQL for the Expense and Revenue columns respectively.

```
select * from R_PSCD where
FBAL_UPD_FL= 1 and
FBAL_BKT_IND= 2
```

```
select * from R_PSCD where
FBAL_UPD_FL= 1 and
FBAL_BKT_IND= 1
```

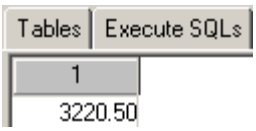
Sample SQL Result:

Pscd Id	Pscd Nm
A017	Operating Transfer Out
A025	Accrued Operating Transfer Out
A100	Accrued Expenditure Close
A101	Cash Expenditure Close
A104	Pre Encumbrance Close
A105	Encumbrance Close
D011	External Accrued Expenditure / Expense
D012	Accrued Bond Principle Expense
D013	Cash Expenditure/Expense - Warrants
D014	External Cash Expenditure/Expense
D015	Cash Bond Principle Expenditure/Expense
D021	Inventory Accrued Expenditure / Expense
D022	Inventory Cash Expenditure/Expense
D023	Cost of Goods Sold
D024	Internal Accrued Expenditure / Expense
D025	Internal Cash Expenditure/Expense
F020	Depreciation Expense
F021	Gain/Loss Expense from Sale of Fixed Asset
F022	Internal Expense from Fixed Asset Exchange
S005	Inventory Adjustment Expense
P003	Pre Encumbrance
P005	Encumbrance
R007	Bad Debt Expense

Pscd Id	Pscd Nm
R010	Earned Revenue
R021	Inventory Accrued Revenue
R022	Inventory Collected Revenue
A004	Bond Revenue
A018	Operating Transfer In
A026	Accrued Operating Transfer In
A102	Billed Revenue Close
A103	Collected Revenue Close
F030	Sale of Fixed Asset Revenue
F031	Gain/Loss Revenue from Sale of Fixed Asset
F032	Internal Revenue from Fixed Asset Exchange
R023	Internal Accrued Revenue
R024	Internal Collected Revenue
R102	Collected Unearned Revenue
I006	Investment Interest Revenue
R002	External Billed Earned Revenue
R003	External Collected Earned Revenue
R005	Billed NSF Revenue
R006	Collected NSF Revenue

1. Query the SA2 input ledger to determine if the amount that is reported from the ledger on the SA2 Report is correct. The query should include the posting codes from Step #1, as well as the Fund and Sub Fund values that are listed on the OOS line.
2. Here is a sample SQL from the LDGR_FYDAD. If another ledger is input into the SA2 process, then it should be used.

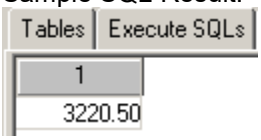
```
select sum(AM) from LDGR_FYDAD
where PSTNG_CD_ID in (select
PSCD_ID from R_PSCD where
FBAL_UPD_FL = 1 and
FBAL_BKT_IND = 2) and FUND_CD =
'100' and SFUND_CD = '01'
```



3. Now query the SA2 input ledger's source journal to determine if the amount that is reported from the ledger matches what is recorded on the journal. Just as the previous query did, this query should include the posting codes from Step #1, as well as the Fund and Sub Fund values that are listed on the OOS line. Even if the report does not list any amount for journal activity, this step should still be done to verify the ledger amount. Here is a sample SQL:

```
select sum(PSTNG_AM) from
JRNL_ACTG where PSTNG_CD_ID
in (select PSCD_ID from R_PSCD
where FBAL_UPD_FL = 1 and
FBAL_BKT_IND = 2) and FUND_CD
= '100' and SFUND_CD = '01'
```

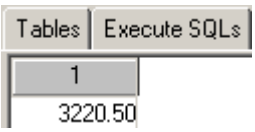
Sample SQL Result:



4. At this point, the two queries have not revealed anything suspicious. The ledger query matches the Ledger Expense Account Balance amount reported by the SA2 Report and that matches the query for the same amount from the input journal. If the journal query resulted in a different amount from the ledger, that difference should be explained by the Journal Expense Account Balance.
5. Now query the Posting Line Catalog to determine if the amount that is reported from the journal query matches what is recorded on the catalog. Just as the previous query did, this query should include the posting codes from Step #1, as well as the Fund and Sub Fund values that are listed on the OOS line. Here is a sample SQL. The transaction phase code is a critical component to this query because the catalog contains draft and pending transactions that should not have updated the Cash Balance table.

```
select sum (PSTNG_AM) from
PSTNG_LN_CAT where
PSTNG_CD_ID in (select PSCD_ID
from R_PSCD where FBAL_UPD_FL=
1 and FBAL_BKT_IND= 2) and
FUND_CD = '100' and SFUND_CD =
'01' and DOC_PHASE_CD in (3, 5)
```

Sample SQL Result:



6. At this point, all three queries have revealed the same amount in all three components: posting line catalog, journal, and ledger. Each is in sync with the other and the table is now the suspect component. If the posting line query resulted in a different amount from the journal, that difference should be explained by the Posting Accepted Decrease amount. If the same amount was not revealed in all 3 queries, please see the common comparison section after the BBAL report section.
7. Now that the table is found to be incorrect, a corrective action is needed. Database tools can be used to make table corrections, but the recommended approach is to use the Cash and Fund Balance Sync Process. The job and the related run sheet can be found in the General Accounting / Batch Jobs folder. The primary reason for using the batch job is that it ensures that both tables are corrected: Fund Balance Detail (FBALDQ/R_FBAL) and Fund Balance Summary (FBALSQ/R_FBAL).

An alternative approach for adjusting table amounts is a Journal Voucher transaction. That process starts with the creation of a new posting code that is setup to update FBAL in the Expense bucket, but is not indicated as one that goes to the Accounting Journal. The posting code should belong to the Closing Classification of 6 and have no value in the Account Type field. As a Journal Voucher will have to balance, another posting code should be created. This one

should have no value in the Account Type field or in any of the FBAL fields. It should also be one that does not go to the Accounting Journal and should have a Closing Classification of 6.

Now a journal voucher would be created using both of the new posting codes. The one with the Expense setting would be credited for $150.00 for Fund 100 with no BSA used. A debit line to the other posting code will then be created for $150.00 to the same Fund with no BSA. After that journal voucher is submitted, the FBALDQ and FBALSQ tables will be updated. The Journal Engine should be run if posting is not real time. Both posting lines will be marked as posted, but no journal records will actually be created because of the posting code settings. Now a new run of the report will show the table balance in sync with the historical activity. If the Expense amount on the table had been short of what was recorded in the historical sources, then the debit and credit would have been reversed. When correcting the Revenue amount, a credit would increase the table amount and a debit would decrease it.

The advantage of the journal voucher approach is that an audit trail is created and the adjustment can happen during the day instead of at night when no activity is happening because the sync process requires no concurrent transaction processing.

Sample 3 – BBALD records are Out of Sync

City of Greenbow

REPORT ID: 0 PAGE:
RUN DATE: 07-10-2007 BBALI, CBAL AND FBAL vs LEDGERS, JOURNALS, AND POSTING LINE CATALOG
RUN TIME: 10:42:30

SOURCE: R_BBAL_ITD
FUND: 621
SFUND: BLNK
BSA: 1130
SBSA: 001

SOURCE		TOTAL
LEDGER	BALANCE	$1,804.00
JOURNAL	BALANCE	$0.00
POSTING	BALANCE	$0.00
ARCHIVED	BALANCE	$0.00
TABLE	BALANCE	$1,724.00
	DIFFERENCE	$80.00

Difference is calculated as follows:

Difference = Ledger + Journal + Posting Increase + Archived – Table

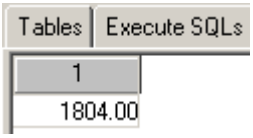
The following are a recommended set of steps to determine what system component(s) is causing the out of sync condition. It has almost always been the table that is wrong, but that should be verified before correcting the table to match what has been historically recorded in the ledger, journal, and posting line catalog. These steps are not the complete listing for every investigation, but are the main steps that should be done first before looking for the causes of why a component doesn't show the expected total. The 'why' step does not lend itself to run sheet documentation.

There is not the determination of a list of posting codes in the system for BBAL. Updates to ITD Balance Sheet Detail (BBAL/R_BBAL), ITD Balance Sheet Summary (BBALS/R_BBAL_SMRY), and FY Balance Sheet Detail (BBALFY/R_BBAL_FY_DETAILS) are made when a posting line contains a balance sheet or offset balance sheet account. Posting codes only go as far as requiring an account be entered/defaulted and various validations for the account.

1. Query the SA2 input ledger to determine if the amount that is reported from the ledger on the SA2 Report is correct. The query should include the Fund, Sub Fund, BSA, and Sub BSA values that are listed on the OOS line.

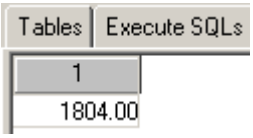
   Here is a sample SQL from the LDGR_FYDAD. If another ledger is input into the SA2 process, then it should be used. Also, as sub elements are not always used, if the OOS line reports BLNK as a sub value, the query should be modified to remove the { ='value' } and replace it with {is null}.

```
select sum (AM) from
LDGR_FYDAD where FUND_CD =
'621' and SFUND_CD is null and
BSA_CD = '1130' and SBSA_CD =
'001'
```



2. Now query the SA2 input ledger's source journal to determine if the amount that is reported from the ledger matches what is recorded on the journal. Just as the previous query did, this query should include the Fund, Sub Fund, BSA, and Sub BSA that are listed on the OOS line. Even if the report does not list any amount for journal activity, this step should still be done to verify the ledger amount. Here is a sample SQL:

```
select sum (PSTNG_AM) from
JRNL_ACTG where FUND_CD =
'621' and SFUND_CD is null and
BSA_CD = '1130' and SBSA_CD
= '001'
```



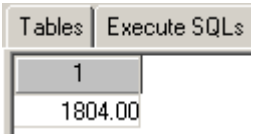
3. At this point, the two queries have not revealed anything suspicious. The ledger query matches the Ledger Balance amount reported by the SA2 Report and that matches the query for the same amount from the input journal. If the journal query resulted in a different amount from the ledger, that difference should be explained by the Journal Balance.
4. Now query the Posting Line Catalog to determine if the amount that is reported from the journal query matches what is recorded on the catalog. Just as the previous query did, this query should include the posting codes from Step #1, as well as the Fund and Sub Fund values that are listed on the OOS line.

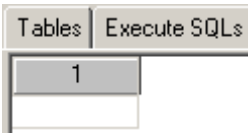
Here is the sample SQL for accounts used in the BSA as well as the OBSA posting line fields. The transaction phase code is a critical component to this query because the catalog contains draft and pending transactions that should not have updated the Cash Balance table. Both queries should be run and added if totals are found for each. When comparing posting line totals to journal totals, keep in mind that posting line queries may produce a total of the opposite sign than those from a journal.

```
select sum (PSTNG_AM) from PSTNG_LN_CAT where FUND_CD = '621' and SFUND_CD
is null and OBSA_CD = '1130' and OSBSA_CD = '001' and DOC_PHASE_CD in (3, 5)
```

```
select sum (PSTNG_AM) from PSTNG_LN_CAT where FUND_CD = '621' and SFUND_CD
is null and BSA_CD = '1130' and SBSA_CD = '001' and DOC_PHASE_CD in (3, 5)
```

Sample SQL Results:





5. At this point, all three queries have revealed the same amount in all three components: posting line catalog, journal, and ledger. Each is in sync with the other and the table is now the suspect component. If the posting line query resulted in a different amount from the journal, that difference should be explained by the Posting Balance. If the same amount was not revealed in all 3 queries, please see the common comparison section after this BBAL report section.
6. Now that the table is found to be incorrect, a corrective action is needed. Database tools can be used to make table corrections, but the recommended approach is to use the Rebuild BBAL process. The job and the related run sheet can be found in the General Accounting / Batch Jobs folder. The primary reason for using the batch job is that it ensures that all related tables are corrected: ITD Balance Sheet Detail (BBLD/R_BBAL), ITD Balance Sheet Summary (BBALS/R_BBAL_SMRY), FY Balance Sheet Detail (BBALFY/R_BBAL_FY_DETAILS), all 3 cash balance tables, both fund balance tables, and others not directly viewed: R_BSA_CBAL, R_BSA_FBAL, R_BSA_CBAL_AUD.

When the tables are not correct, but the historical sources are, there is an alternative approach – a Journal Voucher. That process starts with the creation of a new posting code that is setup just like the one that is incorrectly recorded in the journal and ledger, but this new one is not marked to go the Accounting Journal. As a Journal Voucher will have to balance, another posting code should be created. This second posting code should have no value in the Account Type field and should not be marked to go to the Accounting Journal. Both should have a Closing Classification of 6.

Now a journal voucher would be created using both of the new posting codes. The first posting code would be debited for $80.00 using the Fund, Sub Fund, BSA, and Sub BSA of the OOS record. A credit line to the other posting code will then be created for $80.00 with only the Fund and Sub Fund. After that journal voucher is submitted, the BBALD, BBALS, and BBALFY tables will be updated. The Journal Engine should be run if posting is not real time. The Journal Engine will mark both posting lines as posted, although no journal records will actually be created because of the posting code settings. Now a new run of the report will show the table balance in sync with the historical activity. The following table depicts whether the first posting code used with the BSA and Sub BSA should be debited or credited based on the account type and the relationship of the table balance with the historical balance:

Account Type	Table to Historical	Debit or Credit
Asset	Table short	Debit

Asset	Table over	Credit
All others	Table short	Credit
All others	Table over	Debit

When the BBAL tables are correct, but the historical sources are not, journal voucher updates are **never** an option. Unlike CBAL and FBAL, a posting code cannot be created that will update the Accounting Journal but not update the BBAL tables. Updates to the BBAL tables happen anytime a balance sheet exists on a posting line.

Common Review Results

Below is a table of several of the more common diagnostic outcomes with a list of possible events that caused the outcome, ordered from the more likely to the least likely. A problem with SA02 reporting logic is the implied last possibility

When the tables are correct, but the historical sources are not, then other solutions are needed that are problem-specific and beyond the scope of this run sheet. Other assurance reports are designed to find out-of-sync conditions in these areas. Many such conditions have to be addressed before or in conjunction with assurance 02.

- Posting lines – Systems Assurance 06 and 16
- Journal – Systems Assurance 03, 07, and 08
- Ledger – Systems Assurance 03, 07, and 08

Diagnostic Result	Possibilities
• Reported amount from ledger does not equal the ledger query	• Incremental run done since last full run so some amount of ledger query is in the journal report amount. • Ledger Engine has run since report was generated. • A full run with the Exclude Unledgerized Records (EXCL_UNLEDGR_REC) parameter set to N (No) will remove discrepancies and make review easier.
• Amount reported from journal when all journal records were thought to be ledgerized • Reported amount from journal query does not equal the ledger query amount • Reported amount from journal query does not equal the reported ledger amount plus reported journal amount	• A transaction has processed against the account since the report was created • Unit of work for process id LDGRPOST is marked as failed on Journal Log table • A gap logged for process id LDGRPOST now has journal records All activity against an account has to be stopped during the analysis phase. In activating a Fund, Sub Fund, BSA, or Sub BSA is one alternative to allow research during the online day. Before the next SA run, the Ledger Engine should be run. Once against failed work and again against gaps. Analysis can still be performed in this situation even if #1 happens as long as an updated table amount is acquired.

• Amount reported from posting line catalog when all posting lines were thought to be journalized	• A transaction has processed against the account since the report was created where real time journal posting is not activated or the transaction posts asynchronously • Posting line that should be journalized has not been All activity against an account has to be stopped during the analysis phase. In activating a Fund, Sub Fund, BSA, or Sub BSA is one alternative to allow research during the online day. Before the next SA run, the Journal Posting Initiator and the Journal Engine should be run. Analysis can still be performed in this situation even if #1 happens as long as an updated table amount is acquired.
• Amount from posting line catalog query different than that of journal query	• Posting line has been marked as journalized but wasn't Please see the following queries that can be used to reveal such a posting line.

If the query from the Posting Line Catalog returned a different amount than the journal query and that difference was not equal to the reported posting amount, the following queries can be used with a replacement of place-holding x values being removed and replaced by appropriate values or if not used with the statement: is null.

BBAL

```
SELECT A.DOC_CD, A.DOC_ID, A.PSTNG_CD_ID, A.PSTNG_AM, A.FUND_CD,
A.SFUND_CD, A.DOC_PHASE_CD, A.JRNL_PSTNG_IND, A.BUD_PSTNG_IND,
A.DOC_LAST_DT
FROM PSTNG_LN_CAT A WHERE A.DOC_PHASE_CD IN (3, 5) AND
A.JRNL_PSTNG_IND = 3 AND A.FUND_CD = 'xxx' AND A.SFUND_CD is null AND
PSTNG_CD_ID not in (SELECT PSCD_ID from R_PSCD where ACTG_TYP_JRNL_FL = 0)
and NOT EXISTS
(SELECT B.DOC_CD FROM JRNL_ACTG B WHERE A.DOC_CD = B.DOC_CD AND
A.DOC_DEPT_CD = B.DOC_DEPT_CD AND A.DOC_ID = B.DOC_ID
AND A.DOC_VERS_NO = B.DOC_VERS_NO AND A.DOC_VEND_LN_NO =
B.DOC_VEND_LN_NO AND A.DOC_COMM_LN_NO = B.DOC_COMM_LN_NO
AND A.DOC_ACTG_LN_NO = B.DOC_ACTG_LN_NO AND A.DOC_PSTNG_LN_NO =
B.DOC_PSTNG_LN_NO)
```

CBAL

```
select A.DOC_CD, A.DOC_ID, A.PSTNG_CD_ID, A.PSTNG_AM, A.FUND_CD,
A.SFUND_CD, A.DOC_PHASE_CD, A.JRNL_PSTNG_IND, A.BUD_PSTNG_IND,
A.DOC_LAST_DT from PSTNG_LN_CAT A where A.DOC_PHASE_CD in (3, 5) and
A.JRNL_PSTNG_IND = 3 and FUND_CD = 'xxx' and SFUND_CD = 'xxx' and
A.PSTNG_CD_ID in (select PSCD_ID from R_PSCD where ACTG_TYP_JRNL_FL= 1) and
not exists (select B.DOC_CD from JRNL_ACTG B where A.DOC_CD = B.DOC_CD and
A.DOC_DEPT_CD = B.DOC_DEPT_CD and A.DOC_ID = B.DOC_ID and
A.DOC_VERS_NO = B.DOC_VERS_NO and A.DOC_VEND_LN_NO =
B.DOC_VEND_LN_NO and A.DOC_COMM_LN_NO = B.DOC_COMM_LN_NO and
```

```
A.DOC_ACTG_LN_NO = B.DOC_ACTG_LN_NO and A.DOC_PSTNG_LN_NO =
B.DOC_PSTNG_LN_NO)
```

FBAL

```
select A.DOC_CD, A.DOC_ID, A.PSTNG_CD_ID, A.PSTNG_AM, A.FUND_CD,
A.SFUND_CD, A.DOC_PHASE_CD, A.JRNL_PSTNG_IND, A.BUD_PSTNG_IND,
A.DOC_LAST_DT from PSTNG_LN_CAT A where A.DOC_PHASE_CD in (3, 5) and
A.JRNL_PSTNG_IND = 3 and FUND_CD = 'xxx' and SFUND_CD = 'xxx' and
A.PSTNG_CD_ID in (select PSCD_ID from R_PSCD where FBAL_UPD_FL= 1 and
FBAL_BKT_IND= x) and not exists (select B.DOC_CD from JRNL_ACTG B where
A.DOC_CD = B.DOC_CD and A.DOC_DEPT_CD = B.DOC_DEPT_CD and A.DOC_ID =
B.DOC_ID and A.DOC_VERS_NO = B.DOC_VERS_NO and A.DOC_VEND_LN_NO =
B.DOC_VEND_LN_NO and A.DOC_COMM_LN_NO = B.DOC_COMM_LN_NO and
A.DOC_ACTG_LN_NO = B.DOC_ACTG_LN_NO and A.DOC_PSTNG_LN_NO =
B.DOC_PSTNG_LN_NO)
```

2.1.3 Systems Assurance 03 – Debits Equal Credits

When to Run

On-demand or as part of the nightly systems assurance process against one journal and one ledger is the recommended approach for nightly or other periodic runs.

Description

The job takes a journal or ledger as input and then determines if that file equals in terms of debits and credits. The job can be run incrementally where the next record after the ending record number from the last run is the starting point or in the complete mode where all records are read. The job is also enhanced with the ability to perform the assurance based on a list of included transaction codes, or on all transaction codes except those listed in an exclude list. The following are examples of data stores that may be utilized by this systems assurance process:

Journals (4)

- Accounting Journal s
- Budget Fiscal Year Not Equal to Fiscal Year Journal
- Internal Journal
- Fixed Asset Accounting Journal

Ledgers (10+)

- Accounting Ledger (APD)
- Accounting Ledger (FY)
- Accounting Ledger (BFY)
- Accounting Ledger (ITD)
- Ledger SA Budgets
- Full Detail Accounting Ledger
- Any of the generic ledgers supplied (L001 to L010)

This job has eight parameters discussed in detail later.

The first output is a report is created to show those records that do not balance. The first page shows the input parameters for the job. After that, subsequent pages show records found to not balance sorted by the same fields defined in the summarization level parameter. When the summarization level is 2, for each record listed for a journal, the transaction ID (transaction code, dept, ID, version) is listed along with the fund, fiscal year, accounting period, and the total amount of debits and credits from that transaction. The last column is an error message column which reads ‘Unequal Debits and Credits.” A summary line that counts how many transactions do not balance then follows all lines. When the summarization level is 1 for a journal, the transaction ID detail is removed and records are just presented in terms of the fund, fiscal year, and accounting period with total debits and credits. This type of output is the same when a ledger is reviewed by the job.

The second output is a record to the **Journal Log** table showing the parameters for which the job as run.

Multiple instances of this report are permitted. When running in full mode, multiple instances are allowed given different sources are being assured. Running incrementally against different

sources is also permitted. However, simultaneous incremental runs against the same source will produce undesired results.

This systems assurance job should not be run against any journal or ledger that does not contain balanced entries. All ledgers built from the Cost Accounting Journal will contain only 1-sided entries if accounting events are done for Charges, Back End Splits, and Revenue Credits. If any ledgers are built from the Cash or 1099 Journals, they will contain only 1-sided entries because of how each of those journals is posted. Running the SA3 job against any of these will still assure the ledger; however, users will have to determine which of the results that would normally indicate a problem are truly not a problem.

Major Input

- Any single journal or ledger.
- JRNL_LOG (Journal Log)

Output

- JRNL_LOG (Journal Log) updated
- System Assurance 3 Report

Sort Criteria

The report is ordered by Fund, Fiscal Year and Accounting Period when available.

Parameters

Batch Parameters

Description (Caption)	Parameter Name	Default Value
Client name appearing on report	Client Name	
Comma-delimited list of Transaction Codes for which to run report. Leave blank to include all Transaction Codes.	Include Transaction Codes	
Comma-delimited list of Fiscal Years for which to run report. Leave blank to run for all Fiscal Years.	Fiscal Years	
Comma-delimited list of Transaction Codes to exclude from report. Leave blank to exclude no Transaction Codes.	Exclude Transaction Codes	
Y to only process journal records posted since last report. N to produce full report. Y not allowed if data source is a ledger.	Incremental	N
Journal or ledger source table.	Journal/Ledger Name	JRNL_ACTG
Comma-delimited list of Accounting Periods for which run report. Leave blank for all Accounting Periods.	Periods	

Description (Caption)	Parameter Name	Default Value
1 = Fund, FY, Accounting Period. 2 = Fund, FY, Accounting Period, and Transaction.	Summary Level	1
Maximum Pre-fetch Count used to control the number of ledger records selected.	MAX_PREFETCH_COUNT	

- Client Name to appear on the report. The name appears as the first line in the header of the report. This is an optional field with no default. The system wide default for Client Name will be used when established.
- Include Transaction Codes is an optional comma-delimited list of transactions codes for which the report should be executed. If blank, report is executed for all transaction codes (except for those listed in Exclude Transaction Codes, if any). Otherwise, value should be a comma-delimited list of transaction codes, for example, "PO,DO,RQA,RQS,RQM,SC,PRC". There is no default nor any editing of this field for valid transaction codes.
- Fiscal Years is an optional comma-delimited list of four-digit fiscal years for which the report should be executed. If blank, report is executed for all fiscal years. Otherwise, value should be comma-delimited list of years, for example, "1999,2000,2001,2002". There is no default nor any editing of values for this field.
- Exclude Transaction Codes is an optional comma-delimited list of transactions codes to be excluded from the report. If blank, no transaction codes are excluded from the report (unless Include Transaction Codes is non-blank providing a list of specific transaction codes). Otherwise, value should be a comma-delimited list of transaction codes, for example, "RQA,RQS,RQM". There is no default nor any editing for valid transaction codes.
- Incremental may be set to **Y** to indicate the source journal/ledger should be checked for Debits Equal Credits from the point at which the last incremental execution ended. If set to **N** (for No), the report will be performed against the entire source journal/ledger. If set to **Y** (for Yes), the report will be performed starting at the next record after the end of the prior execution (as recorded on JRNL_LOG) against the same journal/ledger. This parameter only applies to journals. Any setting than **N** when a ledger is being analyzed causes the job to fail, writing an error to the log. The default value for this field is **N** for a full analysis
- Journal/Ledger Name is the data object name of the table to use as the input journal or ledger. The default is the JRNL_ACTG (accounting journal) and any other value specified must be entered in the same 'all capitals' format. An invalid value entered will cause the job to fail, issuing an error to the log.
- Periods is an optional comma-delimited list of fiscal periods for which the report should be executed. If blank, report is executed for all fiscal periods. Otherwise, value should be a comma-delimited list of fiscal periods, for example, "1,2,3,4,10,11,12". There is no default nor any editing of values for this field.
- Summary Level may be set to 1 or 2. When 1, the source journal/ledger records are summarized by Fiscal Year, Fiscal Period, and Fund Code. When 2, the source journal/ledger records are summarized by Fiscal Year, Fiscal Period, Fund Code, Transaction Code, Transaction Department Code, Transaction Identifier, and Transaction Version Number.
- MAX_PREFETCH_COUNT defines the number of records saved into memory after a query has been performed. This is an optional field with a default value of 1000.

Sort Criteria

- If Summary Level is 1, the report is sorted by Fiscal Year, Fiscal Period, and Fund Code.
- If Summary Level is 2, the report is sorted by Transaction Code, Transaction Department Code, Transaction Identifier, Transaction Version Number, Fiscal Year, Fiscal Period, and Fund Code

Selection Criteria

See parameters.

Problem Resolution

- If the job fails, no restore is required. Rerun the job.
- When run in incremental mode, an out-of-sync condition found in 1 run will not be listed in a subsequent run. Please do not assume the ledger is now correct. A full run against such a ledger will reveal the out-of-sync condition still exists.
- When a ledger is found to be out of sync because one or more records are missing, the appropriate measure to take is to rebuild the ledger by scheduling the Rebuild Ledger batch job found in the Postings folder. Please see the Rebuild Ledger run sheet in the *CGI Advantage Financial – General Accounting Run Sheets* guide for more details.
- If the following message is issued in the job log, 'No New Activity Found,' it indicates that an incremental run has been performed against a journal were no new records have been created since the last run.

2.1.4 Systems Assurance 04 – The Accounting Lines Requesting Payment vs. Disbursement Request

When to Run

On-demand or as part of the nightly systems assurance process as a compliment to the Systems Assurance 05 job.

Description

This process ensures the integrity of all outstanding requests for payment on the Disbursement Request DISRQ page, comparing each record with the respective Payment Request record, Payroll record, Accounting Based Spending record and Travel record with an event type marked to update Disbursement Request. Comparisons ensure that records are not missing from the DISRQ for open accounting lines as the inquiry is the source of data for the Automatic Disbursement (AD) Chain. Other comparisons use the line amount (LN_AM) and closed line amount (AL_CLSD_AM) fields to confirm that all items on the DISRQ are correct.

The first output is a report showing any problems between the DISRQ and the transaction components. The report lists items in two sections. The first includes those accounting lines that do not match a corresponding DISRQ record. Columns exist to show the line amount and closed line amount from the transaction components as well as the amount on DISRQ. Another column exists to detail what is wrong with each line. The second section lists those accounting lines that do not have a corresponding DISRQ record.

The second output is a record to the **Journal Log** table showing job parameters and listing the count of records written to the report as 'Total number of transactions reviewed'. This count is not the number of DISRQ records reviewed.

Major Input

The Data objects, which are the input to the process

ABS Accounting Line	ABS_DOC_ACTG
PR Accounting Line	PR_DOC_ACTG
PYRL Accounting Line	PYRL_DOC_ACTG
TRVL Accounting Line	TRVL_DOC_ACTG
Disbursement Request	R_AP_DISB_RQST

Output

- Disbursement Request Out of Sync Listing report
- JRNL_LOG - (Journal Log) update

Parameters

Batch Parameters

Parameter	Description	Default Value
Client Name	Optional parameter for a name appearing on the report.	(none)

(CLIENT_NM)		
Commit Block Size (COMMIT_BLOCK)	A performance parameter used to control recording information for the report.	1000
Select ss (SELECT_BLOCK)	A performance parameter used to control record selection from the 5 tables.	1000
Earliest Date Range	An optional parameter used to reduce the range of DISRQ records selected to those where the Transaction Last Modified On date (DOC_LAST_DT) is equal to or greater than a specified date (mm-dd-yyyy).	(none)
Latest Date Range	An optional parameter used to reduce the range of DISRQ records selected to those where the Transaction Last Modified On date (DOC_LAST_DT) is equal to or less than a specified date (mm-dd-yyyy).	(none)

Sort Criteria

Transaction Code, Department, ID, Version Number and Line Numbers (internal line numbers)

Selection Criteria

Select all transactions with Transaction Last Date within the range of Earliest and Latest Date Range.

- If Earliest Date was left blank, then select records with a Transaction Last Date < Latest Date.
- If Latest Date was left blank, then select records with a Transaction Last Date > Earliest Date.
- If both Earliest and Latest Date were left blank, then select records with a Transaction Last Date < the Current System Date.
- If both Earliest and Latest Dates were not blank then select records with a Transaction Last Date >= Earliest Date and < Latest Date.

Problem Resolution

When an accounting line is not in sync with the disbursement request record, but still open, a cancellation of the transaction with that line should be done (if allowed) to effectively end involvement of each record with disbursements. Then a new transaction should be entered. If the transaction cannot be cancelled, that accounting line should be modified down to $0.00 and closed. When transaction modifications or cancellations are not available or do not fix the out of sync condition, database updates with a tool such as SQL should be considered as a last resort.

2.1.5 Systems Assurance 05 – Disbursement Request (DISRQ) Table vs. ABS, PR, and TRVL Transaction Accounting Lines

When to Run

On-demand or as part of the nightly systems assurance process

Description

This process ensures the integrity of all records on the Disbursement Request (R_AP_DISB_RQST) by comparing its records with the records on the PR Accounting Line Catalog (PR_DOC_ACTG), ABS Accounting Line Catalog (ABS_DOC_ACTG), and Travel Accounting Line Catalog (TRVL_DOC_ACTG).

The job takes the **Disbursement Request Table** (DISRQ) and compares the records it finds there to the accounting line components for the Payment Request (PR/PRC), Accounting Based Spending (ABS/GAX), and Travel (TRVL) transactions to verify that records are not missing from the accounting line components that are on the DRT and that information there corresponds to what is on those two transaction components. The items checked for consistency are the line amount (LN_AM) and the closed line amount (AL_CLSD_AM). These two amounts confirm that items on the DRT are unpaid as they should be. Also, the job checks to ensure that no partial payments have been made towards an accounting line.

This job takes three parameters discussed in detail later.

The first output is a report is produced showing, by vendor code, any problems between the DRT and the transaction components. Under each vendor code is a listing of transaction ID's down to the accounting line number that are not reflected the same way on the DRT. These records are sorted by transaction code, dept, ID, version number, and accounting line number. Columns exist to show the line amount and closed line amount from the transaction components. Another column exists to detail what is wrong with each line. Valid error messages in this column are: ABS_DOC_ACTG Entry Not Found for Disb RQST, PR_DOC_ACTG Entry Not Found for Disb RQST, Closed ACTG Line on Disbursement Request, Partial ACTG Line on Disbursement Request.

The second output is a record to the **Journal Log** (JLOG) table showing the parameters for which the job was run.

Major Input

The Data objects, which are the input to the process

ABS Accounting Line	ABS_DOC_ACTG
PR Accounting Line	PR_DOC_ACTG
TRVL Accounting Line	TRVL_DOC_ACTG
Disbursement Request	R_AP_DISB_RQST

Output

- Disbursement Request Out of Sync Listing report
- JRNL_LOG - (Journal Log) update

Parameters

Batch Parameters

Description (Caption)	Parameter Name	Default Value
Client Name appearing on report	Client Name	
Earliest Date Range (mm-dd-yyyy)	Earliest Date Range	
Latest Date Range (mm-dd-yyyy)	Latest Date Range	

- Client Name to appear on the report. The name appears as the first line in the header of the report. This is an optional field with no default. The system wide default for Client Name will be used when established.
- Optional Earliest and Latest Date Range Fields exist to specify a specific range of transaction last dates (DOC_LAST_DT) to check. When an accounting line is found with a Run Time Date greater than the Earliest Date, it is selected. When an accounting line is found with a Run Time Date less than the Latest Date, it is selected. No edit exists to prevent the Latest Date from being prior to the Earliest Date because the dates are not formatted as such. Edits do exist to prevent invalid dates from being used. When a date is left blank, that end of the range is open to the earliest or latest record. When dates are specified, the system defaults the time associated with them to 00:00:00. For this reason, if only one day is desired for assurance, that date should be entered in the Earliest Date field and the next day should be entered in the Latest Date field. If assurance is desired up through a certain date, then the next day after that one should be entered in the Latest Date field.

Sort Criteria

Transaction Code, Department, ID, Version Number and Line Numbers (internally)

Selection Criteria

Select all transactions with a Run Time Date on Disbursement Request within the range of Earliest and Latest Date Range.

- If Earliest Date was left blank then select records with a Run Time Date < Latest Date.
- If Latest Date was left blank then select records with a Run Time Date > Earliest Date.
- If both Earliest and Latest Date were left blank then select records with a Run Time Date < Current System Date.
- If both Earliest and Latest Dates were not blank then select records with a Run Time Date >= Earliest Date and < Latest Date.

Problem Resolution

No database restore is required. Rerun the job.

2.1.6 Systems Assurance 06 – Transaction Component Levels In-sync

Job Name	Systems Assurance 06 – Transaction Component Levels In-sync
Recommended Frequency	Weekly. Can be run On Demand for a small set of records by entering the transaction(s) code as a job parameter.
Parallel processing enabled	Yes. Can be run in parallel for different transaction codes.
Can the job be restarted?	No
Exception reports generated	Yes, the job produces a report when the out of sync condition is encountered.

Overview

This Systems Assurance 06 (SA6) job ensures that certain common attributes across the transaction components are in sync. These include Transaction Function, Transaction Phase and Transaction Status and Journal Posting Indicator across all Posting Line records. Transaction status is only checked between the common and specific headers because it does not exist at lower levels. This process reads the Transaction Header Catalog (DOC_HDR) and selects records based on selection criteria.

For each record selected from the Transaction Header Catalog, this process finds all records from the other Catalog components (such as Vendor Catalog, Commodity Catalog, Accounting Line Catalog and Posting Line Catalog) and all Transaction specific components for those items, compares them to the Transaction Header Catalog record to ensure that all before mentioned attributes are consistent. Transaction specific components such as the line group of the journal voucher and the accounting distribution of the payment request, purchase order, and requisition are not part of this assurance job. Each transaction found to be out-of-sync is identified in a page header. Following it are all the components that were checked. A component is listed more than once if the transaction contains more than one record on that component. Each line for a component is listed along with the line numbers. When the job is run with or without any selection parameters, and no problems are found, the report is produced with a total section of zeros and the phrase, 'No out of sync records found."

Template transactions by design will not have all the required transaction components (for example, posting lines). These transactions are listed in the out log but not on the report because this is an acceptable condition.

The following table shows the various steps that the SA6 Job goes through and the messages issued at each step.

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.

Process Steps	Messages

2. Selection of records	• Selecting eligible records from the Transaction Header Catalog.
	  • If no eligible records are found, the following message will be issued: “No eligible records found that match the selection criteria.”
	  • If eligible records found, the following message will be issued: “Number of transactions selected = xxx”.
	• Selection of Records completed.

3. Reviewing transactions for out of sync condition	• Reviewing transactions for Out of sync:
	  • The following message will be issued during the review: “Number of transactions reviewed = xxx.” The above message will be issued initially after reviewing the number of transactions that matches the commit block size value. After that, the incremented value and the frequency of the message will be based on the commit block size.
	  • The following message will be issued when no inconsistencies found: “No out of sync records found”.
	  • The following message will be issued when any inconsistencies found: “Out of sync records found”.
	• Review completed.

4. Creating the Systems Assurance Transaction Component Listing report	• Rendering report started. • Rendering report completed.

Restart information: This job cannot be restarted. If the job failed for any parameter validation, then enter the correct parameter and schedule a new job. If the job failed for any other reason, investigate the reason for failure, correct it and schedule a new job.

Major Input

The Data objects, which are the input to the process

- Header Catalog (DOC_HDR)
- Vendor Catalog (DOC_VEND)
- Commodity Catalog (DOC_COMM)
- Accounting Line Catalog (DOC_ACTG)
- Posting Line Catalog (PSTNG_LN_CAT)
- Transaction Specific components (Examples: PO_DOC_HDR, PO_DOC_VEND, PO_DOC_COMM, PO_DOC_ACTG)

Batch Parameters

Parameter	Description
Client Name	Optional Parameter.
Transaction Code	Optional Parameter.
Earliest Date Range (MM-DD-YYYY)	Optional Parameter. If entered the value must be in the following format: MM-DD-YYYY.
Latest Date Range (MM-DD-YYYY)	Optional Parameter. If entered the value must be in the following format: MM-DD-YYYY.
Commit Block Size	Optional Parameter.

- Client Name to appear on the report. The name appears as the first line in the header of the report. This is an optional field with no default. The system wide default for Client Name is used when established.
- The Transaction Code field allows the program to review only a specific or a few specific transaction codes instead of all that exist in the application. The optional field allows for multiple transaction codes if separated by commas. There is no default for the field, which means that all transaction codes are checked. If entered it must be a valid Transaction code on the DCTRL table.
- Optional Earliest and Latest Date Range Fields exist to specify a specific range of transaction last dates (DOC_LAST_DT) to check. When an accounting line is found with a transaction last date greater than or equal to Earliest Date, it is selected. When an accounting line is found with a transaction last date less than or equal to Latest Date, it is selected. No edit exists to prevent the Latest Date from being prior to the Earliest Date because the dates are not formatted as such. Edits do exist to prevent invalid dates from being used. When a date is left blank that end of the range is open to the earliest or latest record. When dates are specified, the system defaults the time associated with them to 00:00:00. For this reason, if only one day is desired for assurance, that date should be entered in the Earliest Date field and the next day should be entered in the Latest Date field. If assurance is desired up through a certain date, then the next day after that one should be entered in the Latest Date field.

Output

- Systems Assurance Transaction Component Listing report
- JRNL_LOG - (Journal Log) – New records will be inserted with the job parameters.

Job Return code

The following table shows the potential return codes for the SA6 job.

Return Code	Condition
Successful (1)	This return code will be issued when all selected records are processed successfully and there are no out of sync conditions found.

Return Code	Condition
Warning (4)	This return code will be issued when there are no eligible records selected on the source table(s) based on the selection criteria.
Non Fatal Error (8)	This return code will be issued when the out of sync condition is encountered.
Failed (12)	This return code will be issued under the following conditions: • Parameters are not valid • 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 Code, Department, Transaction ID, Version Number and Line Numbers

Selection Criteria

- Select all transactions with matching Transaction Code parameter (or all transactions if Transaction Code parameter is left blank) with Transaction Last Date within the range of Earliest and Latest Date Range.
- If Earliest Date was left blank, then select records with a Transaction Last Date <= Latest Date.
- If Latest Date was left blank, then select records with a Transaction Last Date >= Earliest Date.
- If both Earliest and Latest Date were left blank, then select records with a Transaction Last Date <= Current System Date.
- If both Earliest and Latest Dates were not blank then select records with a Transaction Last Date >= Earliest Date and <= Latest Date.

Problem Resolution

There is no need to back out any updates if this job fails in any of the steps since this job does not update any tables. 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)	Parameters are valid.	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other instructions
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)	Entered Parameters are not valid **Sample Message**: Transaction code xxxx (xxxx being the value from the parameter) is not valid on the DCTRL table.	Enter a valid Transaction code and schedule a new job.	N/A
	Failed because of runtime exceptions for unexpected situation	Investigate the reason for failure, correct it and schedule a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	Investigate the reason for the termination, correct it and schedule a new job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues	Investigate the reason for System failure, correct it and schedule a new job.	N/A

Step 2: Selection of records

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

Possible Return Codes	Condition	Recommendation	Other instructions
Successful (1)	Eligible records selected from the Transaction Header Catalog.	N/A	N/A
Warning (4)	Job ended with a Warning because there are no eligible records on the Transaction Header	Review the batch parameters and schedule a new job with correct parameters.	N/A

Possible Return Codes	Condition	Recommendation	Other instructions
	Catalog. **Sample Message:** "No eligible records found that match the selection criteria".		
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Failed because of runtime exceptions for unexpected situation.	This step can fail with fatal conditions only on encountering unknown exceptions. If that happens, investigate the exception reported by the process, resolve the error and schedule a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	Investigate the reason for the termination, correct it and schedule a new job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues	Investigate the reason for System failure, correct it and schedule a new job.	N/A

Step 3: Processing records:

This step will be performed only if eligible records were selected for reviewing the out of sync condition.

Possible Return Codes	Condition	Recommendation	Other instructions
Successful (1)	No Out of Sync condition encountered	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	Out of Sync condition encountered. **Sample message**:	Review the Out of Sync Report and take appropriate action to correct the out of Sync Condition.	N/A

Possible Return Codes	Condition	Recommendation	Other instructions
	Out of Sync records found.		
Failed (12)	Failed because of runtime exceptions for unexpected situation.	This step can fail with fatal conditions only on encountering unknown exceptions. If that happens, investigate the exception reported by the process, resolve the error and schedule a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	Investigate the reason for the termination, correct it and schedule a new job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	Investigate the reason for System failure, correct it and schedule a new job.	N/A

Step 4: Creating the Systems Assurance Transaction Component Listing report

This step will be performed after reviewing all the transactions for the out of sync condition.

Possible Return Codes	Condition	Recommendation	Other instructions
Successful (1)	System Assurance Transaction Component Listing Report created successfully.	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Failed because of runtime exceptions for unexpected situation.	This step can fail with fatal conditions only on encountering unknown exceptions. If that happens, investigate the exception reported by the process, resolve the error and schedule a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	Investigate the reason for the termination, correct it and	N/A

Possible Return Codes	Condition	Recommendation	Other instructions
		schedule a new job.	
System Failure (20)	When the job is terminated because of database server or network issues.	Investigate the reason for System failure, correct it and schedule a new job.	N/A

2.1.7 Systems Assurance 07 – Journal/Ledger Cross Reference table In-sync

When to Run

On-demand or as part of the nightly systems assurance process. The assurance should only be done after all journals have been ledgerized with the Ledger Engine being run. If running SA07 in full mode, not ledgerizing first will result in all unledgerized journal records being listed as out of sync conditions. If running SA07 in incremental mode, those unledgerized journal records will not be considered by the SA07 program. Please see the Problem Resolution section for more details on out of sync conditions. This assurance should not be run if any ledgers are currently being rebuilt, for records not yet ledgerized will be listed in the report.

Description

This assurance job is used to validate that all journal records reported to be ledgerized into one or more ledgers, based on entries created in the **Journal Log** by the Ledger Engine, were in fact posted to that ledger or ledgers by reading the **Journal/Ledger Cross Reference** table. The assurance can work in one of two modes. Full mode, which is indicated by the specification of a ledger as a batch parameter, will evaluate all journal records in the source journal for that ledger. Incremental mode, which is indicated by leaving the ledger specification blank, will evaluate all journal records ledgerized since the last run of SA07 in incremental mode.

Full mode can list journal records as out of sync for a number of reasons:

1. Not in ledger because ledger was not active at the time journal records were created
2. Not in ledger because the block of Ledger Engine work that contained the journal record failed to ledgerized
3. Not in ledger because the block of Ledger Engine work that would have contained the journal record was recorded as a gap that was filled in later
4. Not in ledger because Ledger Engine has not been run to select new journal records

Incremental mode can list journal record from the first 3 of those scenarios listed above. The last is not listed because the incremental model evaluates only those journal records ledgerized since the last run of SA07 in incremental mode. This is determined by comparing the END DOC UNID from the last SA07 incremental run entry in the **Journal Log** to the END DOC UNID of the last ledger post entry in the **Journal Log**. Any journal entries created since the last Ledger Engine will not be listed with an incremental run. This is because these entries should be evaluated in the next SA07 run because they should be ledgerized by that point.

When it is desired that all ledgers be evaluated in a full mode, the deletion of all Journal Log entries for SA07 will allow a run of SA07 without any ledger specified to evaluate each from the beginning journal record to the last journal record ledgerized.

The first output is a report produced showing a ledger on each page break. For that ledger, the journal records that were found to not exist on the **Journal/Ledger Cross Reference** are listed. Shown for each journal record is the record number and the transaction ID down to the posting line that created the ledger record. An error message stating that the journal record “Did not post to ledger” follows each record. When no ledger records are found to be our-of-sync, the report is still created and has a single message, “All Journal and Ledger Records Are In Sync.”

The second output is a record to the **Journal Log**. When run to verify a single ledger the process creates a record on the Journal Log after a successful run recording the Job ID (SA07), the ledger input parameter in the Journal Name field, date and time stamp when run, and a description of “Journal/Ledger Cross Reference Recheck Ledger.” Nothing is recorded in the

Transaction Unique ID fields. When run to incrementally verify all journals that post to ledgers, the Job ID is the same, there are individual records created for each journal that supports ledgers, date and time stamp when run, and the description for each is “Journal/Ledger Cross Reference Incremental.” The Beginning Transaction ID and Line store the information from the first journal record read, and the Ending Transaction ID and Line store the information from the last journal record read.

Multiple instances of this report are permitted only when running in full mode with individual ledgers being specified. Running incrementally against a single ledger is not permitted as entering the Ledger Name parameter signifies a full run.

Major Input

- Any single ledger and the journal it is created from
- All ledgers and the journals they are created from
- JRNL_LDGR_XREF (Journal Ledger Cross Reference)
- JRNL_LOG (Journal Log)
- JRNL_LDGR_CTRL (Journal/Ledger Controls)

Output

- JRNL_LOG (Journal Log) - updated
- System Assurance 7 Report

Parameters

Batch Parameters

Description (Caption)	Parameter Name	Default Value
Client Name appearing on report.	Client Name	
Only use to verify a single ledger in a non-incremental fashion. Leave blank for normal incremental execution against all ledgers. The value entered should be an uppercase DataObject name of a ledger as found on the Journal Ledger Control table.	Ledger Name	
Maximum Pre-fetch Count used to control the number of ledger records selected.	MAX_PREFETCH_COUNT	

- Client Name to appear on the report. The name appears as the first line in the header of the report. This is an optional field with no default. The system wide default for Client Name will be used when established.
- Ledger Name is the data object name of the table to use as the ledger to be assured. There is no default value for this required field. Any value specified must be entered in the same ‘all capitals’ format. An invalid value entered will cause the job to fail, issuing an error to the log. When no value is entered, the program runs against all ledgers.
- MAX_PREFETCH_COUNT defines the number of records saved into memory after a query has been performed. This is an optional field with a default value of 1000.

Selection Criteria

See parameters.

Problem Resolution

When journal records are listed as out of sync for a ledger, the first remedy is to run the Ledger Engine in all three modes: Normal, Gap Processing, and Failed Work Processing. After those runs, if the SA07 report still lists journal records, the JRNL_LOG table should be searched to find any units of work for the Ledger Engine that are in a limbo status. Search for PROC_ID = LDGRPOST where ST_FL = 1, 6, or 10. If any are found, then the ST_FL needs to be set to 4, 9, and 4 respectively. The Ledger Engine should then be run in Failed Work mode if any records were changed to ST_FL = 4. The Ledger Engine should then be run in Process Gap mode if any records were changed to ST_FL = 9. If the report still lists a record or records that are marked as "Did not post to ledger," then the ledger engine is having difficulty with that record or records. The Problem Resolution section of the Ledger Engine batch job should be consulted.

For performance reasons, the Identify and Archive Stale Gaps job should be run periodically to remove gaps in journal records being tracked and processed by this program. Please refer to the Identify and Archive Stale Gaps run sheet in the *CGI Advantage Financial – General Accounting Run Sheets* guide for more details on scheduling and recommended parameters.

2.1.8 Systems Assurance 08 – Ledger to Journal Diagnostic Process

When to Run

On-demand or as part of the nightly systems assurance process

Description

This system assurance process allows the user to verify that each ledger record sums up to the journal records reported to have been summarized into that ledger record. This systems assurance process is intended to be used on an ad-hoc basis, specifically when certain ledger record amounts need further investigation. One recommendation is to assure the input ledger into the Annual Close process before closing a year.

Job reviews the line amounts on a ledger record with the line amounts of journal records that the **Journal/Ledger XREF** table says are in that ledger record to assure that the ledger line amount is the sum of all journal records ledgerized into it. This job is intended to supply three types of verification. The first verification is to see if the Ledger Engine updated the cross reference table but not the ledger record. The second is to see if the Ledger Engine updated the ledger record but not the cross reference table. The third is not an application processing verification but a database one to see if any records have been corrupted on the ledger, cross reference table, or the source journal of that ledger.

A report is produced showing a ledger record on each page break. Shown for records that are out of balance, is all the journal records found that should be part of that ledger record identified by both a journal record ID and the corresponding transaction ID down to the posting line number. Also shown is the key and ledger record ID found to be out of sync, along with amounts for the ledger record and journal records. When a ledger record is found to be out-of-sync, then an error message is listed next to the total amount for journal records, “Ledger Summary Total Not = Journal Total.” When no ledger records are found to be our-of-sync, the report is still created and has a single message, “No Out of Sync Ledger Records.”

The process also creates a record on the Journal Log after a successful run recording the Job ID (SA08), the ledger input parameter in the Journal Name field and a description of “Ledger Diagnostic Process.” If the Begin and End Record Number parameters were used, then they are listed in the Beginning Journal Record Number and Ending Journal Record Number columns. If only one Ledger Record parameter was used, that value is in both Journal Record Number fields. This entry is not used by subsequent runs of the SA08 job, but is just for a historical record.

ITD Ledgers, those defined to not retain either Fiscal Year or Budget Fiscal Year, will not work with this systems assurance program after the source journal to such a ledger has been archived. The reason for this is that the archival of records from that journal and the subsequent removal of those same journal records from the cross reference table. With the clearing of these two sources of data, the assurance of any inception-to-date ledger will not be possible.

Major Input

- Any single ledger and the journal it summarizes
- JRNL_LDGR_XREF (Journal Ledger Cross Reference)

Output

- System Assurance 8 Report
- JRNL_LOG (Journal Log) updated

Parameters

Batch Parameters

Description (Caption)	Parameter Name	Default Value
Client Name appearing on report.	CLIENT_NM (Client Name)	(blank)
Beginning of range of ledger REC_NO for which to check for out-of-sync condition. Leave blank if specifying 'Record Number' or to indicate report should start at the lowest REC_NO on the ledger.	From Record Number	(blank)
Ledger Name	Ledger Name	LDGR_SA_BUD
1= Report Out-of-sync, 2=Report All.	Mode	1
Ledger REC_NO for which to check for out-of-sync	Record Number	
End of range of ledger REC_NO for which to check for out-of-sync condition. Leave blank if specifying 'Record Number' or to indicate report should end at the highest REC_NO on the ledger.	To Record Number	
Set the Commit Block Size using this value.	Commit Block Size	
Set the select block size using this value.	Select Block Size	

- Client Name to appear on the report. The name appears as the first line in the header of the report. This is an optional field with no default. The system wide default for Client Name will be used when established.
- From Record Number is optional and has no default value. It is the beginning ledger record number (value of REC_NO) for which the user wants to produce diagnostics. If blank, diagnostics will be produced starting from the ledger record with the lowest REC_NO value. Only provide a value for this parameter if you wish to provide diagnostics for a subset or range of ledger records. You may provide a value for From Record Number without providing a value for To Record Number.
- Ledger Name is the data object name of the table to use as the input ledger. The default if LDGR_FYTD_ACTG (Fiscal Year to Date Accounting Ledger) but any other valued may be entered in the same ‘all capitals’ format. An invalid value entered will cause the job to fail, issuing an error to the log. Multiple ledgers cannot be entered. If it is desired to check more than one ledger, the job must be run multiple times with different parameters.
- Mode may be 1 or 2. Mode = 1 provides diagnostics for only those selected ledger records that are out-of-sync. Mode = 2 provides diagnostics for all selected ledger records. The user should expect the report to take significantly longer to process for a large range of ledger records with Mode = 2. Note: For Mode 2 even if there are only In sync conditions (that is no out of sync records), job return code would be the Non fatal error and job log would display the “Out of Sync condition found" message. Running in mode 2 is not suggested as it will produce a report that is too large to work with if there has been significant volume in journal records.
- Record Number is optional parameter, which will point the program to just one record on one ledger. Specify the ledger record number (value of REC_NO) for which the user wants to

produce diagnostics. Only provide a value for this parameter if you wish to produce diagnostics for a single ledger record. There is no default for this parameter and it will not accept multiple values separated by commas.

- To Record Number is optional and has no default value. It is the ending ledger record number (value of REC_NO) for which the user wants to produce diagnostics. If blank, diagnostics will be produced ending with the ledger record with the highest REC_NO value. Only provide a value for this parameter if you wish to provide diagnostics for a subset or range of ledger records. You may provide a value for From Record Number without providing a value for To Record Number.

Sort Criteria

Ledger records in the output report are automatically sorted by the ledger record's Record Number (REC_NO). ournal records for each reported ledger record are also automatically sorted by their Record Number (REC_NO).

Selection Criteria

See parameters

Problem Resolution

No database restore is required. Rerun the job.

2.1.9 System Assurance 09 – Customer Account Assurance

Job Name	System Assurance09
Recommended Frequency	On-demand or as part of the nightly system assurance process
Single Instance Required	Yes
Can be Restarted?	Yes
Reports Generated	Yes

Overview:

Job reviews records on the Customer Account (CUSTA) page and performs a balance check between the stand alone buckets (amounts updated directly by transactions and not calculated amounts) on that table with the records found in the journal specified as the input parameter and records in the Posting Line Catalog that have a final or historical transaction status but not yet posted to any journals. All amounts are checked except the 3rd party ones, reserved credit balance, and unreserved credit balance. The short and over payment buckets are validated in a net effect manner.

The first output is to the SA_CUST_ACCT, a shadow table, which allows the job to re-post activity to a shadow table for comparison to the Customer Account page. The first time the job is run in the *incremental* or *full mode*, records are created for each customer account; recording activity for each amount. Subsequent runs in incremental mode update the existing records and create new records for customer accounts not used before. These subsequent runs look to the Journal Log (JLOG or JRNL_LOG) to find where the previous run left off in the input journal and starts from that point. Running incremental mode performs much better than a full run; however, a full run should be done periodically. When the job is run in the full mode, the shadow table is purged of all records, except those from a *pre archive* mode, and new records are created from the first transaction processed against a customer account. The system does not create records on the shadow table for archived customers because CUSTA and shadow table records are removed when a customer is archived.

The second output is a report is produced for each customer account record (that is, Customer account + Billing Profile + AR Dept + AR Unit) and amount that is not in sync. Information is given for:

1. The CUSTA amount
2. Total amount found in the journal
3. Total amount found in the Posting Line Catalog
4. Total amount archived from the journal
5. The difference (which is CUSTA – posting catalog total –journal total – archive total)

When no customer account records are found to be our-of-sync, the report is still created and has a single message, “All Customer Account Records are in Sync.” The diagnostic details to be displayed in report are generated from the input journal (most often the Accounting Journal) and Posting Line Catalog. The input parameters ‘Diagnostic From Date’ & ‘Diagnostic To Date’ assist users with the diagnostic effort by limiting the data presented in the report to a specific date range, so evaluation can be done since the point the line became out of sync.

The third output is a record on Journal Log. After a successful run recording the Job ID (SA09), the journal input parameter, a beginning transaction ID and line, an ending transaction ID and line, and a message (Customer Account VS Journal and Posting Lines) is written out to the log. That message is followed by the mode: Full, Incremental, or Pre-Archive.

When a journal archive is planned for the journal used in this system assurance job, prior to running the Journal/Ledger archive job, this SA9 job has to be run in Pre-Archive mode. In Pre-Archive mode, the job will empty all records from SA_CUST_ACCT, except for those with Posting Source of “Archived” (4). The process will add/make updates to these “Archived” records. The Journal/Ledger Archiving chain then deletes the journal records for the Archive FY. A Full run of the System Assurance 9 is expected at any time after archiving is complete. Please plan for additional processing time of this full run since incremental runs are the norm.

If the pre-archive mode has been selected the SA09 report will not be generated. When an archive has been performed against the journal used for the job, when records are displayed for the archived year(s), the header on the column will denote that record as being archived.

Process Steps	Messages
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
Reposting the records	No progression messages are written while the Reposting of records occurs.
Creating System Assurance 09 Report	Progression messages when the Reposting of Records is completed: • Out of Sync condition found • Completed building the report • Completed generating the report • Run Ended

Major Input

- Journal (JRNL_ACTG) - The record in the Journal/Ledger Control table for the specified journal must have System Assurance 09 checked.
- Posting Line Catalog (PSTNG_LN_CAT)
- Customer Account table (R_CUST_ACCT)
- Event Type Table (R_EVNT_TYP)
- Vendor Customer Archive (R_VEND_CUST_ARCH)

Batch Parameters

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

Parameter Name	Description	Default Value
CLIENT_NM	Client Name Optional field. Appears as the first line in the header of the report. The system wide default for Client Name will be used when established.	No default
Journal Name	Journal Name Required field. Journal to use for assurance. Enter the data object name in all capital letters. The Accounting Journal is the best source.	JRNL_ACTG
COMMIT_SIZE	Commit Size Optional field. A performance parameter defining the number of records that can be processed before using a database commit. If left blank the system will default 100. Must be a positive integer.	100
ARCHIVE_FY	Archive Fiscal Year Required for Pre-Archive mode (3) only. This parameter will filter journal records for the FY to be archived.	No default
DIAG_FROM_DT	Diagnostic From Date Optional field. It is the lower limit to filter diagnostic records being written to the report. If the field is populated, only those records that have a Table Last Date on or after this date will be written to report. The required format of the date is 'mm/dd/ccyy'. If the parameter is omitted, no lower limit will be imposed on the retrieval of diagnostic records.	No default
DIAG_TO_DT	Diagnostic To Date Diagnostic To Date Optional field. It is the upper limit to filter diagnostic records being written to the report. If the field is populated, only those records that have a Table Last Date on or before this date will be written to report. The required format of the date is 'mm/dd/ccyy'. If the parameter is omitted, no upper limit will be imposed on the retrieval of diagnostic records.	No default
MAX_PREFETCH_COUNT	Maximum Pre-fetch Count Optional field.A performance parameter	No default

	defining the number of records that can be retrieved from Journal Log for processing before issuing a database commit.	
PROG_CTR_SZ	Progression Counter Size Optional field but will default to 1000 if left blank. It will determine how many records to process before echoing a running count to the log.	1000
RUN_MODE	Run Mode Required field. Run Mode may be set to **1** to run the job in *Full* mode. A value of **2** (*Incremental*) means the records are processed from the last journal record number processed during last Incremental or Full run mode. For value of **3** the job will run in *Pre-Archive* mode and will not generate any report. A value of **4** (*Report*) means no processing of records. Only the report will be generated here.	No default
SHOW_JOURNAL_DIAG	Show Journal Records Optional Field. Show Journal Diagnostics may be set to **Y** to produce diagnostics for journal records were posted for an out of sync bucket under a separate heading of Journal Diagnostic Detail. Such detail will show the journal record ID, the transaction ID, and the amount of the journal record. The default is **N**, which will not show any such information. If the parameter value is invalid or left blank, the system assumes **N.** The user should expect the report to take significantly longer to process if any diagnostics are enabled. Use the Diagnostic From and To Dates to limit the amount of data.	N
• SHOW_PLCAT_DIAG	• Show Posting Line Records • Optional Field. Show Posting Line Diagnostics may be set to **Y** to produce diagnostics for Posting Line records for an out of sync bucket under a separate heading of Posting Line Diagnostic Detail. Such detail will show the record ID, the transaction ID, and the amount of the Posting Line record. The default is **N**, which will not show any such information. If the parameter value is invalid or left blank, the system assumes **N.** • Note - the report may take	N

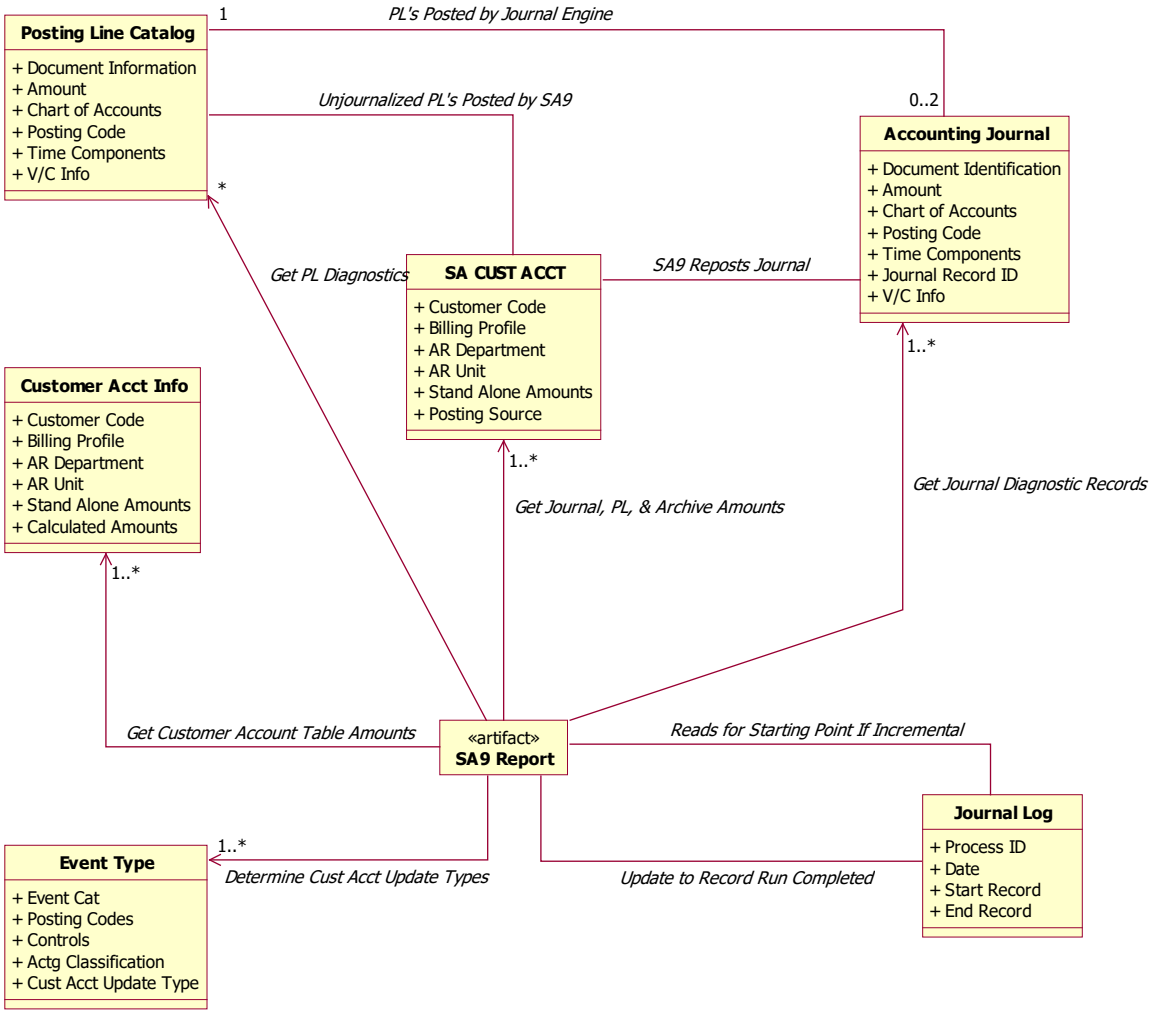
	significantly longer to process if Show Posting Line Diagnostics is set to **Y**, but volume here should not be high with prompt running of the Journal Posting Initiator batch job and the Journal Engine.	

Major Output

- SA09 Report – when mode is not pre-archive
- SA_CUST_ACCT shadow table
- JRNL_LOG (Journal Log) – updated

Report Sample

Class Diagram of Process



Job Return code

The following table shows the potential job return codes for the System Assurance 9 job.

Return Code	Condition
Successful (1)	All of the records are processed successfully and no out of sync condition is reported.
Warning (4)	N/A
Non Fatal Error (8)	Out of sync condition encountered.
Failed (12)	The job will fail under the following conditions: Required parameters are not entered Entered parameters are invalid Run time exceptions for unexpected situations
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Sequence

Customer, Department, Unit, Billing Profile

Selection Criteria

- Journal records are selected if they satisfy the following:
  - VEND_CUST_CD, AR_DEPT_CD, AR_UNIT_CD, BPRO_CD are not null
  - VEND_CUST_CD is not found in the R_VEND_CUST_ARCH table with the FIN_CMPLT_FL indicator set to 1 (completed), which indicates that the Vendor/Customer code has been successfully archived, and should not be considered in this selection.
  - For incremental runs, DOC_UNID > END_DOC_UNID recorded from last SA09 run on JRNL_LOG
- Posting Line records are selected if they satisfy the following:
  - DOC_PHASE_CD = final
  - VEND_CUST_CD, AR_DEPT_CD, AR_UNIT_CD, BPRO_CD are not null
  - VEND_CUST_CD is not found in the R_VEND_CUST_ARCH table with the FIN_CMPLT_FL indicator set to 1 (completed), which indicates that the Vendor/Customer code has been successfully archived, and should not be considered in this selection.
  - JRNL_PSTNG_IND = “Not Ready” or “Ready” or PSTNG_CD_ID is a memo account
- All Customer Account records are selected.

Problem Resolution

If the process was discontinued for any reason then the job has the ability to be restarted from the point it left off only if it was run in the Incremental Mode.

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

Step 1: Parameter Validation

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	Successful	N/A	N/A
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	N/A	This step does not issue this return code.	N/A
Failed (12)	Required Parameters are not entered. **Sample Message:** Run Mode parameter is required.	Schedule a new job after entering the proper Run Mode.	
	Entered Parameters are not valid. **Sample Message:** Diagnostic From Date is not in the valid format mm/dd/ccyy	Schedule a new job after entering the correct Diagnostic From Date	
	Failed because of runtime exceptions for an unexpected situation.	The failure reason needs to be investigated before scheduling a new job	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling a new job.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before scheduling a new job.	

Step 2: Creating System Assurance 09 Report - This step will be performed only when the parameters are valid.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	No out of sync condition encountered.	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
Warning (4)	N/A	This step does not issue this return code.	N/A
Non Fatal Error (8)	Out of sync condition encountered.	Verify the out of sync condition and take the appropriate action to resolve the out of sync condition.	N/A
Failed (12)	Failed because of runtime exceptions for an unexpected situation.	The failure reason needs to be investigated before scheduling a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before scheduling a new job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues	The reason for the System Failure needs to be investigated before scheduling a new job.	N/A

Troubleshooting

If the job encounters s system failure while running in the “full run” mode, correct the problem and rerun the job in the “full run” mode. The job cannot be rerun in the “incremental run” mode unless the SA_CUST_ACCT table is restored.

In all other cases of an error with the batch job, no database restore is required. Correct the problem and rerun the job.

Out of sync conditions that are the result of transaction processing or data setup on the Event Type table that have changed (Customer Account Update Flag and Customer Account Update Type) will require database updates to the Customer Account table.

For performance reasons, the Identify and Archive Stale Gaps job should be run periodically to remove gaps in journal records being tracked and processed by this program. Please refer to the Identify and Archive Stale Gaps run sheet in the *CGI Advantage Financial – General Accounting Run Sheets* guide for more details on scheduling and recommended parameters.

2.1.10 Systems Assurance 10 – Link Assurances

The Systems assurance job, SA10 Budget Linking Assurance, is developed to:

- Verify that the linked amount stored on the link table is calculated correctly.

Verify that the linked revenue amounts on expense budget lines are correct at the linked level as defined by the floor, ceiling, and percentage link parameters.

Description

SA 10 is used to verify the links created between the expense and revenue budget structures are in sync. The output from this job consists of two Link Assurance Reports. If any of the links are determined to be out-of-sync, the job will create a record on one of the two reports. The reports will be written to depending on the scenario(s) causing the out-of-sync condition. The two available reports are:

- Link Assurance Report 1 - This report displays the links which are causing the out-of-sync condition and handles situations where the linked amounts on the offline tables specific to linking are out-of-sync.
- Link Assurance Report 2 - This report displays the links which are causing the out-of-sync condition and handles situations where there is a difference between the total linked amount on the link table and the linked amounts on the expense budget lines.

The Job will return a status of Non Fatal Error if at least one record is found to be out-of-sync, Successful if no records are found to be out-of-sync, and Failed if any exception is encountered or invalid parameters are entered.

There are three report modes available for SA10. The three modes determine which reports will be generated when the job is executed. Report Mode 1 will create Link Assurance Report 1 and it will verify that the linked amounts on the link tables are in sync. Report Mode 2 will create Link Assurance 2 and it will verify the linked amounts on an expense budget tables are in sync. Finally, Report Mode 3 will generate both reports and will verify that both the linked amounts on the link tables and the linked amounts on the expense budgets are in sync.

More specifically, the process in mode (1) or (3) will take the links defined for those expense budget lines found on the expense budget structure for a particular budget level / BFY combination and re-compute the linked amount using the floor, ceiling, and link percentage. It will retrieve the linked bucket amount from the revenue budget line and perform the linked amount calculation logic. Next, that calculated amount will be compared to the linked amount stored on the link table. The recalculated amount will be subtracted from the amount on the link table. Any difference between these numbers will be specified in Link Assurance Report 1 along with specifics on the linked amount from the link table and the re-calculated amount from the revenue budget lines.

The process in mode (2) or (3) will take the links defined for those expense budget lines found on the expense budget structure for a particular budget level / BFY combination and compare the linked amount(s) on the budget line to the sum of the linked amounts on link records. When subtracted from each other, these amounts should equal zero. If the links for any expense budget are determined to be out-of-sync, a record containing all the links for the expense budget line, the link amounts for the link records, and the linked amount from the expense budget line will be written to the report. The report will also specify the bucket name that is out-of-sync on the expense budget. If more than one bucket is found to be out-of-sync for a given expense budget line, then all out-of-sync records for that budget will be shown together in the report.

When to Run

On-demand or as part of the nightly systems assurance process.

Major Input

- Expense and Revenue Budget Structures which can be linked (BUD_STRU_29_LVL_2, BUD_STRU_30_LVL_1, BUD_STRU_1_LVL_3, BUD_STRU_2_LVL_4)

  Budget Linking has been currently designed to allow linking for the above mentioned Budget Structures only. If later the Budget Linking process is modified to allow for other Budget Structures to be Linked then those Budget Structures will also have to be added.
- General Link Table (GN_LNK)

Output

- Link Assurance Report 1 Assures that the linked amounts stored on the General link table (GN_LNK) are calculated correctly.
- Link Assurance Report 2 Assures that the linked revenue amounts on expense budget lines at the linked level are what they should be according to the links defined.

Parameter	Description	Default Value	Required	For System Use (non-overrideable by user)
BFY	Budget Fiscal Year.	No Default	No	Yes
Client Name	Client Name appearing on report	No Default	No	Yes
EXP_BUD_STRU_ID	Expense structure ID used to find links of a particular structure	No Default	Yes	Yes
EXP_BUD_LVL_ID	Expense level ID used to find links of a particular structure.	No Default	Yes	Yes
REV_BUD_STRU_ID	Revenue structure ID used to find links of a particular structure.	No Default	Yes – If MODE is 2 or 3	Yes

Parameter	Description	Default Value	Required	For System Use (non-overrideable by user)
REV _BUD_ LVL _ID	Revenue level ID used to find links of a particular structure.	No Default	Yes– If MODE is 2 or 3	Yes
MODE	The process can have 3 different run modes: (1) Verify Linked Amounts on Links;.(2) Verify Linked Amounts on an Expense Budget. (3) Verify Both.	3	Yes	Yes

- Budget Fiscal Year (BFY) indicates the budget fiscal year for which the report should be executed. It limits the number of link records being verified. If no BFY is specified then link records for all BFY are selected. More than one BFY can be specified by separating the values with commas.
- Client Name to appear on the report. The name appears as the first line in the header of the report. This is an optional field with no default. The system wide default for Client Name will be used when established.
- Expense Budget Structure Id and Expense Budget Level Id indicate the expense structure for which the link assurance should be executed. If no value is entered then an error message is logged to the Log File.
- Revenue Budget Structure Id and Revenue Budget Level Id indicate the revenue structure linked with the expense Structure for which the link assurance should be executed. These are required fields. If no value is entered and the Mode is 2 or 3 then an error message is logged to the Log File.
- The Report Mode indicates the reports to be generated. The default mode is 3. If no value is entered or the Run mode is not between 1 and 3 then an error message is logged to the Log File.

  Note: If no link can exist for the entered Expense Structure and level with the entered Revenue Structure and level then an error is logged to the Log File.

Sort Sequence

- For Link Assurance 1 report the selected records are ordered by FRM_CONCAT_KY, FRM_BKT_ID, TO_CONCAT_KY, TO_BKT_ID
- For Link Assurance 2 report the selected records are ordered by TO_CONCAT_KY, FRM_CONCAT_KY, TO_BKT_ID, FRM_BKT_ID

Selection Criteria

Link Assurance Report 1

The selection criteria for processing records from the General Link Table (GN_LNK) is:

- Expense Structure Id = Expense Budget Structure Id entered as Parameter
- Expense Level Id = Expense Budget Level Id entered as Parameter
- Budget Fiscal Year In (Expense BFY entered as Parameter)

The report will be ordered by:

- Revenue Concat Key
- Revenue Bucket Id
- Expense Concat Key
- Expense Bucket Id

Link Assurance Report 2

The selection criteria for processing records from the General Link Table (GN_LNK) is:

- Expense Structure Id = Expense Budget Structure Id entered as Parameter
- Expense Level Id = Expense Budget Level Id entered as Parameter
- Revenue Structure Id = Revenue Budget Structure Id entered as Parameter
- Revenue Level Id = Revenue Budget Level Id entered as Parameter
- Budget Fiscal Year In (Expense BFY entered as Parameter)

The report will be ordered by:

- Expense Concat Key
- Revenue Concat Key
- Expense Bucket Id
- Revenue Bucket Id

Note: If the BFY is not entered as a parameter then the link records for all BFY(s), which match all the other selection criteria/parameters, will be selected.

Problem Resolution

If the process was discontinued for any reasons then the job has the ability to be restarted from the point it left off.

2.1.11 Systems Assurance 11 – Budget Systems Assurance

The systems assurance job, SA11 Budget Systems Assurance, verifies the Budget Journal record amounts with the budget amounts in the Budget Tables.

Description

The Budget Systems Assurance 11 job will verify that records in the budget journal are in sync with budget amounts recorded on all budget tables. These actual amounts are the stand-alone budget buckets that are updated by budget transactions. Examples of budget buckets include: Adopted, Allocated, Amendments and Revenue Budget Reversion amounts. The actual amounts are the event-driven budget bucket amount fields updated by budget transactions. When the activity recorded in the journal does not match a budget bucket amount field, the job writes a line to the SA11 report with the supporting information that was requested in the input parameters.

The Job will return a status of Non Fatal Error if at least one record is found to be out-of-sync, Successful if no records are found to be out-of-sync, and Failed if any exception is encountered or invalid parameters are entered.

There are three report modes available for SA11:

- **Full Run Mode** - In this mode the process will purge the data from the Shadow table as per the parameters specified by the user and repost the records from the Source Journal to the Shadow table. Then the process will generate the report on the basis of the parameters specified by the user. The Journal Log will hold the last processed Record Number.
- **Report Only** - In this mode the process will generate the report for the records, which are already posted in the Shadow table. The Journal Log will not be affected in this mode. Unlike the other two modes that allow for targeting a single budget structure and level, this mode reports on all active structures for levels with data.
- **Incremental Mode** - In this mode the process will purge the data from the Shadow table only if there was no record found in the Journal Log for this process from the prior run. Then it will repost the records from the Source Journal to the Shadow table that are greater than the End Journal Record Number in the Journal Log and less than the Last Record Number in the Source Journal. While reposting the records the process will ignore the parameters specified by the user. Parameters specified by the user will be considered while generating the report.

Performance Recommendation

Additional indices should be added to the Budget Journal in order to assist SA11 in performing its work. The indices should reflect the budget structures that your specific site uses. If, for example, your site uses Structures 30, 39 and 80, three indices should be added to the Budget Journal, one for each structure. We recommend adding only one index per structure at the lowest required budget level as the most effective method for this process, while it will have only a marginal affect on the processing of budget transactions.

The three indices to be added for this fictitious site would be the primary key of Level 1 for Structure 30, the primary key of Level 2 for Structure 39 and the primary key of Level 3 for Structure 80. It is recommended that you add one index per structure and the fields inside of that index will be down the lowest required level for the structure. So, for Structure 30, since it has only 1 level, it will have an index for level 1. Structure 80 has 4 levels, however in baseline Level 4 is an optional level. So, the most effective index would be with the key fields for Level 3.

The indices' SQL would look like the following for an Oracle DB:

CREATE INDEX NJRNL_BUD_ST80 ON JRNL_BUD

(BFY, FUND_CD, DEPT_CD, APPR_CD, UNIT_CD, OBJ_CD, ACTV_CD)

CREATE INDEX NJRNL_BUD_ST3 ON JRNL_BUD

(BFY, FUND_CD, DEPT_CD, UNIT_CD, RSRC_CD)

CREATE INDEX NJRNL_BUD_ST39 ON JRNL_BUD

(DEPT_CD, MJR_PROG_CD, PROG_CD, PPC_CD, FPRFL_CD, FPRTY_CD, FLINE_CD)

It is recommended that one index be added per structure that is used by your location to the Budget Journal. The index will be the primary key fields for the lowest required level in each structure at your site in use. No more than one index per structure should be added to the budget journal.

When to Run

For enhanced performance, it is recommended that the SA11 process run in Incremental Mode on a nightly basis, and then periodically in Full Run Mode.

Major Input

- All budget tables (for example, bud_stru_31_lvl_1 for structure 31 level 1), excluding the Activity Tables.
- All allotment tables (for example, a lot_stru_26_lvl_1 for structure 26 level 1), excluding the Activity Tables.
- Structure Table – GN_BUD_STRU
- Bucket Table – R_GN_BKT
- Budget Level Control Table – GN_BUD_LVL
- Posting Code Table – R_PSCD
- Budget Journal – JRNL_BUD (can be any other budget related Journal)

Output

- SA11 Shadow Table
- Systems Assurance SA11 Report
- Entry into Journal Log

Parameters

Description (Caption)	Parameter Name	Default Value
Budget Fiscal Year	BFY	
Budget Structure	Structure	
Budget Structure Level	Level	
Client Name appearing on report	Client Name	
Journal to use for assurance	Journal Name	JRNL_BUD
Report all or Report out of sync	Report Level	N

only. Valid values (Y/N).		
Show budget activity for buckets that are out of sync. Valid values (Y/N).	Show Budget Diagnostics	N
Reveal miscellaneous identifiers in report for diagnostic purposes. Valid values (Y/N).	Show Identifiers	N
Show journal records for buckets that are out of sync. Valid values (Y/N).	Show Journal Diagnostics	N
Run Mode. 1 = Full Run, 2 =Report Only, 3 = Incremental Run	RUN_MODE	1
Assure allotments. Valid values(Y/N).	Check Allotments	Y

- Budget Fiscal Year (BFY) indicates the budget fiscal year for which the report should be executed. It limits the number of budget records being verified. All budget records that do not have a BFY in their key are always selected. More than one BFY or “9999” can be specified by separating the values with commas. This is a required field.
- Budget Structure indicates the structure for which the report should be executed. It limits the number of structures being verified. More than one structure can be specified by separating the values with commas. If no structures are specified, the system will use all active structures. When the process is run in Report Only mode, the Structure ID entered in this parameter will not be considered and the process will use all active structures.
- Budget Structure Level indicates the level for which the report should be executed. It limits the number of levels being verified. One level can be specified to correspond to each structure specified. If no levels are specified, the system will use all active (non-audit) levels for each structure specified. When the process is run in Report Only mode, the Structure Level entered in this parameter will not be considered and the process will use all active levels.
- Client Name to appear on the report. The name appears as the first line in the header of the report. This is an optional field with no default. The system wide default for Client Name will be used if one exists.
- Journal Name is the data object name to use as the input source.
- Report level determines whether the report produced will list all budget records or just list those with an out of sync condition. The default for this field is N, which will cause the report to show only those buckets for budget records that are out of sync.
- Show Budget Diagnostics may be set to Y to produce diagnostics from the activity level of a budget structure, showing all non-zero transaction activity against an out-of-sync bucket. The default is N, which will not show such details.
- Show Identifiers may be set to Y to reveal internal identifiers for buckets and budget lines, more specifically the Bucket ID and Line ID. These ID numbers are helpful in quickly locating budget records on the budget tables. The default is N, which will not show any such information.
- Show Journal Diagnostics may be set to Y to produce diagnostics for journal records that should have been posted for an out of sync bucket under a separate heading of Historical Detail. The total of these records appears in the Recorded on the Journal field of the report. Such detail will show the journal record ID, the transaction ID, and the amount of the journal

record. The default is N, which will not show any such information. The user should expect the report to take significantly longer to process if any diagnostics are enabled.

- Run Mode may be set to '1' to purge the existing Shadow table, reprocess the source input file and repost the summarization results for comparison with budget tables. A value of '2' means the system will reuse the existing Shadow table of processed source records for comparison with budget tables. The '2' value would be useful if the user wants to regenerate an out-of-sync report to include diagnostic detail (i.e. the initial report did not include diagnostic detail). Run Mode may be set to '3' to allow the system to use data in the existing Shadow table incrementally adding data from the Budget Journal from the last record read as defined by an entry on JLOG. If no record exists on JLOG for SA11, the system will select records from the first line to the end of the table.
- Check Allotments may be set to Y to verify any allotment periods defined for the structure(s).

Sort Sequence

Sorted on the basis of the Key Fields for a particular Structure Id, Level Id and -Posting Code, Allotment period.

Selection Criteria

While reporting, the process will generate the report based on BFY, Structure, Level, Budget Line key and Budget Bucket ID.

Problem Resolution

If the process was discontinued for any reason then the job has the ability to be restarted from the point it left off only if it was run in the Incremental Mode.

2.1.12 Systems Assurance 12 – Budgeting Incremental Updates Systems Assurance (SA12)

The Systems Assurance 12 Budgeting Incremental Updates Systems Assurance shows the differences between the Budget Tables and the Budget Constraint Amount table.

Description

SA12 generates a report to show the differences between the Budget Tables and the Budget Constraint Amount table. The report generated by this process does not include a Diagnostic section, so in order to fix the problem that may occur, the user needs to run the Recalculate Budget Control Amounts Batch job. In this process references to Budget lines include both budget and Allotment lines.

The SA12 Chain Job has three main parts:

- SA12 Pre-Processor - This job is responsible for dividing up the work that the job will do and restarting any previously failed jobs.
- SA12 Update - This job updates the Budget Constraint Amount Shadow table.
- SA12 Report - This job generates a report to show inconsistencies between the actual Budget Constraint Amount table and the SA12 Budget Constraint Amount Shadow table. Comparisons logic in this job is mimicked from the Recalculate Budget Control Amounts Batch job logic.

There are three supporting tables for SA12. (Note that there are no online pages for these tables)

- **SA12 Workload Table**: - This table will provide the list of the all the work needs to done.
- **SA12 Facilitator Table**: - This table will hold the blocks of workload records such that multiple blocks can be processed in parallel.
- **SA12 Control Check Table**: - This table will be populated by the Preprocessor job by selecting the active system wide constraints and storing the descriptive information in a single place.

In case of invalid parameters or Dirty Flag errors, the job will be aborted before the Preprocessor is initialized. In case of Dirty Flag errors, the user will be instructed to clear those errors with the Recalculate Budget Control Amounts job before running the SA12 job.

When to Run

When you need to validate that the values on a Budget Line and the Budget Constraint Amount table are the same.

Major Input

All budget tables (for example, bud_stru_31_lvl_1 for structure 31 level 1). Excluding the Activity Tables.

- All allotment tables (for example: alot_stru_26_lvl_1 for structure 26 level 1). Excluding the Activity Tables.
- Control tables.

Output

- SA12 Workload Table
- SA12 Facilitator Table
- SA12 Control Check Table
- SA12 Budget Constraint Amount Shadow Table

SA12 Pre-Processor

Description (Caption)	Parameter Name	Default Value
Parameter File Location for SA12.	AMSPARM	
Budget Fiscal Year	BFY	9999 will default and always be evaluated because of budget structures that do not have a BFY.
Budget Structure	Structure	
Budget Structure Level	Level	
Run Mode: 1=Update Only, 2=Report Only, 3=Update and Report	Run Mode	3
Assure allotments. Valid values (Y/N).	Check Allotments	Y
Constraint ID	Constraint ID	
Restart	Restart	N
Run ID	Run ID	
Parameter File Name	FILE_NM	SA12ParmFile.txt
Block Size	Block Size	1000

SA12 Update

Description (Caption)	Parameter Name	Default Value
Parameter File Location for SA12.	AMSPARM	
Budget Fiscal Year	BFY	9999 will default and always be evaluated because of budget structures that do not have a BFY.
Budget Structure	Structure	
Budget Structure Level	Level	

Run Mode: 1=Update Only, 2=Report Only, 3=Update and Report	Run Mode	3
Assure allotments. Valid values (Y/N).	Check Allotments	Y
Constraint ID	Constraint ID	
Run ID	Run ID	
Parameter File Name	FILE_NM	SA12ParmFile.txt

SA12 Report

Description (Caption)	Parameter Name	Default Value
Parameter File Location for SA12.	AMSPARM	
Budget Fiscal Year	BFY	9999 will default and always be evaluated because of budget structures that do not have a BFY.
Budget Structure	Structure	
Budget Structure Level	Level	
Client Name appearing on report	Client Name	
Run Mode: 1=Update Only, 2=Report Only, 3=Update and Report	Run Mode	3
Assure allotments. Valid values (Y/N).	Check Allotments	Y
Constraint ID	Constraint ID	
Run ID	Run ID	
Parameter File Name	FILE_NM	SA12ParmFile.txt

- The Parameter File Location for SA12 indicates the path where parameter files need to be created.
- Budget Fiscal Year (BFY) indicates the budget fiscal year for which the report should be executed. It limits the number of budget records being verified. All budget records that do not have a BFY in their key are always selected. More than one BFY can be specified by separating the values with commas. BFY 9999 can be specified, but if not the process will always assume that year. This is a required field.
- Budget Structure indicates the structure for which the report should be executed. It limits the number of structures being verified. More than one structure can be specified by separating the values with commas. If no structures are specified, the system will use all active structures.

- Budget Structure Level indicates the level for which the report should be executed. It limits the number of levels being verified. One level can be specified to correspond to each structure specified. If no levels are specified, the system will use all active (non-audit) levels for each structure specified.
- Client Name to appear on the report. The name appears as the first line in the header of the report. The system wide default for Client Name will be used when established.
- Check Allotments may be set to Y to verify any allotment periods defined for the structure(s).
- If Run Mode 1 is chosen, the Budget Constraint Amount Shadow table is updated but no Report is generated. If Run Mode 2 is chosen, then a report is generated from the data already saved in the Budget Constraint Amount table and Budget Constraint Amount Shadow table. If Run Mode 3 is chosen, all of the calculations to generate the report are performed and the report is generated with those calculations.
- Constraint ID is user specified in the event that the user only wants certain constraints to be checked. If this parameter is left blank, all constraint IDs are checked. The Constraint ID appears in the header of the report on the Parameters line.
- The Restart parameter is used when a part of the preprocessor fails and the user wants to only process the failed part. If not entered, the Restart parameter defaults to No.
- Run ID can be user entered or system generated. If restarting the preprocessor, the Run ID must be identical to the Run ID of the previous run. If Restart = No, then the Run ID may be system generated if the user leaves it blank.
- Parameter File Name is a means to create and pass a file to subsequent jobs for parameter consistency.
- The SA12 Pre-Processor job writes values for several parameters (also specified in that first job) to that file. Those parameters written into the file are BFY, Structure, Level, Constraint ID, Check Allotments, Restart, Run ID, Run Mode, and Block Size. If a Parameter File Name is provided for subsequent job steps, the file name should be same as that for the first job. When the Parameter File Name parameter is specified in the remaining job steps, any value specified in these nine parameters is ignored and what is in the file is used.

  While scheduling the chain job if user disables the SA12 Pre-Processor job, no Parameter File is created. Therefore, if the user is providing the Parameter File Name for the other two jobs then he has to make sure that there is a file existing in the same location and with the valid parameter values for the other parameters. The alternative is not to provide any Parameter File Name for the latter two jobs and provide the values for all other parameters while scheduling the job itself by editing job parameters.
- The Block Size parameter in the pre-processor job step is used to indicate the number of records to be processed in one block. The default value of 1000 is just a suggestion. The Pre-processor divides the budget and allotment line records with this block size to create manageable blocks of work that are passed to the update job step to write in SA12_BUD_CNST_SHDW. Adjust the Block Size if 1000 is too large for the given infrastructure and number of records. As a note, each block contains only one budget structure and level, so if there are not enough records to fill the block, then the job begins a new block for a new structure and level.

If there are any errors that are triggered for the parameters or if there are any dirty errors before the Preprocessor is initialized, the job will be aborted. In the case of dirty errors, the user will be instructed to clear those errors before running the SA12 job with the Recalculate Budget Control Amounts job.

Sort Sequence

Sorted on the basis of the UNID and ALOT_FL on the shadow table.

Selection Criteria

All three jobs will select records for particular BFY, Structure IDs, Level IDs and Constraint IDs if entered on the Parameter page or for all active Structures, Levels (Excluding Activity Level) and Constraint IDs in the system. If the Check Allotment flag is selected then Allotment lines are also considered.

The Parameter File is created in the SA12 Preprocessor step which can be used for selection by the SA12 Update and SA12 Report step.

While reporting, the process will generate the report based on Structure, Level, COA (which includes BFY), Allotment and Budget Line.

Problem Resolution

Either using the chain job or the individual steps, the user needs to enter the exact same parameter value set for a contiguous batch job run.

2.1.13 Systems Assurance 13 – Out of Sync differences between Financial and VSS

Batch Job Name	System Assurance 13
Recommended Frequency	On Demand
Single Instance Required	No, multiple instances of the job can be run in parallel.
Can be restarted?	No
Reports generated	Yes

Overview

The System Assurance 13 (SA13) job will be used to report on all out of sync differences between Financial and VSS tables. This batch job will be run from Financial and will consist of the following steps:

- Validate job parameters - In this step, the job validates the input parameters defined as parameters to the batch process.
- Creating a Column List - In this step, the job gets all the columns using the application metadata and stores it temporarily.
- Exclude Column List – In this step, the job excludes all the columns mentioned in the Exclude Parameter File List.
- Dynamic SQL – The batch job uses the table and column information and creates a dynamic SQL to identify the following conditions:
    - Record exists in VSS but does not exist in Financial.
    - Record exists in Financial but does not exist in VSS.
    - Field mismatches between the two systems.
- Report Generation – The batch job finally reports the out of sync conditions identified in the above steps.

  The report will break on the following subsections:

    - Financial Record Not Found in VSS: This section displays individual records found in CGI Advantage Financial without a corresponding record in VSS.
    - VSS Record Not Found in Financial: This section displays individual records found in VSS without a corresponding record in Financial.
    - Fields Mismatch: This section includes those records that are found in either applications but have one or more fields that do not match.

For each database table appearing in the report, there will be summary statistics on the number of records processed and discrepancies found. At the end of the report, a summary section will provide statistics on the total number of records processed and discrepancies found (i.e. across all tables verified by the SA13 process).

Note: The SA13 process should be run ONLY after ALL of the Export and Load ('synchronization') jobs have been run in both Financial and VSS. For example, the Financial jobs include: Export from Advantage, Tracking Table Export, Load to Advantage, and others. The VSS jobs include: Load to VSS, Export from VSS, Export DPC Table, DPC Load, and others. If users have pending data changes in each database that have not been synced across the two databases, the SA13 job will report those records as out of sync.

In order for the SA13 process to run, the tables from VSS on which the process needs to perform system assurance should be accessible by the Financial Application job managers. The database schema that would have those VSS tables or the database link that would enable access to that VSS schema is specified in the VSSSchemaForSA13 parameter in the System Configuration ADV30Params.ini file.

As database link connection enables access between two databases, so if database setup is such that the Financial schema and VSS schemas are on different databases, a database link might be required to establish the connection between them, before running the SA13 job.

Refer to the "How to configure Financial and VSS schemas for SA13" section below, before running this job.

Encrypted fields will not be able to be included in the system assurance process. These fields can only be compared if the same key store is used to encrypt data between Financial and VSS.

Basically, the process flow should be as follows:

1. Run the synchronization jobs to sync the data between Financial and VSS.
2. Refresh the VSS tables into a schema that is accessible by the Financial Application job managers.
3. Run SA13 to identify and report on out of sync conditions between VSS and Financial.

When to Run

User-defined frequency

Major Input

Job Parameters: (List of tables, Exclude Attributes)

Major Output

This batch job generates an Out of Sync report.

Parameters

The following job parameters are listed below.

Parameter	Description	Default Value
1st List of Tables to be Assured	Required 1st List of Tables to be Assured	R_AD; R_AMHIST; R_BUS_TYP; R_CNTAC; R_COMM_CD; R_CTRY; R_CTY; R_DEPT; R_EV_PRICE_DT; R_EXP_SOPT; R_EXT_EMAIL_LTRCNF;

		R_GEN_SOPT; R_MSTR_AD; R_PNT_VEND_CUST; R_SO_QA; R_TIN_1099_INFO; R_UNIT;
2nd List of Tables to be Assured	Optional 2nd List of Tables to be Assured.	R_UNIT_MEAS; R_VEND_ABA_MAINT; R_VEND_BUS_TYP; R_VEND_CERT; R_VEND_CNTAC; R_VEND_COMM; R_VEND_CUST; R_VEND_SRVC_AREA; R_VEND_USER; R_VSS_USER; MA_CTLG; MA_CTLG_PIC_ATT; R_VEND_W8_FORM; R_W8_FORM_TYP; R_W8_FORM_PRT; R_W8_LST; R_W8_FORM_FLD
File Name having Semi-colon delimited list of Excluded Attributes that will not be used for data comparison	Optional The column name can be put as TBL_LAST_DT;AMS_ROW_VERS_NO for columns to be excluded from all the tables or R_AD.PAGE_CD to be excluded from a specific table.	SA13_Exclude_Param.txt
Client Name	Optional. Client Name appearing on report.	No Default value
AMSPARM	$$AMSROOT$$/Parms	Parameter Location for SA13 Job. The exclude file ***SA13_Exclude_Param.txt*** shall be stored in this location.

Sort Sequence

NA

Selection Criteria

NA

Problem Resolution

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

How to configure Financial and VSS schemas for SA13

The following sample steps and SQL statements can be followed to create DBLinks between Financial and VSS databases.

1. In the steps detailed below, perform the following replacements based on the actual values being used during setup:

   Replace <<VSS_DB_LINK>> with standard DB Link name being used at site.

   Replace <<SYSTEM_PASSWORD>> with system user password in the Financial Database.

   Replace <<VSS_USER>> with the VSS user id from the VSS database

   Replace <<VSS_USER_PASSWORD>> with the VSS user id from the VSS database

   Replace <<SERVER_NAME>> with VSS Database server name

   Replace <<PORT_NUMBER>> with VSS Database Port number, example 1521

   Replace <<SID_NAME>> with VSS Database Sid name

   Replace <<VSS_DBLINK_USER>> VSS Database Link user Id

   Replace <<VSS_DBLINK_USER_PASSWORD>> VSS Database Link user password

   Replace <<FIN_TABLESPACE>> with the financial tablespace name.

   Replace <<FIN_DB_NAME>> with the TNS entry for the Financial database name

2. Create a Database Link (VSS_DB_LINK) in Financial database to access VSS database.

   Provided below is a sample command to create a DBLink, which should be run as a system or a database user with proper privileges.

   ```
   CONNECT SYSTEM/<<SYSTEM_PASSWORD>>@<<FIN_DB_NAME>>


   CREATE PUBLIC DATABASE LINK <<VSS_DB_LINK>> CONNECT TO <<VSS_USER>>
   IDENTIFIED BY <<VSS_USER_PASSWORD>> USING
   '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<<SERVER_NAME>>)(PORT=<<PORT_N
   UMBER>>))(CONNECT_DATA=(SID=<<SID_NAME>>)))';
   ```

3. On Financial database create a VSS Database Link user Id (VSS_DBLINK_USER) to access the tables in VSS database

   ```
   CREATE USER <<VSS_DBLINK_USER>>

   IDENTIFIED BY <<VSS_DBLINK_USER_PASSWORD>>

   DEFAULT TABLESPACE <<FIN_TABLESPACE>>

   TEMPORARY TABLESPACE TEMP

   PROFILE DEFAULT

   ACCOUNT UNLOCK;
   ```

4. Grant privileges and roles for VSS Database Link user (VSS_DBLINK_USER) created in step above.

GRANT RESOURCE TO <<VSS_DBLINK_USER>>;

GRANT CONNECT TO <<VSS_DBLINK_USER>>;

GRANT CREATE SYNONYM TO <<VSS_DBLINK_USER>>;

ALTER USER <<VSS_DBLINK_USER>> DEFAULT ROLE ALL;

GRANT CREATE SESSION TO <<VSS_DBLINK_USER>>;

5. Login to the Financial Database/schema as user (VSS_DBLINK_USER created above), and create synonyms for the following tables.

   These tables are those VSS tables that Financial schema needs access to perform lookups.

   Connect <VSS_DBLINK_USER>>/<<VSS_DBLINK_USER_PASSWORD>>@<<FIN_DB_NAME>>

   CREATE SYNONYM R_AD FOR R_AD@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_BUS_TYP FOR R_BUS_TYP@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_CNTAC FOR R_CNTAC@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_CTRY FOR R_CTRY@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_CTY FOR R_CTY@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_DEPT FOR R_DEPT@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_EV_PRICE_DT FOR R_EV_PRICE_DT@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_EXP_SOPT FOR R_EXP_SOPT@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_EXT_EMAIL_LTRCNF FOR
   R_EXT_EMAIL_LTRCNF@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_GEN_SOPT FOR R_GEN_SOPT@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_MSTR_AD FOR R_MSTR_AD@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_PNT_VEND_CUST FOR R_PNT_VEND_CUST@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_SO_QA FOR R_SO_QA@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_TIN_1099_INFO FOR R_TIN_1099_INFO@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_VEND_ABA_MAINT FOR R_VEND_ABA_MAINT@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_VEND_CERT FOR R_VEND_CERT@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_VEND_CNTAC FOR R_VEND_CNTAC@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_VEND_CUST FOR R_VEND_CUST@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_VEND_SRVC_AREA FOR R_VEND_SRVC_AREA@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_VEND_USER FOR R_VEND_USER@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_VSS_USER FOR R_VSS_USER@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_AMHIST FOR R_AMHIST@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_COMM_CD FOR R_COMM_CD@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_UNIT FOR R_UNIT@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_UNIT_MEAS FOR R_UNIT_MEAS@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_VEND_BUS_TYP FOR R_VEND_BUS_TYP@<<VSS_DB_LINK>>;

   CREATE SYNONYM R_VEND_COMM FOR R_VEND_COMM@<<VSS_DB_LINK>>;

   CREATE SYNONYM MA_CTLG FOR MA_CTLG@<<VSS_DB_LINK>>;

CREATE SYNONYM MA_CTLG_PIC_ATT FOR MA_CTLG_PIC_ATT@<<VSS_DB_LINK>>;

CREATE SYNONYM R_VEND_W8_FORM FOR R_VEND_W8_FORM@<<VSS_DB_LINK>>;

CREATE SYNONYM R_W8_FORM_TYP FOR R_W8_FORM_TYP@<<VSS_DB_LINK>>;

CREATE SYNONYM R_W8_FORM_PRT FOR R_W8_FORM_PRT@<<VSS_DB_LINK>>;

CREATE SYNONYM R_W8_LST FOR R_W8_LST@<<VSS_DB_LINK>>;

6. CREATE SYNONYM R_W8_FORM_FLD FOR R_W8_FORM_FLD@<<VSS_DB_LINK>>;Provide the <<VSS_DBLINK_USER>> value for 'VSSSchemaForSA13' parameter in the ADV30Params.ini.

2.1.14 System Assurance 15 – Open Amounts

Batch Job Name	System Assurance 15
Recommended Frequency	On Demand
Single Instance Required	No, multiple instances of the job can be run in parallel.
Can be restarted?	No
Reports generated	Yes

Overview

The Systems Assurance 15 process verifies that at the budget line level, one or more budget amounts (defined by parameters) are in sync with the total open amounts on accounting transactions that match that budget line. If the values on the transactions are out of sync with budget amounts recorded on all budget tables, then budget amounts will not accurately reflect the amounts open (or closed). A report generated by the job will display the values on the budget, the open amounts on the related transactions and the amount posted to the journal. In some cases, certain journal vouchers are used to establish or maintain a budget amount. This process allows those journal vouchers to be included when reporting out of sync conditions. This process will not correct the out of sync conditions. That has to be done with a journal voucher. Please see the Problem Resolution for more details on that procedure.

The following two configuration options will result in new COA being inferred to the Posting Line that are not on the original accounting line; therefore, Out of Synch conditions will be reported on the SA15 report:

- When using the Front End Split (FES) capability to infer new COA onto the Posting Lines.
- When running the SA15 report against a Reimbursable Budget Structure that uses Funding Profiles to infer new COA to the Posting Lines.

The tables used as part of the comparison of open amounts are, the Common Accounting Line Catalog (DOC_ACTG), a budget table (BUD_STRU_##_LVL##) and the Accounting Journal (JRNL_ACTG). The sum of open amounts from DOC_ACTG adjusted by the net debit or credit posted to (JRNL_ACTG) by included journal vouchers is compared with the budget amounts on the specified budget table. If these amount match, that budget amount is in sync with all open accounting transactions.

If the records are not in sync and a detailed listing is requested (via a job parameter), the system will again reference the DOC_ACTG and JRNL_ACTG tables for reporting purposes. The open amount of each DOC_ACTG record that matches the out of sync budget line is compared to the sum of postings by (each version) and against (by references) the accounting line found in JRNL_ACTG. Out of sync records are displayed in the main section as well as the detailed section. Note: The detailed section displays the open amount and posting amount for only those accounting lines where the open amount calculated from the accounting journal is different from the accounting line's open amount. Included journal vouchers will never appear in the detailed listing because:

- Journal vouchers do not have an open amount separate from their posting amount.
- Journal Vouchers can only be memo referenced.

Prior system experiences with this type of out of sync condition have revealed that if one budget level updated by an accounting transaction becomes out of sync with the open amount of that accounting line, then all budget structures and levels updated by that accounting line will be out of sync. For this reason, assuring only 1 spending budget level and perhaps 1 revenue budget level

is necessary. To enhance the performance time of this program, the highest budget level should be assured (i.e. level 1 of structure 29 instead of level 2) because it has fewer budget lines.

Prior to the running of this job, the following should be considered.

- Prior to running this job, to ensure the integrity of the SA15 report, System Assurance 01 should be run. Any existing out of sync conditions documented on the SA01 report should either be corrected or noted as they may also be reflected in the SA15 report.
- When CGI Advantage transaction archiving functionality is being used, one should verify that the final latest version of transactions being assured for the budget structure and level, have not been archived. If these transactions are archived and an out of sync condition is encountered for that budget, the report may not accurately reflect the activity for the budget line. To mitigate this, only current budget lines should be assured and transaction archiving should only be done against older, no longer assured, budget years.
- When CGI Advantage journal archiving functionality has been used on the Accounting Journal, one must verify that the System Assurance 15 job is not run for a Budget Fiscal Year that could have been updated by a Fiscal Year archived from the Accounting Journal.
- Budget Structures selected should not be reimbursable as Chart of Accounts for their definition (Funding Profile, Priority, and Line) is not present on the DOC_ACTG.
- Allotments are not assured in this report.
- If the Bucket ID supplied has requested that the Pending Increase or Decrease Buckets be include (BUDTAM flags), be aware that the amount being assured is net of those pending amounts.
- To enable efficient processing for this job, it is suggested that several indices be added to the database based on the budget structure and level(s) being assured. Two tables, the Accounting Journal and the Common Transaction Accounting Line Catalog, should have indices if testing of this report proves to be slower than desired. Please see the later section below entitled “Indices to Improve Performance”.
- Assurance for a budget amount updated by internal transaction processing (IET, ITA, OC, etc) is not advised because such accounting comes from transactions where one party is defined outside of the accounting line. The Exchange Details section of the IET or the Warehouse used by an OC is two examples. Amounts in this category include Accrued Expenditures, Cash Expenditures, Collected Revenue, and Billed Revenue. Two of these four are never referenced and budget amounts not liquidated do not present the opportunity to become out of sync with open amounts. Assurances against the other two can still be done and those budget lines used for internal accounting can be overlooked on the report if assurances are desired against the other types of budget lines.

The various steps involved in the process are – Parameter Validation, Selection of Records and Reporting.

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 is complete.
2. Selection of Records	• Fetching of records started. • If the selection returns 0 records, then the following message

Process Steps	Messages
	will be issued: “All records are in sync.” • Number of records processed is n: (where n is a count of budget lines and accounting line read that are out of sync).

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

- All active budget tables (for example, BUD_STRU_31_LVL_1 for structure 31 level 1)
- Common Transaction Accounting Line Catalog – DOC_ACTG
- Budget Structure - GN_BUD_STRU
- Budget Bucket – R_GN_BKT
- Budget Level Control – GN_BUD_LVL
- Posting Code – R_PSCD
- Accounting Journal – JRNL_ACTG
- Event Type – R_EVNT_TYP
- Fiscal Year – R_FY

Batch Parameters

Parameter Name	Description	Default Value
BFY (Budget Fiscal Year)	Enter active BFY’s separated by commas and 9999 if multi-years are used for structure being assured. *If left blank, all budget lines will be selected.* *Not required for structures without BFY.* *Not required and ignored if Budget Line UNID is entered.* *Value entered must be valid on the R_FY table and should have at least 1 corresponding line defined on the structure and level.*	blank
STRU_ID (Budget Structure)	Budget Structure *Required parameter to identify a single structure to assure.* *Must be an active budget structure and should not be one that is Budget Type = Reimbursable.*	blank

Parameter Name	Description	Default Value
STRU_LVL_ID (Budget Structure Level)	Budget Structure Level *Required parameter to identify a single structure to assure.* *Must be a valid level for the structure and not the activity/audit level.*	blank
BUD_BKT_ID (Budget Bucket ID)	Budget Bucket ID *Required parameter to identify one or more budget amounts to assure.* *Must be valid on BUDTAM and not a calculated or pending amount.*	12,13
UNID (Budget Line UNID)	Budget Line UNID *Optional parameter that allows the assurance to be done against only one budget line.* *Only 1 value is allowed at a time.* *Often used when a budget line was out of sync and has had a correction processed against it.*	blank
PROG_CTR_SZ (Progression Counter Size)	Progression Counter Size *Optional parameter for processing counts.* *Will default to 1000 if left blank.*	1000
CLIENT_NM (Client Name)	Client Name Appearing on Report *Optional field.*	Blank
COMMIT_BLOCK (Commit Block Size)	Commit Block Size *Optional parameter for performance control.* *Will default to 1000 if left blank.*	1000
SELECT_BLOCK (Select Block Size)	Select Block Size *Optional parameter for performance control.* *Will default to 1000 if left blank.*	1000
DET_TRANS_LST (Detailed Transaction Listing)	Detailed Transaction Listing: Detailed (1), Not Detailed (2) *A required parameter that when set to 1 will enable the report listing of each accounting line mapped to a budget line found not to be in sync.*	2

Parameter Name	Description	Default Value
INCL_JV_DOC_CDS (Included Journal Voucher Transaction Codes)	Included Journal Voucher Transaction Codes: Optional parameter used to select journal vouchers to be included when determining whether a budget line is out of sync. May include one or more Journal Voucher Transaction Code (comma-delimited). By default no journal vouchers are included in the out of sync calculation	Blank

Major Output

- Systems Assurance for Open Amounts Report

Job Return Code

The following table shows the potential job return codes for the System Assurance 15 job.

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 job does not end with this status. Where normally a job would end as warning if no records were found to match selection criteria, the nature of this job does not allow for this result. Instead, if valid selection criteria are entered but no records are found to match, the job will end as successful and the report will list all records were in sync. Please enter selection criteria carefully.
Non Fatal Error (8)	The job status will never be Non Fatal Error. Budget line out of sync with open amounts on accounting lines
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

N/A

Selection Criteria

From BUD_STRU_##_LVL##: If BFY parameter is supplied but not UNID, BFY is used to select lines from the specified structure and level. If UNID parameter is supplied, the single line identified by that UNID is selected from the specified structure and level. The amount or amounts selected is determined from the Budget Bucket parameter.

From DOC_ACTG: Selection uses the full key to each budget line with the addition of each event type ID found to contain a posting code that updates the amount(s) identified in the Budget Bucket parameter.

From JRNL_ACTG: When Included Journal Voucher Transaction Codes provides JV transaction code(s), selection uses a combination of the transaction identification information (Transaction Code, Department, ID, Version, VL, CL, and AL) from selected DOC_ACTG records with the addition of all posting codes found that update the Budget Bucket parameter(s).

When Detailed Transaction List is set to Detailed(1), selection again uses a combination of the transaction identification information (Transaction Code, Department, ID, Version, VL, CL, and AL) from selected DOC_ACTG records with the addition of all posting codes found that update the Budget Bucket parameter(s). To get all the postings from that accounting line is the first task. The second is to take that same transaction information and look in the referenced transaction information fields of journal records. This second set of amount is the total liquidations against the accounting line. Together, these two amounts form a ‘balance’ of that accounting line.

Indices to Improve Performance

This section discusses some technical aspects that will require interaction with a technical resource that is familiar with the Advantage Financial application as well as a DBA familiar with your environment. This report, as listed inside of the “Selection Criteria” section, has two critical queries that are being performed:

1. One query performs a join between the Budget Structure table, being requested via the Input Parameters, the Transaction Accounting Line Catalog [DOC_ACTG], and (if Journal Voucher Transaction Codes are included) the Accounting Journal [JRNL_ACTG]. This identifies Out of Synch [OOS] Budget Lines.
2. The second query, which will only be processed when the report is in “Detailed” mode, is a join between the Transaction Accounting Line Catalog [DOC_ACTG] and the Accounting Journal [JRNL_ACTG].This tries to identify the Out Of Sync Transaction Accounting Line Catalog records which may be causing the particular Out of Synch Budget Line.

Each of these three tables may be very large in your application. When trying to join these tables together for each of the queries, it may be very costly to perform this operation. To assist in reducing the processing time, it is suggested that at least two new indices be added into your database if this job will be run on a regular basis. If this job is run on a very sporadic basis, it may be worthwhile to create the indices before running and then dropping the indices after completion of the runs. This strategy should be decided based upon the frequency for running the job and the cost of building and dropping the Indices, relative to the daily cost incurred by every transaction processing, by retaining these indices.

There is one very specific index that should be created to assist in the query for identifying OOS Transaction Accounting Line Catalog records. This index would cover the Referenced Fields that exist on the Transaction Accounting Line Catalog, and are used as part of the winnowing criteria for this query. The index would be a non-unique index on the Accounting Journal [JRNL_ACTG]. The fields to be used should be:

1. RFED_DOC_CD
2. RFED_DOC_DEPT_CD

3. RFED_DOC_ID
4. RFED_VEND_LN_NO
5. RFED_COMM_LN_NO
6. RFED_ACTG_LN_NO

The basic syntax for an index if it was to be created in your environment and it is Oracle would be akin to the following:

```
CREATE  INDEX NSA15_JRNL_ACTG
 ON JRNL_ACTG
 ( RFED_DOC_CD,
   RFED_DOC_DEPT_CD,
   RFED_DOC_ID,
   RFED_VEND_LN_NO,
   RFED_COMM_LN_NO,
   RFED_ACTG_LN_NO  )
```

Please note that the name of the index is not in conformance with the Baseline naming scheme, however, it is important that it be named in a different manner than Baseline, so that we do not overlap a possible future new index with the name chosen at your site. Also, you should maintain conformance with the name of this index for your site, so that you do not break your site's processes and procedures. The name above is a sample and may not be applicable at any individual site.

The one that is harder to define is the index to assist in finding the Transaction Accounting records which may have affected a Budget Structure's Level. Since there are many different Structures and Levels and there are also custom Structures and Levels, this can only give you an idea as to what index (or indices) should be built to help the performance in identifying the OOS Budget Lines. The Functional Representative must decide which Structure(s) and Level for which this report will be processed. Once that is decided, the index (ices) fall from the Primary Key of the Budget Structure and Level decided.

As an example, let us say that your Functional Representative has defined that of the Budget Structures that are used by your site, which also meet with the limiting criteria here as to which Structures should be used, they decide that they will run this report against Budget Structure 80 [chosen here since it is a baseline structure] Level 3. Since Level 4 is an optional level, it makes best sense to run this report against Level 3 since we know if a Transaction processed it would either hit this Level, along with the parent Levels 2 and 1, or it did not hit any Levels in this structure.

Now the Technical Representative would perform the following basic steps:

1. Identify the table in the database that retains the data for Structure 80 Level 3. The name of this table is BUD_STRU_80_LVL_3.
2. The next step would be to identify the Primary Key of this table (primary key being database nomenclature for the fields that would uniquely identify one record inside of this table). The primary key for this table is:
    a. BFY
    b. FUND_CD
    c. DEPT_CD

d. APPR_CD
e. UNIT_CD
f. OBJ_CD
g. ACTV_CD

3. The next step is to either directly create Index SQL that is compliant with the database at your site, or request the DBAs either in part or in whole create the Index SQL. This SQL would:
    a. Reside on the Transaction Accounting Line Catalog table [DOC_ACTG]
    b. It would be a Non-Unique Index [meaning that the fields here do not uniquely identify just one record, since there could be thousands of Accounting Lines from transactions processed that affected only one Budget Line in Structure 80 Level 3]
    c. The name again would be in compliance with the sites naming standards, and would ideally not mimic the Baseline model, again to ensure we do not create a future baseline index that is named the same as this custom index.
    d. The SQL would look something like this if your site ran Oracle:

```
CREATE  INDEX NSA15_DOCACTG_BUD_80_3
 ON DOC_ACTG
 ( BFY,
   FUND_CD,
   DEPT_CD,
   APPR_CD,
   UNIT_CD,
   OBJ_CD,
   ACTV_CD  )
```

4. If the Functional Representative wanted to run this report against a different structure, then again the Technical Representative would walk through the above steps.

If the Functional Representative wanted to run this report against different Levels of the same structure (not sure if this meets the basic criteria as to what structures and levels this report should use), the idea would be to identify the child most required level of the structure and create an index for that level. If the functional person is requesting that Optional levels be used, I believe that the one index on the child most required level should suffice, however, a test may be in order to ensure another index is not required for the optional level(s) as well.

Problem Resolution

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

If the job ends with the return code of Non Fatal, then the out of sync condition needs to be investigated. When an accounting line is found to have an open amount not in sync with Accounting Journal postings, it is best to stop any further activity against that accounting line. Normal modifications to the accounting line or references to it as part of normal business activity may cause additional incorrect postings. At a minimum, the accounting line should have the Line

Amount modified down to equal the Closed Amount. This will cause an additional posting, but running the assurance again after just the single budget line UNID will include that posting with the others to determine the total incorrect amount of postings. A secondary step that some may take is to put text in the Accounting Line Description field with strong language warning against reopening this line with a modification. A final precaution would be to add a new COA element on the accounting line that will have an Effective To Date in the past to prevent accidental modifications by an interface or users.

Running the SA01 report before making any corrective adjustment is a critical step. If that report reveals that the budget line is correct but what has been posted to the Accounting Journal is incorrect (transaction never posted to the journal for example), then the following corrective actions for SA15 issues should not be performed. Corrective actions for the SA01 issue should take place first and then another SA15 run.

Running SA06 for the transaction code of the SA15 record is another step that should be performed before any SA15 correction. If there are posting lines from any version of that accounting line that have not posted, SA15 will not have all the necessary Accounting Journal data to compare to the Open Amount of the accounting line. In this same area, transactions with posting lines that have not been journalized will result in the Accounting Journal possibly not having enough data to compare postings to the open amount of an accounting line. Steps such as running the Journal/Posting Initiator and the Journal Engine should be taken to ensure the journal is up to date.

The report produced will look like the one below, when records are found to be out of sync. The second detail section will only be produced if specified in the job parameters. In the header section are the selection criteria used. Beneath that is a listing of each budget line, by unique ID (UNID), followed by a concatenation of the key fields of that budget line. For each budget line the report will show the amount found on the budget table in the Budget Amount column and the total Open Amount of all accounting lines found that were found to match that budget line and budget bucket.

The second detail section of the report shows individual accounting line details. First is a complete identification of the accounting line followed by a column for the Open Amount found on that accounting line. The Journal Amount column will contain a summary amount of all postings by that accounting line and liquidations against that accounting line.

```
RUN DATE:  01-18-2007
RUN TIME:  01:44:18                                                    PAGE: 1
REPORT ID: SA15                    Systems Assurance for Open Amounts

Budget Structure:   29
Budget Level:       1
BFY:                2006

 Encumbered        ( 13)

                                                          Budget          Open
UNID            Line                                      Amount          Amount
----------      --------------------------------      ----------------  ----------------

Accounting Line                                             Open          Journal
DOC,DEPT,ID,VERS,VL,CL,AL                                  Amount          Amount
-------------------------------------------------       ----------------  ----------------
```

Correction procedure for SA15 out of sync condition:

1. Run SA15 against UNID of budget line in Detail Transaction Listing Mode

2. Close out accounting lines listed where the Open Amount (AL one and not budget line Open Amount) does not equal the Journal Amount. Those accounting lines with a Journal Amount equal to the Open Amount are not a problem and should be left alone.
3. Run SA15 against UNID of budget line in Detail Transaction Listing Mode
4. With the accounting line closed out, the Open Amount value for an individual accounting line will always be $0. Evaluation now turns to the Journal Amount.
    a. If that amount is now also $0, there is no adjustment required
    b. If that amount is positive and this is a spending amount, then there is a debit balance against the line – under liquidation. A journal voucher is needed to credit the posting code used by the accounting line that updates the budget bucket and debit the offset used by the accounting line.
    c. If that amount is negative and this is a spending amount, then there is a credit balance against the line – over liquidation. A journal voucher is needed to debit the posting code used by the accounting line that updates the budget bucket and credit the offset used by the accounting line.
    d. If that amount is positive and this is a revenue amount, then there is a debit balance against the line – over liquidation. A journal voucher is needed to credit the posting code used by the accounting line that updates the budget bucket and debit the offset used by the accounting line.
    e. If that amount is negative and this is a revenue amount, then there is a credit balance against the line – under liquidation. A journal voucher is needed to debit the posting code used by the accounting line that updates the budget bucket and credit the offset used by the accounting line.
4. Run SA15 against UNID of budget line to see the Budget Amount now equals the Open Amount of all accounting lines. If so, the report will list all records as in sync.

    Note: in creating correction journal vouchers, some may choose to perform a memo reference to the accounting line that was out of sync. This may require adding a record to the Transaction Allowable References (DARF) table. The level of Chart of Account detail on the correcting journal vouchers should be at the same level found on the accounting line that was out of sync to ensure any other budgets updated by the accounting line also get updated. In the case of wanting to update a reimbursable budget structure, the JVC transaction code will have to be used.

2.1.15 System Assurance 16 – Missing Posting Lines

Job Name	System Assurance 16
Recommended Frequency	Weekly, Daily, or On Demand The more frequent the shorter the run time and corrections are more timely
Parallel processing enabled	Yes
Can be restarted?	Yes – for parameter validation errors only
Reports generated	Yes

Overview

The majority of accounting lines processed should have at least one posting line generated. The Accounting Configuration subsection of the General Accounting Users Guide contains information about how and when posting lines are generated. There are several exceptions to the rule of having at least one posting line for every accounting line:

1. Transactions with accounting lines that do not create posting lines by design. The Internal Transaction Initiator.
2. Accounting lines introduced on a transaction version with a line amount of $0.00.
3. Event type and posting pair combinations without posting codes on an accounting line without a real reference (partial, final, or inverse) to another accounting line.

This batch job generates two reports of accounting lines with no posting lines where a transaction has reached the final and historical final transaction phase. The first report displays transactions that have transaction function of new or cancellation. The second report displays transactions that have transaction function of modification.

As the report reads from the common Accounting Line Catalog, where all transactions store accounting lines, selection uses two important selection criteria for performance. The first is a Transaction Last Date which is used to select records on a 'greater than or equal to' basis. When running the assurance for the first time, the Transaction Last Date parameter should not be used as the program should evaluate all existing records. After the first run, the parameter can be used and incremented to prevent subsequent runs from evaluating what has already been assured. The second parameter is the list of excluded transaction types. As delivered, the parameter includes all transaction types that do not have accounting lines, or have them but they never should have posting lines. Use of this parameter allows for multiple instances of this report program to process concurrently. By having one instance that excludes a subset of transaction types that should have posting lines and another instance excluding the subset that the first is checking, the Accounting Line Catalog can be processed faster.

A report is always generated if parameter validation is successful. If there are no records found for a given report, then a single line will be written to the detail section of the report stating, 'No Records Found.'

Performance Tuning

To enable efficient processing for this job, it is suggested that an index be added to the DB. The Posting Lines Catalog should have this index, if testing of this report proves to be slower than desired. Please see the later section below entitled "Index to Improve Performance".

This batch program also allows simultaneous processing of more than one instance. Using the Exclude Transaction Types parameter in multiple instances of the program, will allow individual instances to assure a select set of transaction types, while all instances together would assure all transaction types.

Process Steps	Messages
1. Parameter Validation	Run Started Parameters are listed Systems Assurance Process has started Parameter validation has started *(If the parameter is invalid, the invalid value will be displayed in the log along with the error message.)* • The valid format for Transaction Last Date is MM/DD/YYYY. (*This is issued for invalid format as well as for an invalid date.)* • Excluded Transaction Types parameter cannot be blank *(Parameters left blank to default by the program will be displayed)* • Progression counter size is empty so setting its default value to: 1000 *(Parameters left blank to default by the program will be displayed)* *(When no accounting line records are found to match selection criteria, the following message is issued an no others are issued besides run ended)* • No records on DOC_ACTG were found • Parameter validation completed
2. Generation of Report	Reports output folder mapped Locations of two report outputs given *(selection of record messages are inserted here before report completion)* Rendering Report Started Rendering Report Completed Run Ended
3. Selection of Records	Selecting eligible records Starting accounting line selection Number of records processed are: #### *(Where #### is the progression counter size. Message repeats until all records are selected. Count of last message will likely be less than progression counter.)* Finished accounting line selection Selecting eligible records completed

Major Input

- Common Accounting Line Catalog (DOC_ACTG): provides data for first selection source
- Transaction Control table (R_GEN_DOC_CTRL): provides selection with a list of active transaction codes in those transaction types not excluded
- Posting Line Catalog (PSTNG_LN_CAT): provides data for second selection source

Major Output

- Modification Transactions with Missing Posting Lines report
- New and Cancellation Transactions with Missing Posting Line report

Batch Parameters

Parameter Name	Description	Default Value
CLIENT_NM (Client Name)	Client Name for report. [*Optional*]	(blank)
DOC_LAST_ DT (Transaction Last Date)	The Transaction Last Date (Valid format is MM/DD/YYYY). [*Optional*]	(blank)
EXCLUDED_ DOC_TYP (Excluded Transaction Types)	Excluded Transaction Types (Enter one or more delimited with a comma, all upper case). [*Required*]	BG, EV, IN, ITI, PE, PI, RC, RN, SO, TI, TM, VCC, VCM
PROG_CTR_ SZ (Progression Size Counter)	Progression Counter Size [*Optional but will default to 1000 if left blank*]	1000
(Excluded Transaction Codes)		UR
(Excluded Event Types)		PR08, PR09, PR10, ST51, ST61

Chain / Job Return Code

Return Code	Condition
Successful (1)	All parameters were valid. If records were found on the DOC_ACTG table and none of those records have a missing posting line
Warning (4)	All parameters were valid, and if no records on DOC_ACTG were found to match DOC LAST DT parameter for a transaction type not listed in the EXCLUDED DOC TYP parameter.

Non Fatal Error (8)	All parameters were valid, selection found accounting line records were found that matched selection criteria, and reports were successfully generated. At least one accounting line was found without a posting line where it should have possibly had one. (See Problem Resolution section for verification steps.)
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

None, as accounting lines are presented in the order found in the Accounting Line Catalog (DOC_ACTG).

Selection Criteria

Common items for both reports:

- If the Transaction Last Date parameter is entered, then both queries will retrieve the accounting lines that are >= to the value entered. If the date is blank then the queries will search the entire DOC_ACTG.
- Accounting lines for transaction codes are selected that are not in one of the excluded transaction types and are marked as active on the Transaction Control table.
- Accounting lines are only retrieved where the transaction phase (DOC_PHASE_CD) is *Final* (3) or *Historical Final* (5).
- Selected accounting lines are only selected for the report if at least one matching posting line cannot be found on the Posting Line Catalog.

Criteria for only New and Cancellation selection

- Records that are transaction function (DOC_FUNC_CD) of *New* (1) or Cancellation (3) for the New and Cancellations report.
- Records where $0.00 is the Line Amount (LN_AM), Closed Amount (AL_CLSD_AM), and Referenced Amount (RFED_LN_AM) are not selected because these accounting lines will not have a posting line because there is no value there to post.

Criteria for only Modification selection

- Records that are transaction function (DOC_FUNC_CD) of *Modification* (2) are selected for the Modification report.

Indexes to Improve Performance

This index would be a non-unique index on the Posting Lines Catalog [PSTNG_LN_CAT]. The fields to be used should be:

1. DOC_ACTG_LN_NO
2. DOC_VEND_LN_NO
3. DOC_CD
4. DOC_ID
5. DOC_DEPT_CD
6. DOC_VERS_NO
7. DOC_PSTNG_NO

The basic syntax for an index if it was to be created in your environment and it is Oracle would be akin to the following:

```
CREATE INDEX NSA16_ PSTNG_LN_CAT
  ON PSTNG_LN_CAT (
DOC_ACTG_LN_NO,
DOC_VEND_LN_NO,
DOC_CD,
DOC_ID,
DOC_DEPT_CD,
DOC_VERS_NO,
DOC_PSTNG_NO
  )
```

This index would be a non-unique index on Transaction Accounting [DOC_ACTG]. The fields to be used should be:

1. DOC_ACTG_LN_NO
2. DOC_VEND_LN_NO
3. DOC_CD
4. DOC_ID
5. DOC_DEPT_CD
6. DOC_VERS_NO
7. LN_AM

The basic syntax for an index if it was to be created in your environment and it is Oracle would be akin to the following:

```
CREATE INDEX NSA16_ DOC_ACTG
  ON DOC_ACTG (
DOC_ACTG_LN_NO,
DOC_VEND_LN_NO,
DOC_CD,
DOC_ID,
DOC_DEPT_CD,
```

```
DOC_VERS_NO,
LN_AM
  )
```

This index would be a unique index on Transaction Accounting [DOC_ACTG] and is recommended for incremental runs. The fields to be used should be:

1. DOC_CD
2. DOC_DEPT_CD
3. DOC_ID
4. DOC_VERS_NO
5. DOC_VEND_LN_NO
6. DOC_COMM_LN_NO
7. DOC_ACTG_LN_NO
8. DOC_LAST_DT

The basic syntax for an index if it was to be created in your environment and it is Oracle would be akin to the following:

```
CREATE INDEX NSA16_ DOC_ACTG
  ON DOC_ACTG (
DOC_ACTG_LN_NO,
DOC_VEND_LN_NO,
DOC_CD,
DOC_ID,
DOC_DEPT_CD,
DOC_VERS_NO,
LN_AM
  )
```

*Please note that the names of the indexes are not in conformance with the Baseline naming scheme; however, it is important that they be named in a different manner than Baseline, so that we do not overlap a possible future new index with the name chosen at your site. Also, you should maintain conformance with the name of these indexes for your site, so that you do not break your site's processes and procedures. The names above are samples and may not be applicable at any individual site.

Problem Resolution

Transactions processed to final without a posting line that are not the last step in a chain (AD/MD/CR/etc) or a stand-alone transaction (OC/FA/JV/etc) should be corrected by submitting another transaction to perform all accounting not originally recorded. The Journal Voucher is the most common transaction type for such corrections, but is not the only choice.

Transactions that have been referenced by others require additional analysis to determine how much should be corrected, given that a subsequent transaction may or may not have liquidated a

portion of the original amount not recorded. Any open amount on such an accounting line should be put off on a new accounting line for future referencing and modifications. The procedure for doing so would be to reduce the accounting line without a posting line to equal the closed amount. If this modification triggers a posting line, that posting should be accounted for in the adjustment. The accounting line now closed is then copied and set to be the amount that was previously open. The missing postings should then be recorded with a Journal Voucher.

The report will list any memo asset that is fully depreciated and then disposed. This FD will not have any posting lines as the depreciation of the memo asset has removed the account balances for the asset so there is nothing left to post. After reviewing this is the case, the DOC_LAST_DT parameter should be incremented to the day after the FD as part of normal SA16 processing so that the record will not appear on future reports.