CGI Advantage® 4

Administration Utilities Run Sheets Guide

Table of Contents

1 Purpose of the System Administration Guide

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

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

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

1.1 Common terms and glossary used

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

2 Description of Processes

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

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

System Wide Batch Parameters:

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

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

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

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

Note:

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

Pivot Date/Year Validation:

Note:

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

2.1 Administration Conversion Batch and Chain Processes

Descriptions of the Administration Conversion Processes are described in this section.

- Data Conversion for Prior Bypass Functionalities

2.1.1 Data Conversion for Prior Bypass Functionalities

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

Overview

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

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

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

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

Major Input

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

Major Output

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

Job return codes have been mentioned in below table.

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

Sort Criteria

N/A

Selection Criteria

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

Problem Resolution

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following two conditions. • Encounters any runtime exceptions. • Failed during restart. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	If another instance of the job has already been scheduled and run successfully, then this job should not be restarted – only a new job should be scheduled.
	Failed while restarting the job since another instance of the job has already been run successfully. **Sample Message**: Cannot restart the job since another instance of this job has already been run successfully.	**Recommendation:** Schedule a new job.	
	If job failed due to Column FINAL_ASSIGNEE_LEVEL and FINAL_ASSIGNEE_ID	Make sure that Bypass Lower Level Approval functionalities are deployed properly.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and run successfully, then this job should not be restarted – only a new job should be scheduled.
System Failure (20)	When the job is terminated because of database	The reason for the System Failure needs to	If another instance of the job has already

Possible Return Codes	**Condition**	**Recommendation**	**Other Instructions**
	server or network issues.	be investigated. The job can either be restarted or schedule a new job.	been scheduled and run successfully, then this job should not be restarted – only a new job should be scheduled.

2.2 Administration Utilities Batch and Chain Processes

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

- Advantage/infoAdvantage Security System Assurance
- Automated Workflow Escalations
- Business Process Metada Generation
- Designer to XML Synchronization Process
- ECM Synch
- Enterprise Search Lite Indexer
- Expired Alert Clean Up
- Expired Security Role Clean Up
- Flow Job Scheduler
- Job Parameters Updater
- Lock Idle User
- Maintain User Configurations
- Menu Generator
- Multi-threaded Table Loader
- Multi-threaded Transaction Loader
- Password Expire Email Notification
- Populate Advantage Metadata
- Populate Business Object Metadata
- Populate Transaction Codes
- Populate Workflow Meta Data
- RLS Existing Integrated Role Users
- Server Data Object (SDO) Cache Report
- System Maintenance Utility
- Workflow Delegate To Alternate Approver
- Worklist Details Synchronization

2.2.1 Advantage/infoAdvantage Security System Assurance

Job Name	Advantage/infoAdvantage Security System Assurance
Recommended Frequency	This job should be run whenever there are differences in infoAdvantage security information maintained in the various Advantage applications.
Single Instance Required	No
Can be restarted?	No
Reports generated	Yes

Overview

The Advantage/infoAdvantage Security System Assurance batch job is used to synchronize security information between the Advantage and infoAdvantage.

This batch job can be run in one of two modes:

- Reporting
- Synchronization

This job provides a system assurance process driven from the Advantage side and focused on detecting infoAdvantage security entities defined in Advantage but not present in infoAdvantage security. Specifically the following situations will be detected and optionally corrected:

- Advantage Security Roles with the Integration Flag checked on the Security Role page that do not correspond to infoAdvantage User Groups
- Active Advantage Users belonging to one of more Advantage Security Roles with the Integration Flag checked on the Security Role page that do not correspond to infoAdvantage Users or correspond to infoAdvantage Users that do not belong to all of the infoAdvantage User Groups indicated by the Advantage Security Roles
- The User Name and Email Address associated with the user on the User Information page are synched to the infoAdvantage application, if the UPDATE_BO_USER parameter on the Application Parameter (APPCTRL) table is set to *true*.

When to Run

This job should be run whenever there are differences in infoAdvantage security information maintained in the various Advantage applications. The differences can be due to system restores, software issues or security maintenance applied directly within the infoAdvantage application.

Major Input

- Security Role (R_SC_SEC_ROLE)
- User Information (R_SC_USER_INFO)
- User Role Link (R_SC_USER_ROLE_LNK)

- Integration User Information (INTG_SC_USER_INFO)
- Integration User Role (INTG_SC_USER_ROLE)
- IN_APP_CTRL for UPDATE_BO_USER parameter value.

Batch Parameters

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

Parameter	Description	Default Value
JOB_MODE	The job running mode. 0 for the reporting mode and 1 for the synchronization mode.	0
SERV_NAME	Name of the BO XI Server	HOST:PORT
PASSWORD	Password to connect to BO	No
USER_ID	User ID to connect to BO	No

Major Output

- Updates occur in the infoAdvantage application if the job is run in synchronization mode.
- The System Assurance Report is generated if the job is run in report mode.

Job Return Code

The following table shows the potential job return codes for the Advantage/infoAdvantage Security System Assurance job.

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Failed (12)	The job will fail under the following conditions: Parameters are invalid. If the records are not processed properly. If the specified BO XI Server is not being connected.

Sort Criteria

No sorting criterion is being used.

Selection Criteria

- Security Roles that have Integration Flag (INTG_SEC_ROLE_FL) checked are selected.

- Users that are active and have been assigned the Security Roles that have the Integration Flag checked are selected.

Problem Resolution

If the job fails, correct the problem and submit a new job.

2.2.2 Automated Workflow Escalations

Chain or Job Name	Automated Workflow Escalations
Recommended Frequency	Daily as part of the nightly cycle or on demand
Single Instance Required	Yes
Can be restarted?	Yes
Reports generated	Pending Worklist Items

Overview

The Automated Workflow Escalations batch job performs the warning notification and escalation of Worklist items that are pending in a User's Worklist for more than a specific period of time. This is to ensure that the Worklist item is worked on by the specified time period by sending email reminders to Assignee and escalating the item to more Assignee(s) by creating additional routings to them so that they can work on them.

Approval Rules can be set up for a transaction to enable warning notification and escalation for each Approval Level. Warning Threshold Age, Escalate Threshold Age 1 and Escalate Threshold Age 2 can be specified for each Approval Level(s) on Approval Rules. The following condition has to be satisfied during the setup that also drives the sequence of events on aging: Warning Threshold Age < Escalate Threshold Age 1 < Escalate Threshold Age 2. When a Worklist item ages beyond the Warning Threshold Age then the job sends a Warning e-mail notification to the Assignee asking him/her to take action on the item. The warning email notification is sent every time the job is run until the Worklist item reaches Escalate Threshold Age 1 or Assignee worked on the item and the item is no longer in his/her Worklist. When Worklist item reaches Escalate Threshold Age 1 then the job performs Level 1 escalation of the item. Escalation involves creation of an additional routing to Escalate Assignee ID 1 specified for the Approval Level on Approval Rules and notifying the Escalate Assignee ID 1 and Original Assignee of the action through e-mail. After the escalation, both Escalate Assignee and Original Assignee will be able to work on the Worklist item through their Worklist. When any one of them approves the item, it moves ahead in the Workflow chain. If the Worklist item is still not worked on and it reaches Escalate Threshold Age 2 then the job performs Level 2 escalation, which is similar to Level 1 escalation exception that the additional escalated routing and email notification goes to Escalate Assignee ID 2.

Age of the Worklist item is calculated as the number of days between the Assigned Date of Worklist Item and the Application System Date on Application Parameters. Wherever email notification to Original Assignee/Escalate Assignee ID 1 or 2 (addressed in general as Assignee below) is involved during warning notification or escalation, the recipients of the email are as follows:

- If the Assignee is a User then the e-mail notification will be sent to the particular User.
- If Assignee is an Approval Role and none of the Role's Approvers has done 'Take Task' on it then the e-mail notification will be sent to all of the Approvers belonging to that Role.
- If the task has been taken by an Approver with that Role, then the e-mail notification is sent to that particular Approver only.

Example:

Approval Rules setup for an Approval Level:

Assignee ID:	UserOrig
Warning Threshold Age:	5
Escalate?:	true
Escalate Threshold Age 1:	7
Escalate Assignee ID 1:	UserA
Escalate Threshold Age 2:	9
Escalate Assignee ID 2:	UserB

Suppose the Assigned Date for a Worklist Item is 10/1/2011.

If the job is run daily, then when Application System Date on Application Parameters equals 10/6/2011 or 10/7/2011 a warning email notification is sent out to UserOrig. Warning notification is sent until the item reaches Escalation Threshold Age 1.

If the job is run when Application System Date equals 10/8/2011 then Level 1 escalation happens and item is additionally routed to UserA. Email notification of escalation is sent to UserOrig and UserA.

If the job is run when Application System Date equals 10/10/2011 then Level 2 escalation happens and item is additionally routed to UserB. Email notification of escalation is sent to UserOrig and UserB. Now the item can be worked on by any one of the following Assignee(s) in whose Worklist the item is sitting in: UserOrig, UserA, or UserB.

Note if the item was worked on in between receiving such email notifications then the item moves out of that Approval Level and is no longer aging or available for further events of warning or escalation. For example, if UserOrig approved the item on 10/6/2011 on receiving warning notification then no further warning or escalation happens on that item.

The job can send three types of email notifications to an Assignee User consolidating all of the Transactions selected under each category:

- Workflow Warning

  This email notification is sent to the Assignee of the Worklist Items eligible for the Warning e-mail notification. The e-mail enlists the Worklist Items pending in his/her Worklist.

- Worklist Notification about items escalated from his/her Worklist

  This email notification is sent to the Original Assignee from whose Worklist the items are being escalated. The e-mail enlists all of the Worklist items that have been escalated from his Worklist.

- Workflow Escalation to Escalate Assignee

  This email notification is sent to the Escalate Assignee to whom the Worklist Items are being escalated. The e-mail enlists the Worklist Items being escalated to the Escalate Assignee’s Worklist.

The Transaction information in the email is a hyperlink that when selected will take the User to his/her Worklist upon successful login.

The job creates a report, which enlists all of the pending Worklist items that were processed for Warning or Escalation. The report consists of three sections:

- Warning Email notifications sent

  This section enlists the Worklist Items with their Age and the Assignee to whom the Warning email notification was sent. The Assignee can be a Role or a User specified by the column "Is Assignee Role".

- Items Escalated at Level 1

  This section enlists the Worklist Items with their Age and the Escalate Assignee to whom the Worklist Items were escalated at Escalation Level 1.

- Items Escalated at Level 2

  This section enlists the Worklist Items with their Age and the Escalate Assignee to whom the Worklist Items were escalated at Escalation Level 2.

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

Process Steps	**Messages**
1. Parameter Validation	• Validating Batch Parameters • Parameter validation successful/failed. If the parameter is invalid, the invalid value will be displayed in the log along with the reason of failure.
2. Initialize job resources	• Initializing Job resources
3. Worklist record processing	• Processing Worklist items • Records processed: <records count> • Total Worklist records processed: <Records Count>- if the job is new • New Worklist records processed: <Records Count>- if the job is restarted • No eligible records found to be processed- if no records were found eligible for processing.
4. Send Email Notifications	• Sending emails • Sending emails completed
5. Create Report	• Generating Report • Report generated successfully
6. Job complete	• Job processing complete

Restartability Information

If the job fails in any of the above steps the job can be restarted after resolving the error. If the job is restarted, it will start from the step where it failed earlier and will not start from the beginning.

Major Input

- Worklist (WF_APRV_WRK_LST) and Worksheet (WF_APRV_SH) for selecting records eligible for Warning and Escalation. Approval Rules (R_WF_APRV) is referred for setup.

- Records added to WF_ESCL_TRACK for sending Email notification and creating Report
- IN_APP_CTRL for Application System Date
- WF_NOTIF_TMPL for referring the template to be used for sending e-mail notification

Batch Parameters

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

Parameter	Description	Default Value	Explanation
CLIENT_NM	Client Name (Optional)	No Default	Client Name to be printed on top of the report.
COMMIT_SIZE	Commit Size (Optional)	500	Very high value can cause out of memory errors, very low value can cause too frequent commits and hence more processing time.
DOC_CODE	Transaction Code (Optional, Comma separated list)	No Default	Leaving it blank processes all Transactions. Otherwise enter the list of Transaction codes for which warning/escalation should happen.
DOC_DEPT_CODE	Transaction Dept Code (Optional, Comma separated list)	No Default	Leaving it blank processes selected Transactions regardless of its Transaction Department. Otherwise, enter the list of Transaction Department codes for the selection of Transactions for warning/escalation.
MODE	Job Mode (Required) Valid Values: 1- Warning Notification 2- Escalate 3- Both	3	If the job should only do Warning notification for due Worklist items then select the Warning Notification mode. In this mode, no escalation happens even if the item is due for escalation. If the job should only do escalation for due Worklist items then select Escalate mode. In this mode no warning notification happens even if the item is due for warning notification. When mode is Both

			(recommended), warning notifications are sent out for items due for Warning Notification and escalation happens for items due for escalation.

Major Output

- Escalated Routing on Worklist (WF_APRV_WRK_LST)
- New/Updated entries in WF_ESCL_TRACK. These entries are used by the job for sending email notifications and creating report. This table can be queried for an action taken on any Transaction for any particular user.
- Pending Worklist Items report

Job Return Code

Return Code	Condition
Successful (1)	All of the Worklist Items are processed, emails are sent and the report is generated successfully.
Warning (4)	No eligible records found.
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: - Parameters are invalid - Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

- The job applies the below sorting criteria for selecting the Worklist records:
- Transaction Code
- Transaction Dept Code
- Transaction ID
- Transaction Version No
- Approval Level
- Worklist Type
- Assignee
- Assignee Flag

- Worklist ID

Selection Criteria

The job will select the records that meet following criteria:

- Approval Level > 0
- The Worklist item was not added on current Application System Date.
- The Worklist item is not result of Escalated Routing
- The Worklist item meets the Transaction Code criteria specified in Job parameters
- The Worklist item meets the Transaction Dept Code criteria specified in Job parameters
- Following additional criteria is added:
  - The Application System Date equals or exceeds Escalation 2 Eligibility Date and the Worklist Item has not been escalated at Escalation Level 2
  - OR the Application System Date equal or exceeds Escalation 1 Eligibility Date but is less than Escalation 2 Eligibility Date and the Worklist Item has not been escalated at Escalation Level 1
  - OR the Application System Date equals or exceeds Warning Eligibility Date but is less than Escalation 1 Eligibility Date and the Worklist Item has not been escalated

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the Worklist Items are processed, emails are sent and report is generated successfully.	N/A	N/A
Warning (4)	No eligible records found on the Worklist.	Schedule a new job when new transactions are submitted.	Alternatively, the job can be rescheduled with a different set of parameters.
Non Fatal Error (8)	N/A	N/A	N/A
Failed (12)	Job failed due to failing parameter validations	Schedule a new job.	N/A
	Job failed due to Fatal conditions.	The job can fail under the following two conditions. 1) Encounters any runtime exceptions and	If the job is not restarted, then no email notifications will be sent to the users and no report will be

Possible Return Codes	Condition	Recommendation	Other Instructions
		2) Failed during restart. Investigate the exception reported by the process, resolve the error and restart the job.	generated for the records processed by this job in the failed run.
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated and the job should be restarted.	If the job is not restarted, then no email notifications will be sent to the users and no report will be generated for the records processed by this job in the failed run.
System Failure (20)	When the job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	If the job is not restarted, then no email notifications will be sent to the users and no report will be generated for the records processed by this job in the failed run.

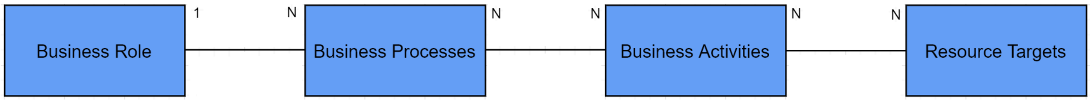
2.2.3 Business Process Metadata Generation

Chain or Job Name	Business Process Metadata Generation
Recommended Frequency	On demand when new business processes, business activities, and resource targets have been created.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	None

Overview

This batch job takes the various components of business role configuration and transforms it into XML Page Metadata for the left-hand navigation menus, business process landing pages, and home page extensions. This job also complements the Publish action found on many of the configuration pages. Please see the *Business Role User Guide* for more information on this action and the configuration pages.

Business role is the primary means of navigation within CGI Advantage. A role is a hierarchy of three organizing components to present system resources to end users. The hierarchy is as follows:



The XML created by this job is used by the Key Value Metadata generator process to add all the respective pages and extensions into the PALDB.

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

Process Steps	Messages
1. Parameter validation	Validating Batch Parameters. • Validate if the Meta Data Path parameter is empty. If empty then raise an error.
2. Select records	Select records as per the selection criteria listed in the Selection Criteria section.
3. XML creation for menu pages	Create menu pages for all active roles. The menu pages will have all the necessary hierarchy of business process/ business activities, and business roles.
4. XML creation for business process	Create business process landing pages for all business processes within every role.

Process Steps	Messages
landing pages	
5. Create home page extensions	Create Home Page extensions for all Business Roles and Home Page combinations.
6. Job complete	Run ended

Restartability Information

The job has to be rerun. It does not store any checkpoints and cannot be restarted.

Major Input

- Business Role (R_BUS_ROLE)
- Business Process (R_BUS_PROC)
- Business Activities (R_BUS_ACTV)
- Resource Target (R_RSRC_TRGT)
- Business Role Association (R_BUSROLE_BUSPROC , R_BUSPROC_BUSACTV, R_BUSACTV_RSRCTRGT)
- Home Page Configuration (R_HOME_PAGE_CNFG, ADV_WIDGET_LIBRARY, R_QUICKLINK_LIB)

Batch Parameters

Parameter	Description	Default Value	Explanation
GENERATE_HOME_PAGE	Valid values are: • true (setting the value to true will generate home pages) • false (setting the value to false will not generate home pages)	false	If set to true, this job creates/updates Home Page XML.
META_DATA_PATH	Metadata output location	$$AMSROOT$$/ApplicationMetadata/metadata/common	This batch job creates XML as the output and writes to the location specified under this parameter.

Major Output

- Creates Menu XML for all business roles and add necessary extensions in the Application.xml. The Menu xml has all the metadata information.
- Creates a business process landing page for all business processes and also add page entries in the Application.xml and CrossReference.xml files.
- Create home page extensions in the application.xml file.

Job Return Code

Return Code	Condition
Successful (1)	All records processed and xml metadata created successfully.
Warning (4)	This Return Code is issued under the following condition: • Skip records (Process, Activity, or Target) if any of them failed during processing and write to job log.
Non-Fatal Error (8)	This job does not use this return code.
Failed (12)	The job may fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations
Terminated (16)	This return code is issued when the job is terminated by the user.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues.

Sort Criteria

- The Business Role Processes are sorted by DISP_ORD
- The Business Process Activities are sorted by DISP_ORD
- The Business Activities Resources are sorted by GROUPING and RSRC_DISP_ORD.

Selection Criteria

This batch job selects the records in the following sequence.

1. Select all active records from Business Role (R_BUS_ROLE).
2. For all active business roles, get business role to business process associations from R_BUSROLE_BUSPROC where the business process is active.
3. For all active business processes, get business process to business activity associations from R_BUSPROC_BUSACTV where the business activity is active.
4. For all active business activities, get business activity to resource target associations from R_BUSACTV_RSRCTRGT where the resource target is active.

5. For all active business roles associated with home page, get home page configuration from R_HOME_PAGE_CNFG where the home page is active.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the necessary Menu and Business Process landing page XML metadata is created successfully for all roles. The Home Pages are also added for each role.	N/A	N/A
Warning (4)	If the batch job skips records due to bad data, then the exceptions are written to the job log.	Verify the job logs and rectify the problem.	View job logs for help on the problem.
Failed (12)	If the batch job fails then the batch job has to be rerun or the System Administrator has to be notified.	Rectify the problem and schedule a new job.	View job logs for help on the problem.

2.2.4 Designer to XML Synchronization Process

Chain or Job Name	Designer to XML Synchronization Process
Recommended Frequency	On Demand
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

The Configure Page (DESIGNER) page (typically referred to as “Designer”) allows page and business object extensions to be created and tested rapidly. However, to support the rapid testing of extensions, the extensions created by Designer are stored in a database table (referred to as “online extension storage”) that is not efficient for production operation. Also, once you start configuring a page or business object using Designer, it makes an image of the current custom extensions for the page (if any) in the online extension storage that will not include any subsequent file-based extensions made to the same page.

The Designer to XML Synchronization Process resets this cycle:

- Intelligently merges your recent Designer page and business object extensions to the XML-based Advantage metadata repository – the source of truth for page and business object design metadata.
- Resets the online extension storage so that the system resumes reading custom extensions from the runtime metadata files (often referred to as “PalDB files”).
- Updates the Extension Status on the corresponding Transaction COA Display Configuration (COACONF) record when page extensions generated using this page are processed.

You can also run the Designer to XML Synchronization Process in the Financial or HRM application to generate Transaction and Department conditions that can be used to configure conditional page extensions.

Where to Run Designer to XML Synchronization

The Designer to XML Synchronization job must be run separately for each Advantage application for which the DESIGNER page has been used to configure pages.

- For example, for Financial pages, run the Designer to XML Synchronization job from the Financial jobs of the Batch Catalog to process Financial pages and write metadata extensions in the location configured via the batch parameter.
- For Common pages, run the Designer to XML Synchronization job from any of the applications within the unified environment. The job writes Common page extension files to the sitemaintained sublayer under each application. There is no need to run the job for each application.

Pages/Business Objects Belonging to Application	Run From	Reads From	Writes To
Administration	Admin Batch Catalog	Common schema associated with Admin	Custom Extensions metadata location reserved for the environment. Writes to Folder: adm/sitemaintained In addition, extensions configured for Common pages are also written to other <app>/sitemaintained folders within the unified environment.
Financial	Financial Batch Catalog	Common schema associated with Financial	Custom Extensions metadata location reserved for the environment. Writes to Folder: fin/sitemaintained In addition, extensions configured for Common pages are also written to other <app>/sitemaintained folders within the unified environment.
HRM	HRM Batch Catalog	Common schema associated with HRM	Custom Extensions metadata location reserved for the environment. Writes to Folder: hrm/sitemaintained In addition, extensions configured for Common pages are also written to other <app>/sitemaintained folders within the unified environment.
Performance Budgeting	Admin Batch Catalog Note: This is a separate job specifically configured for PB pages than what is run for “Admin” pages.	Common schema associated with PB	Custom Extensions metadata location reserved for the environment. Writes to Folder: pb/sitemaintained In addition, extensions configured for Common pages are also written to other <app>/sitemaintained folders within the unified

Pages/Business Objects Belonging to Application	Run From	Reads From	Writes To
			environment.
VSS	VSS batch catalog	Common schema associated with VSS	Custom Extensions metadata location reserved for the environment. Writes to Folder: vss/sitemaintained

Typical Designer to XML Synchronization Workflow

At a high level, the following extension workflow is expected to occur periodically:

1. Users use the Configure Page (DESIGNER) page to configure pages and business objects for an application area of Advantage. You can also configure all transaction pages in the Financial application to hide or rearrange COA fields. Refer to the Transaction COA Display Configuration (COACONF) page topic in the *CGI Advantage - Chart of Accounts User Guide* for more information.
2. Users must add the configured page and business object to the migration queue by selecting Add to Migration Queue from the page-level menu on the Configure Page (DESIGNER), Configure Business Object or Transaction COA Display Configuration (COACONF) record. Only the configured extensions added to this queue will be picked up by the job for processing.
3. Run Designer to XML Synchronization Process to synchronize the latest “online” extensions to the XML metadata repository.
4. Run Metadata Key/Value Generator (PalDB generation) to incorporate the latest XML-based metadata into the runtime metadata files. If extensions are processed for Common pages, perform this step for all applications within the unified environment.
5. Move the designer extensions generated from the batch run to the “source of truth” metadata in whatever the form (for example, version control, file directory, and so forth) it is maintained for a given environment.
6. Repeat from #1.

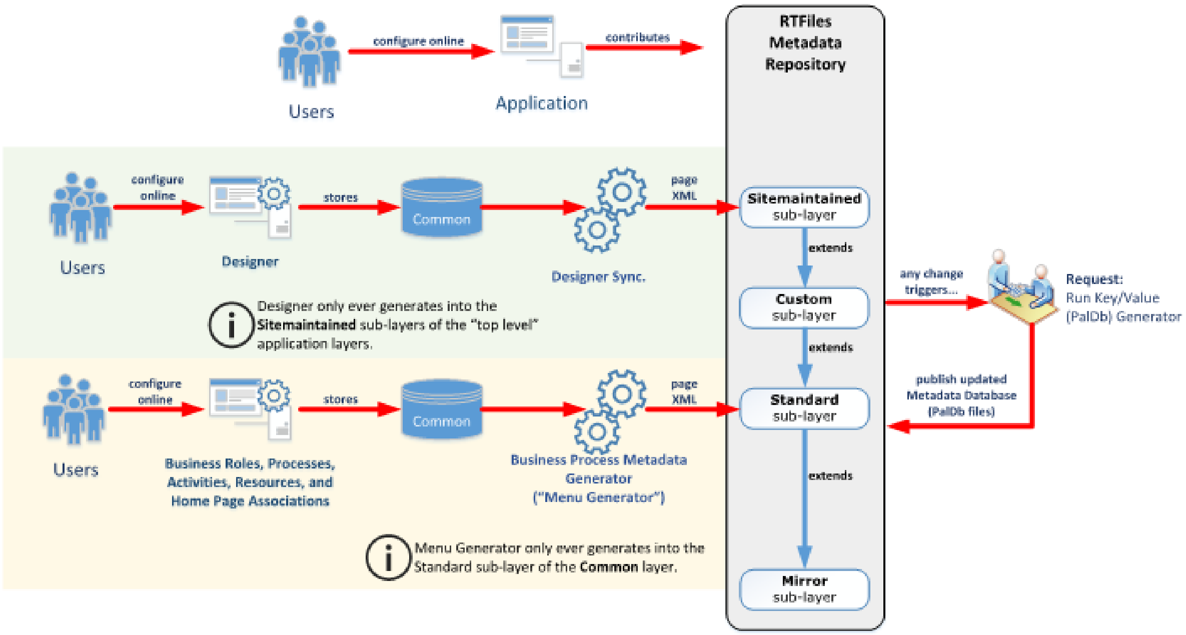
The generation of Transaction and Department conditions can occur as a one-time setup as part of the preparation to migrate to the CGI Advantage 4 Financial or HRM application. Another option is to run as needed to regenerate the conditions when setup changes occur on the Transaction Control (DCTRL) or the Department (DEPT) table to add or delete a record, update the name, or change a record from Active to Inactive or vice versa.

- After a condition entry is generated for a Transaction Code or Department, even if it is later removed from the table, the entry remains in the conditions.xml file with the description showing a “ - Deleted” suffix with the active property set to “false” to indicate that the value is has been deleted from the table.
- If a record becomes inactive on the Transaction Code or Department table, the generated entry’s description shows a “ - Inactive” suffix.

Various Sources of “Online” Configuration

The next diagram is a high-level depiction of how some of these most common application configuration tools feed into the runtime metadata.

- Configuration data entry is captured by the application and stored in the Advantage Common database schema table (ADV_EXT_ONLN).
- Various batch process-based utilities merge the latest configuration to the XML metadata repository.
- The Key/Value (PalDB) Generator is run to incorporate the latest XML-based metadata into the runtime metadata files.


Users
configure online
Application
contributes
RTFiles Metadata Repository
Users
configure online
Designer
stores
Common
Designer Sync.
page XML
Sitemaintained sub-layer
extends
Custom sub-layer
extends
Standard sub-layer
extends
Mirror sub-layer
Designer only ever generates into the Sitemaintained sub-layers of the “top level” application layers.
any change triggers...
Request: Run Key/Value (PalDb) Generator
publish updated Metadata Database (PalDb files)
Users
configure online
Business Roles, Processes, Activities, Resources, and Home Page Associations
stores
Common
Business Process Metadata Generator (“Menu Generator”)
page XML
Menu Generator only ever generates into the Standard sub-layer of the Common layer.

Designer-based Extension Metadata Flow Scenarios

It is helpful to understand the overall flow of the Designer-based extension process using a few scenarios. Please refer to the “Designer Extensions” section in *CGI Advantage 4 Extract Deploy Extension Metadata Guide* for additional details on scenarios.

The following table shows the various steps that the job goes through and the messages issued at each step, as applicable.

Process Steps	Messages
1. Parameter Validation	• Metadata path parameter is invalid. • Metadata path location <META_DATA_PATH> does not exist. • Metadata path location <META_DATA_PATH> does not have Write Permission. • Designer Sync Application is not passed as a

Process Steps	Messages
	parameter. Using <Application where batch running>. • Invalid value in Generate Condition Metadata parameter, defaulting to No. • Parameter validation failed. • Exception occurred during job parameter validation.
2. Selection & Processing Records	• Processing only pending extensions that have been explicitly marked as ready for synchronization. • Processing pending extensions of all pages and business objects. • Processing <Page Name / Business Object Name>. • Processing <Page Name / Business Object Name> failed. • No records to process. **For Page Extensions:** • Processing Page <Page Name>. • Processing Common Page <Page Name>. • Writing Common Page extensions for <app> application at the metadata location <META_DATA_PATH>/<app> • Processing pending extensions only for page <Page Name > passed as a job parameter, only if the page has been explicitly marked as ready for synchronization in Designer. • A problem occurred when trying to build the desired metadata for page: <Page Name>. • Provided page <Page Name> is currently in pending status and is not marked ready for sync. • No eligible pending records found for the specified page. Please make sure the fully qualified page name is correct. The parameter value is case-sensitive. • Page <pageName> has wrong number of parts. Should be in the format: app.type.group.pagename. • Page Alias cannot be extracted from the metadata for page <pageName>. Missing base key with value of 'overflow'. • Page <pageName> is overridden in the Designer

Process Steps	Messages
	but the configuration children are missing. • Page <pageName> is overridden in the designer but the configuration properties are missing. • Problem writing Page <pageName> xml file. • Extensions identified for COA Configuration ID <ID>. Updating extensions status. • Extension status on COA Configuration ID <ID> updated successfully. **For Business Object Extensions:** • Processing Business Object <Business Object Name>. • A problem occurred when trying to build the desired metadata for business object: <Business Object Name>. • Business Object <businessObjectName> has wrong number of parts. Should be in the format: app.type.businessObjectName. • Business Object Alias cannot be extracted from the metadata for page <businessObjectName>. Missing base key with value of 'overflow'. • Business Object <businessObjectName> is overridden in the Designer but the configuration children are missing. • Business Object <businessObjectName> is overridden in the designer but the configuration properties are missing. • Problem writing <businessObjectName> xml file. **For Transaction and Department conditions:** • Condition Metadata generation has started. • Condition Metadata generated Successfully.

Restartability Information

- This job cannot be restarted and has to be rerun. It does not store any checkpoints.

Major Input

- Designer Extensions Storage (ADV_EXT_ONLN)
- Transaction Control (DCTRL)
- Department (DEPT)

- Custom extensions metadata location that deployed in the current state of application environment.

Batch Parameters

Parameter	**Description**	**Default Value**	**Explanation**
GENERATE_CONDITION	Generate Condition Metadata. Valid values are Yes/No.	Financial Job: **No** HRM Job: **No**	This optional parameter is available for Financial and HRM jobs only. Set it to *Yes* to generate Transaction and Department conditions that can be used to configure conditional page extensions.
META_DATA_PATH	Metadata Output Location. Required and default value can be overridden.	**Administration Job:** /apps/CGIADV/RTFiles/**admin**/Custom/Extensions/ApplicationMetadata/metadata. **Financial Job:** /apps/CGIADV/RTFiles/**fin**/Custom/Extensions/ApplicationMetadata/metadata. **HRM Job:** /apps/CGIADV/RTFiles/**hrm**/Custom/Extensions/ApplicationMetadata/metadata. **VSS Job:** /apps/CGIADV/RTFiles/**vss**/Custom/Extensions/ApplicationMetadata/metadata. **Performance Budgeting Job:** /apps/CGIADV/RTFiles/**pb**/Custom/Extensions/ApplicationMetadata/metadata.	It is very important that this parameter value is properly configured to match the location where custom extensions are maintained for a given application environment. This should be the location where custom extensions from different sources (for example, file-based extensions) are placed and deployed for the current state of the application environment. Please refer to the appropriate environment setup documentation to find exact location details and correct the parameter value if necessary.
SYNC_APPLICATION	Application to Synchronize Default value **cannot** be	Administration Job: **adm** Financial Job: **fin** HRM Job: **hrm**	The main reason this parameter has a default value is to facilitate Advantage 4 PB implementations to

Parameter	Description	Default Value	Explanation
	overridden.	VSS Job: **vss** Performance Budgeting Job: **pb**	run the Metadata Sync job on the same Administration application server for both Administration and PB application pages configured in the Designer.
PAGE_QUALIFIED_NAME	Full Page Qualified Name (optionally use to process a single page with changes pending synchronization) Optional.		Do not use if you generated bulk page extensions using the COACONF page because the bulk page extensions must be processed together. This parameter is useful to run metadata synchronization for a single page when multiple designers are working on extensions and the system should not pickup extensions that are not ready and are being worked on by different users.

Major Output

- Create extension metadata files for pages that do not have prior extensions.
- Merge designer extensions on top of current custom extensions for a given page or business object, if applicable.
- Create/Update additional metadata configuration artifacts (for example, application.xml entries for extended pages or business objects) as applicable.
- Create/Update Transaction and Department entries in conditions.xml.

Job Return Code

Return Code	Condition
Successful (1)	All of the selected Designer Extensions Storage records are processed successfully. Transaction and Department conditions are generated successfully.
Warning (4)	The job issues a warning message if no Designer Extension Storage records are selected for processing and the Generate Condition

Return Code	Condition
	Metadata parameter value is *No*.
Failed (12)	The job will fail under the following conditions: • Parameter validations failed. • Page Qualified Name parameter provided but no record found matching the parameter value and the History flag is false in the Designer Extensions Storage (ADV_EXT_ONLN) table. • If no records are found in the Designer Extensions Storage (ADV_EXT_ONLN) table with the History flag equal to false and a Status indicating added to migration queue (that is equal to 5). • Exceptions occurred while processing records and writing XML files to an output location because permission issues, space issues, and other unpredictable exceptions.
Terminated (16)	This Return Code is issued when the job is terminated by the user.
System Failure (20)	This Return Code is issued when the job is terminated because of database server or network issues.

Sort Criteria

N/A

Selection Criteria

Extension processing

- If PAGE_QUALIFIED_NAME is populated: Record matches the PAGE_QUALIFIED_NAME parameter value, SYNC_APPLICATION parameter value, History flag is false, and Status indicates added to migration queue in the Designer Extensions Storage (ADV_EXT_ONLN) table.
- If PAGE_QUALIFIED_NAME is not populated: All records that match the SYNC_APPLICATION parameter, History flag is false, and Status indicating added to migration queue in the Designer Extensions Storage (ADV_EXT_ONLN) table.

Transaction and Department conditions generation

- Records are selected from the R_GEN_DOC_CTRL and R_DEPT tables. Condition entry is not generated if the Transaction or Department code value is equal to ALL, or if an identical condition entry already exists at the common layer or fin/hrm custom sub layer.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful	All of the parameters are validated	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
(1)	successfully.		
Warning (4)	Job ended with a Warning because there is no matching 'ready to be processed' records in the Designer Extensions storage table. **Sample Message:** No records to process.	Create extensions needed on the DESIGNER page, add them to the migration queue, if necessary and submit a new job request to process the records.	
Failed (12)	Job failed due to Fatal conditions.	This job can be failed for the following different reasons. • Invalid parameters • Record(s) not in correct status in the Designer Extensions Storage (ADV_EXT_ONLN) table. • Metadata Directory structure missing in output location • Write permission • Space constraints on storage volumes • Unknown run-time exceptions. In all of the above cases, the batch job will issue appropriate messages in the batch log. If the batch fails for any of the above reasons, please review the error and exception messages, address issues, and submit a new Job request.	If the current job run is processing multiple records based on selection criteria, any failures in the middle of processing may result in corrupted metadata files. In those cases, further analysis is needed and includes the following steps. 1. Address/fix issues that caused job failure. 2. Restore or clean up Custom Extensions folders for any inconsistencies. 3. If a restore is deemed necessary, then restore the custom metadata folders from a previous clean version. 4. Identify the storage records processed by the current run and reset the History

Possible Return Codes	Condition	Recommendation	Other Instructions
			flag. 5. Add the page(s)/business object(s) to migration queue if applicable. 6. Submit a new job request.
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. Schedule a new job as needed.	
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. Schedule a new job as needed.	

2.2.5 ECM Synch

Chain or Job Name	ECM Synch
Recommended Frequency	Whenever any changes are made in Advantage subsequent to attaching the file, there will be discord between the data in ECM and Advantage as there is no real time update in ECM after storage of the attachment. Therefore, in order to synchronize the data in ECM and Advantage this batch process should be run.
Single Instance Required	No
Can be restarted?	No
Reports generated	No

Overview

The ECM Synch batch job is used to synchronize the metadata information between the Advantage and Enterprise Content Management (ECM) systems.

If the job encounters any errors while sending this information, users can view the resulting errors in the job's log. The submit date of the job will be defined as the modified date populated on the Header component of the final transaction (Transaction Information section). The job will not keep any checkpoint as no updates are made to the Advantage application. This means that if the job is to process 5 records and after successfully processing 3 records it stops due to issues on the 4th record, the job will return the *Failed* Return Code. Subsequently, when the job is submitted again after rectifying the issue, the job will process all five records again instead of starting from the 4th record.

Once the metadata is initially sent to ECM, any changes made to the Advantage page will not be re-integrated to the ECM until this batch process is scheduled and executed again.

For example, when a user first uploads an attachment to an Advantage transaction, which is in draft phase, the values that were in the Advantage metadata fields at that time will be integrated to the ECM. If the user updates any of the Advantage transaction's metadata fields after adding the attachment and subsequently submits the Advantage transaction to final, the updated metadata will only be sent to the ECM when the Sync Batch Process is run.

When to Run

Whenever any changes are made in Advantage subsequent to attaching the file, there will be discord between the data in ECM and Advantage as there is no real time update in ECM after storage of the attachment. Thus in order to synchronize the data in ECM and Advantage this batch process should be run.

Major Input

- Not applicable

Batch Parameters

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

Parameter	Description	Default Value	Explanation
STRT_DT	Start Date	No Default	Optional. The Start Date is the beginning date the job will process records. All the records greater than or equal to this date will be selected for processing. Date in (MM/DD/YYYY)
END_DT	End Date	No Default	Optional. The End Date field indicates the date until which the job will process records. All the records less than or equal to this date will be selected for processing. Date in (MM/DD/YYYY)

Major Output

- Updates done in ECM

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	When one or more records being processed throw an error.
Non Fatal Error (8)	No eligible records found.
Failed (12)	The job will fail under the following conditions: - Parameters are invalid - Run time exceptions for unexpected situations.
Terminated (16)	This return code is issued when the job is terminated by the user.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues.

Sort Criteria

No sorting criterion is being used.

Selection Criteria

The batch process selects all Advantage transactions submitted to Final and saved table records, containing an attachment based on the set-up on the ECM Metadata Mapping table, within a specified date range provided by the user in the batch job parameters based on the transaction last date (DOC_LAST_DT) present on the transaction component. After selecting the appropriate records to be integrated, this job sends the Chronicle ID and metadata information associated with the record as per the ECM Mapping Table to the ECM so that the metadata information can be updated in the ECM database.

Problem Resolution

If the ECM Synch Batch job runs unsuccessfully, correct the problem and submit a new job.

2.2.6 Enterprise Search Lite Indexer

Chain or Job Name	Enterprise Search Lite Indexer
Recommended Frequency	This should be run when there is any change to the Business Roles configuration or Application Page Registration data.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	None

Overview

This batch job creates Lucene Indexes for all Page Codes and System Resource Targets for Enterprise Search Lite Enterprise Search. The system uses these indexes during runtime for fast text based searches. It's a three step process.

1. Parameter validation. It will validate if the INDEX_PATH parameter is a valid path where the index will be created.
2. This process will then select all Page Codes, Page Description, Page Type and Application from IN_PAGES_VIEW and create an index document for each record using the Lucene API. The process will select only those records where PAGE_SHW_FL is true.
3. The process will then select all active System Resource Targets (R_RSRC_TRGT) from the system joining various tables like R_BUSACTV_RSRCTRGT, R_BUSROLE_BUSPROC, and R_BUSPROC_BUSACTV and again create an index document for each record using the Lucene API.

Restartability Information

The job has to be rerun. It does not store any checkpoints and cannot be restarted.

Major Input

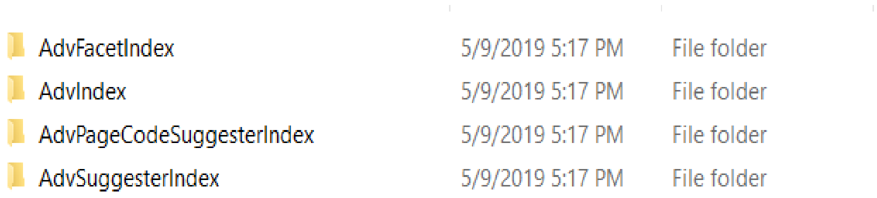
- IN_PAGES_VIEW
- Business Role (R_BUS_ROLE)
- Business Process (R_BUS_PROC)
- Business Activities (R_BUS_ACTV)
- Resource Target (R_RSRC_TRGT)
- Business Role Association (R_BUSROLE_BUSPROC, R_BUSPROC_BUSACTV, and R_BUSACTV_RSRCTRGT)

Batch Parameters

Parameter	Description	Default Value	Explanation
INDEX_PATH	Search index path	None	This batch job will create Lucene index under the directory mentioned by the parameter value.

Major Output

- Creates Lucene Index document for IN_PAGES_VIEW.
- Creates Lucene Index document for R_RSRC_TRGT.
- The output Lucene index is created under the folder mentioned in the parameter and will be organized under an AdvSearch folder with subfolders as shown in the following screen shot.



AdvFacetIndex	5/9/2019 5:17 PM	File folder
AdvIndex	5/9/2019 5:17 PM	File folder
AdvPageCodeSuggesterIndex	5/9/2019 5:17 PM	File folder
AdvSuggesterIndex	5/9/2019 5:17 PM	File folder

Job Return Code

Return Code	Condition
Successful (1)	All records processed and xml metadata created successfully.
Warning (4)	This job does not use this return code.
Non-Fatal Error (8)	This job does not use this return code.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

- IN_PAGES_VIEW is sorted by IN_PAGES_VIEW.PAGE_CD DESC, IN_PAGES_VIEW.APPL_IN_PAGES ASC

- Resource Targets are sorted by R_RSRC_TRGT.RSRC_TRGT_NM ASC, R_BUS_ACTV.BUS_ACTV_NM ASCR_BUS_ROLE.BUS_ROLE_NM ASC.

Selection Criteria

This batch job selects the records in the following sequence.

- This process will select all Page Codes, Page Description, Page Type and Application from IN_PAGES_VIEW.
- The process will then select all active System Resource Targets (R_RSRC_TRGT) from the system joining various tables like R_BUSACTV_RSRCTRGT, R_BUSROLE_BUSPROC, and R_BUSPROC_BUSACTV.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the indexes are created with the latest time and date stamp.	N/A	N/A
Failed (12)	If the batch job fails then the batch job has to be rerun or the system Administrator has to be notified.	Rectify the problem and schedule a new job.	View logs for help on the problem.

2.2.7 Expired Alert Clean Up

Job Name	Expired Alert Clean Up
Recommended Frequency	This job can be run on demand.
Single Instance Required	No
Can be restarted?	No
Reports generated	No

Overview

The Expired Alert Clean Up batch job automates the process of deleting the expired alerts, based on the entered Alert Purge Date. When this job is invoked, it deletes alert records for all recipients from the Alerts recipient (ADV_ALERTS_RECIP) table where the alert end date is less than the alert purge date entered by the user.

This job accepts three parameters:

1. Alert Purge Date
2. Send Email (Yes/No)
3. Send SMS (Yes/No)
4. Expired Alerts Clean Up Message

In addition to purging alert records for recipients, this job also creates an alert to all users in the system indicating that expired alerts have been cleaned up. Also, it will send an email to the users if the Send Email (Yes/No) parameter has a value of "Yes" and send an SMS to the users if the Send SMS (Yes/No) parameter has a value of "Yes".

The following table shows the various steps that the Expired Alert Clean Up job goes through at each step:

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters. • If Alert Purge Date is not entered then the following message is issued: Alert Purge Date is missing. • If Alert Purge Date is not in the format MM/dd/yyyy, then the following message is issued: Please enter date in MM/dd/yyyy format. • If Alert Purge Date is current or future date, then the following message is issued: Alert Purge Date cannot be current or future date. • If the Send Email parameter is not entered, then the following message is issued:

Process Steps	Messages
	Send Email should not be blank. • If Send Email is entered other than Yes or No then following error message is issued: Please enter either Yes or No for Send Email. • If the Send SMS parameter is not entered, then the following message is issued: Send SMS should not be blank. • If Send SMS is entered other than Yes or No then following error message is issued: Please enter either Yes or No for Send SMS. • If Expired Alerts Clean Up Message is not entered, then following message is issued: Expired Alerts Clean Up Message should not be blank.
2. Selection of Records	Select all alert records for recipients that have an alert end date less than the Alert Purge Date.
3. Deletion of Records	Delete all alert records for recipients that have an alert end date prior to the Alert Purge Date.
4. Notification	Create a new Alert in the ADV_ALERTS table indicating that alert purge has occurred.
5. Job Complete	Job processing complete.

When to Run

The job can be run on demand.

Restartability Information

This job does not maintain any checkpoints. If the job encounters any errors while synchronizing data, then you can view the resulting errors in the job log and schedule a new job after rectifying the issue.

Major Input

None

Batch Parameters

Parameter	Description	Default Value	Explanation
Alert Purge Date	Alert Purge Date	No Default	Date entered by the user to delete all the expired alerts prior to this date.
Send Email (Yes/No)	Send Email (Yes/No)	Yes	If *Yes* is entered for this parameter, then an email is sent to the users in the system. If *No* is entered, then an email notification is not sent. Default value is *Yes*.
Send SMS (Yes/No)	Send SMS (Yes/No)	Yes	If *Yes* is entered for this parameter, then an SMS is sent to the users in the system. If *No* is entered, then SMS notification is not sent. Default value is *Yes*.
Expired Alert Clean Up Message	Expired Alert Clean Up Message	All the expired alerts prior to the alert purge date have been removed.	This message will be the subject of the email notification mail and also the alert notification text. The default value is "All the expired alerts prior to the alert purge date have been removed."

Major Output

It deletes all expired alert records for users. Also, an alert notification is sent to all users in the system. Also, an email notification is sent if the Send Email parameter is set to *Yes*.

Job Return Code

Return Code	Condition
Successful (1)	Deletes all expired alerts prior to the Alert Purge Date.
Failed (12)	The job may fail under the following condition: • If the records are not deleted successfully.
Warning (4)	If there are no records to purge, then the following warning message is issued: • No Records found to purge.

Sort Criteria

None

Problem Resolution

If the job fails, submit a new job.

2.2.8 Expired Security Role Clean Up

Chain or Job Name	Expired Security Role Clean Up
Recommended Frequency	This job can be run on demand.
Single Instance Required	No
Can be restarted?	No
Reports generated	Yes

Overview

The Expired Security Role Clean Up batch job is used to remove the security role associated with a user record, based on the Effective To date. This job automates the process of deleting the expired security role from the user record, based on the Effective To date. When a delegated security role is deleted from a user record via the Expired Security Role Clean Up batch job, an e-mail notification is sent to the registered e-mail address of a user record. The subject and the body of the e-mail can be configured on the existing Notification Template reference page. This job can be run in three modes: *Report*, *Update*, and *Both* (Report and Update). The job will generate the report in *Report* mode and *Both* mode, and deletes the records in *Update* mode. The job can be run as per site requirement.

The following table shows the various steps that the Expired Security Role Clean Up job goes through and the messages issued at each step:

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters. • If Report Name is not provided, then the following message is issued: • “Report Name is empty; therefore, Report Name has defaulted to: Expired Security Role Clean Up” • If Mode is not provided, then the following message is issued: • “Mode cannot be blank; therefore, Mode has defaulted to: Report Mode” • If the Mode is not a numeric value, then the following message is issued: • "Mode should be Numeric and should be in between 1 and 3; therefore, Mode has defaulted to 1, which is Report Mode"; • If the Mode value is not in the range of 1 to 3, then the following message is issued: • “Mode should be Numeric and should be in between 1 and 3; therefore, the Mode has defaulted to 1, which is Report Mode.”

Process Steps	Messages
	- When all of the parameters are validated successfully, then the following message is issued: • "Parameter Validation Completed Successfully."
2. Selection of Records	Selects all of the security roles associated to user records for which the Effective To date is less than the current date, that is, all of the expired security roles associated with the user.
3. Process Records	Processing is based on the Mode selected: *1-Report*: The system generates a report that provides the details of all expired security roles for which the Effective To date is less than the current date. *2-Update*: The system deletes all of the expired security roles for which the Effective To date is less than the current date. *3-Both*: The system deletes all of the expired security roles for which the Effective To date is less than the current date and the system also generates a report with details of the expired security roles.
4. Job Complete	Job processing complete.

When to Run

The job can be run on demand.

Restartability Information

This job does not maintain any checkpoints. If the job encounters any errors while synchronizing data, then you can view the resulting errors in the job log and schedule a new job after rectifying the issue. The selection logic ensures that the previously processed records are not picked up again for processing.

Major Input

- R_SC_USER_ROLE_LNK – Security User Role Link
- R_SC_USER_DIR_INFO – Security User Info

Batch Parameters

Parameter	Description	Default Value	Explanation
REPORT_NM	Report Name	No Default	The name of the generated report. If this parameter is blank during the job run, then the

			following title is used on the report: “Expired Security Role Clean Up”.
MODE	Mode	1	The Mode can be set to one of the following values: 1 = Report 2 = Update 3 = Both

Major Output

PDF Report / HTML Report

Job Return Code

Return Code	Condition
Successful (1)	This Return Code is issued under the following conditions: • The report was generated successfully. • On the successful validation of all of the user records and update action.
Warning (4)	This Return Code is issued under the following condition: • While running the batch job, if no eligible records are found, the system completes the batch job and the job issues a Return Code of *Warning*.
Non Fatal Error (8)	This Return Code is issued under the following conditions: • While running the batch job, if only a partial number of records are processed by the system, the system completes the batch job and the job issues a Return Code of *Non-Fatal Error*.
Failed (12)	The job may fail under the following conditions: • Parameters are invalid. • Runtime exceptions for unexpected situations.
Terminated (16)	This Return Code is issued when the job is terminated by the user.
System Failure (20)	This Return Code is issued when the job is terminated because of database server or network issues.

Sort Criteria

User ID

Selection Criteria

The job selects and removes all expired security role IDs associated with user records based on the Effective To date.

Problem Resolution

The following table shows the possible return codes and recommendations for each processing step. For further assistance, contact the System Administrator.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	The job ended with a warning, since no eligible records are found. **Sample Message:** Total no. of expired security roles : 0	N/A	N/A
Non Fatal Error (8)	This Return Code is issued, while running the batch job, if only a partial number of records are processed by the system.	N/A	N/A
Failed (12)	The job may fail due to the following conditions: • Parameters are invalid. • Runtime exceptions for unexpected situations.	If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	N/A
Terminated (16)	When the job is terminated manually by the user.	The reason for the termination needs to be investigated before rescheduling the job.	N/A
System Failure (20)	When the job is terminated because of database server or	The reason for the System Failure needs to be investigated before	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
	network issues.	rescheduling the job.	

2.2.9 Flow Job Scheduler

Chain or Job Name	Flow Job Scheduler
Recommended Frequency	This job is indirectly scheduled from the Integration Manager application when a service is scheduled. It should not be run directly from Advantage application
Single Instance Required	No
Can be restarted?	No
Reports generated	No

Overview

The Flow Job Scheduler batch job is used to schedule a service flow on Advantage Business Administration (ABI).

The various steps in the process are as follows:

Process Steps	Messages
1. Parameter Validation	Parameters are valid or invalid depending on the Validation. If the parameter is invalid, the invalid value will be displayed in the log.
2. Scheduling service flow	Job submitted successfully, which includes the Flow Sequence Number for the Service flow that was scheduled on ABI.

When to Run

This job is indirectly scheduled from the Integration Manager Application when a service is scheduled. It should not be run directly from the Advantage application.

Major Input

Parameter	Description	Default Value	Explanation
FLOW_ID	Flow ID Required	No Default	Required identification of a Flow
PARENT_FLOW_SEQ_NO	Parent Flow	-1	Sequence Number Optional
RUN_SERVER_ID	Run Server ID	-1	Optional identification of a server

Parameter	Description	Default Value	Explanation
URL	ABI Web Service connection URL	No Default	Required ABI Web Service connection URL to schedule Service flow on ABI.

Major Output

- Schedules a Service flow on ABI

Job Return Code

The following table shows the potential job return codes for the Flow Job Scheduler job.

Return Code	Condition
Successful (1)	Parameters provided are valid.
Failed (12)	Parameters provided are invalid.

Sort Criteria

No sorting criterion is being used.

Selection Criteria

No selection criterion is being used since it is indirectly scheduled from the Integration Manager application.

Problem Resolution

If the job fails, correct the problem and schedule the Service from the Integration Manager application, which submits a new job.

2.2.10 Job Parameters Updater

Chain or Job Name	Job Parameters Updater
Recommended Frequency	Daily, Weekly, Monthly or Specific.
Single Instance Required	No
Can be restarted?	No
Reports generated	None

Overview

While running regular frequency offline batch cycles, there is a need to update job input parameters for all jobs. The Job Parameters Updater job updates the batch input parameter values for all such regular jobs based on the setup done on the Job Updater table. This helps in reducing the manual effort in setting up the jobs for execution.

Major Input

The Job Updater (JOBUPD) page can be used to set up data to be picked up by the Job Parameter Updater batch job. This data is a set of table row columns to be updated for specific jobs, which run regularly in a schedule. Each record on this table contains information related to the job input parameter that needs to be updated. The parameter values on this page are defined in the Job Updater Parameter Values page.

The Job Updater Parameter Values (JOBUPDPV) page provides a capability to define literal values as well as light weight expressions for setting variable values based on certain conditions. Please refer to the System Processing Guide for more information.

Batch Parameters

None

Major Output

- Updates done in table row columns defined in Job Updater.

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	When one or more records being processed issue an error.

Return Code	Condition
Non Fatal Error (8)	No eligible records found.
Failed (12)	The job will fail under the following conditions: • Invalid data in Job Updater (JOBUPD) or Job Updater Parameter Values table. • Run time exceptions for unexpected situations.
Terminated (16)	This return code is issued when the job is terminated by the user.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues.

2.2.11 Lock Idle User

Chain or Job Name	Lock Idle User
Recommended Frequency	The Lock Idle User job can be run on demand
Single Instance Required	No
Can be restarted?	No
Reports generated	Yes

Overview

The Lock Idle User job locks or inactivates Idle user accounts (that is, users whom have not logged into the system for more than a specified number of days). If the difference between the Current System Date and the Last Login Date is greater than the value in the Idle Account Active Days field on the Security Configuration reference page, then the Locked Out flag is set to *Locked* by the Lock Idle User process. You can also generate a report that contains the account details of users that satisfy the threshold condition. The job can be run daily during the nightly cycle or as per site requirement.

The following table shows the various steps that the Lock Idle User job goes through and the messages issued at each step:

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters. • If REPORT NAME is not provided, then the following message is issued: • “Report Name is empty; therefore, Report Name has defaulted to: Idle User Account Report” • If Mode is not provided, then the following message is issued: • “Mode cannot be blank; therefore, Mode has defaulted to: Report Mode” • If the Mode is not a numeric value, then the following message is issued: • "Mode should be Numeric and should be in between 1 and 3; therefore, Mode has defaulted to 1, which is Report Mode"; • If the Mode value is not in the range of 1 to 3, then the following message is issued: • “Mode should be Numeric and should be in between 1 and 3; therefore, the Mode has defaulted to 1, which is Report Mode.” • If the Idle Account Active Days parameter value is

Process Steps	Messages
	negative or greater than or equal to 1000, then the following message is issued: • "The Idle Account Active Days parameter must be between a range of 0 - 1000." • If the Idle Account Active Days parameter value is not numeric, then the following message is issued: • "Only a whole number greater than or equal to 0 is accepted for the Idle Account Active Days parameter." • When all of the parameters are validated successfully, then the following message is issued: • "Parameter Validation Completed Successfully."
2. Selection of Records	Selects all user records in the system where the Last Login Date is not empty and which satisfies the threshold limit condition. The value of Idle Account Active Days should be less than the difference between the Last Login Date and the Current System Date.
3. Process Records	Processing is based on the Mode selected: 1-Report: The system generates a report that provides the details of all user accounts that have exceeded the threshold limit. 2-Update: The system sets the Locked Out flag to *Locked* on the Security Configuration (SCCNFG) page for user accounts that have exceeded the threshold limit. 3-Both: The system sets the Locked Out flag to *Locked* on the Security Configuration (SCCNFG) page for user accounts that have exceeded the threshold limit and the system generates a report with details of updated user accounts. Note: If the Idle Account Active Days parameter value is *0*, then the system will not lock any user accounts.
4. Job complete	Job processing complete.

When to Run

The job can be run on demand.

Restartability Information

The job does not maintain any checkpoints. If the job encounters any errors while synchronizing data, then you can view the resulting errors in the job log and schedule a new job after rectifying the issue. The selection logic ensures that the previously processed records are not picked up again for processing.

Major Input

- SCUSER (R_SC_USER_INFO, R_SC_USER_DIR_INFO) - User Information
- SCCNFG (R_SC_SEC_CNFG) - Security Configuration

Batch Parameters

Parameter	Description	Default Value	Explanation
REPORT_NM	Report Name	No default	The name of the generated report. If this parameter is blank when the job is run, then the following title is used on the report: Idle User Account Report.
IDLE_ACCNT_ACTV_DY	Idle Account Active Days	60	This parameter is only used if the Mode is 1, which is Report mode. For Modes 2 (Update) and 3 (Both), the value from the Security Configuration reference page is used.
MODE	Mode	1	The Mode can be set to one of the following values: 1 = Report 2 = Update 3 = Both

Major Output

PDF Report / HTML Report

Job Return Code

Return Code Condition	Return Code Condition
Successful (1)	This Return Code is issued under the following condition: • The report was generated successfully. • Records are returned that satisfy the threshold conditions.

Return Code Condition	Return Code Condition
Warning (4)	This Return Code is issued under the following condition: • None of the records satisfies the threshold conditions.
Non-Fatal Error (8)	This job does not issue this Return Code.
Failed (12)	The job may fail under the following conditions: • Run time exceptions for unexpected situations • Invalid value is provided for the Idle Account Active Days field
Terminated (16)	This Return Code is issued when the job is terminated by the user.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues.

Sort Criteria

User ID

Selection Criteria

The job selects all user records where the difference between the Current System Date and the Last Login Date is greater than the value in the Idle Account Active Days field on the Security Configuration page. Note: The value of the Idle Account Active Days parameter for the Lock Idle User job is used if the job is run in Report Mode (1).

Problem Resolution

The following table shows the possible return codes and recommendations for each processing step. For further assistance, contact the System Administrator.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	Job ended with a Warning because none of the records meet the threshold condition.	N/A	N/A

Possible Return Codes	**Condition**	**Recommendation**	**Other Instructions**
	Sample Message: Total Users Locked: 0		
Non-Fatal Error (8)	This job does not issue this Return Code.	N/A	N/A
Failed (12)	The Parameter Value provided for Idle Account Active Days is not a valid value. **Sample Message:** Only a whole number greater than or equal to 0 is accepted for Parameter – Idle Account Active Days.	Enter a valid value for the Idle Account Active Days parameter and then schedule a new job. The value must be a whole number greater than 0 and less than 1000.	N/A
Terminated (16)	The job is terminated manually by the user.	The reason for the termination needs to be investigated before rescheduling the job.	N/A
System Failure (20)	The job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before rescheduling the job.	N/A

2.2.12 Maintain User Configurations

Chain or Job Name	Maintain User Configurations
Recommended Frequency	On demand
Single Instance Required	Yes
Can be restarted?	No
Reports generated	User Configurations Report

Overview

The Maintain User Configurations batch job is provided to update all existing user preferences by comparing with the default page and identifying the delta. It is recommended to run the process once to update existing user preferences. The process can also be executed on demand to verify and update the user preferences after page extensions are applied.

When performing the comparison and identifying the delta, the Maintain User Configurations job takes into account the following extensions:

- For Admin, Financial, HRM, PB, or VSS pages, global extensions under development, pending synchronization, or deployed to PalDb are included in the comparison.
- For Common pages, only global extensions deployed to PalDb are included in the comparison.

The process may be executed in two separate modes. It is recommended that the process be executed first in *Report Mode* and then in *Report and Update Mode.*

- *Report Mode* to select and process user configurations to list the calculated modifications on the report.
- *Report and Update Mode* to select and process user configurations to list the calculated modifications on the report and update these calculated modifications to the user configurations record. Any records where modifications result in no difference, then such records are deleted from the User Configurations table.

Note: The report contains User ID information. Your site should implement security to limit access.

Steps for running process

1. Parameter Validation
2. Select User Configuration records
3. Calculate modifications or delta of user configurations
    a. Compares current page metadata for the selected card (search or grid) with the snapshot stored on the User Configurations record.
    b. Differences are captured as modifications or delta to metadata.
4. Generate report

   a. For each selected record, lists the calculated metadata difference.
   b. If any record is identified with no difference, then it is marked for deletion on the report.

5. Update and delete User Configuration records based on the report
   a. For each record listed on the report, the process updates user configuration information with the calculated difference.
   b. Deletes the user configuration record if marked for deletion on the report.

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters • Parameters are valid or invalid depending on the Validation. • Batch Parameter validation completed
2. Selection of Records	• Selecting eligible records • If the selection returns 0 records, then the following message is issued: “No records found for processing”. • If the execution of query failed, then the following message will be issued: “Error encountered while retrieving User Configuration records”.
3. Calculate modifications or delta of user configurations	If any error while accessing page metadata for the component, then the following message is issued: • Error while accessing page or component for record ID <ID> • Error while parsing card user configuration data for record ID <ID> • Error encountered while processing user configuration record <ID> Following progression messages are issued: • <Count> records are processed • Total records processed: <Count> • Total records failed to process: <Error Count>
4. Generate report	If no records are identified to report, then the following message is issued: • No records to report or update If any error while processing the report, then the following message is issued:

Process Steps	Messages
	• Error encountered while generating report and updating user configuration records • Error encountered while writing user configuration data to the report • Error encountered while writing overflow data to the report • Error encountered while reporting or updating user configurations
5. Update and delete User Configuration records based on report	If any error while processing records, then the following message is issued: • Error encountered while reporting or updating user configurations

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

Major Input

- User Configurations (ADV_USER_CONFIG)
- Optimized metadata for application (PalDB files)

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

Parameter	Description	Default Value
Application (APPLICATION)	Indicates the application where the Maintain User Configurations job will be executed. The default value cannot be overridden.	Administration Job: adm Financial Job: fin HRM Job: hrm VSS Job: vss Performance Budgeting Job: pb
Client Name (CLIENT_NM)	Client Name	No Default
Commit block size (COMMIT_SIZE)	Commit block size	100
Mode Indicator (MODE_IND)	Mode Indicator. Valid values are: 1 - Report mode or 2 - Report and Update Mode	1

Major Output

- User Configurations (ADV_USER_CONFIG)
- User Configurations Report

Note: Since the Performance Budgeting job is run from the Administrator application, existing report security prevents you from accessing the PB report from the Report output page. You can set up File Transfer Folder (FOLDTL) to refer to the location, where the User Configuration Report output is stored and set up File Transfer Role (FTROLE) to enable authorized users to use File Transfer Maintenance (FUD) to access the PB reports.

Job Return Code

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

Return Code	Condition
Successful (1)	All of the selected user configuration records are processed successfully.
Warning (4)	No eligible records found. This could be because of the following reasons: • No records found on the User Configurations Table for the selected application.
Non-Fatal Error (8)	When the calculation of differences or delta of user configuration failed for at least one record and remaining are processed successfully.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid. • Run time exceptions for unexpected situations.
Terminated (16)	This Return Code is issued when the job is terminated by the user.
System Failure (20)	This Return Code is issued when the job is terminated because of database server or network issues.

Sort Criteria

Records written to the report are sorted by:

- Page Key (PAGE_CODE)
- Card Key (CARD_KEY)
- User ID (USER_ID)

Selection Criteria

Select User Configurations (ADV_USER_CONFIG) records where the first part of the Page Key (PAGE_CODE) matches any value from the Application (APPLICATION) parameter appended by ".page." or ".apgs."

For example, if the Application Name (APPLICATION) parameter is provided as "adm" then user configuration records are selected as,

PAGE_CODE LIKE 'adm.page.%' OR

PAGE_CODE LIKE 'adm.apgs.%'

Problem Resolution

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully. All of the selected records are processed and reported successfully.	N/A	N/A
Warning (4)	Job ended with a Warning because there is no user configuration record that matches the parameter value for the Application name. **Sample Message:** No records found for processing No records to report or update	Ensure there are records on the User Configurations table for the provided parameter value and schedule a new job.	N/A
Non-Fatal Error (8)	This Return Code is issued when the calculation of differences or the delta of user configuration failed for at least one record and the remaining	The job would have skipped those records that have failed to access page or component metadata. This could be for inactive components, which might be accessible through conditional	NA

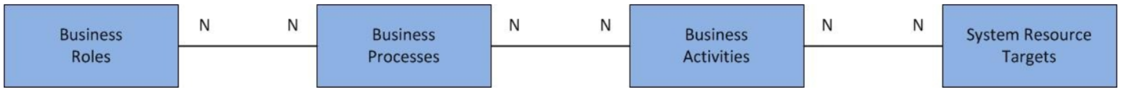
Possible Return Codes	Condition	Recommendation	Other Instructions
	records are processed successfully. **Sample Message:** • Error while accessing page or component for record ID <ID> • Error while parsing card user configuration data for record ID <ID>	extensions. For example, available through Business Role extension or any conditional logic. These records are not listed on the generated report or are processed for updates. Ensure that listed records are valid and schedule a new job.	
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following condition. • Encounters any runtime exceptions If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error, and schedule a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The new job can be scheduled.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The new job can be scheduled.	N/A

2.2.13 Menu Generator

Chain or Job Name	Menu Generator
Recommended Frequency	This should be run when there is any change to the Business Role configuration.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	None

Overview

CGI Advantage has the concept of Business Roles to help organize System Resource Targets (for example, reference pages or inquiries) for navigational purposes. There is a hierarchy of Business Role data to help organize the resources into functional groupings. The hierarchy is as follows:



The system then uses this configuration and creates two elements of role based UI: primary navigation menus that expand from the left and multiple business process landing pages.

This batch job in particular uses the configuration data and creates page metadata in the form of xml for Menu Pages and Business Process Landing Pages. This also creates the extensions for Home Pages. This XML will then be used by the Key Value Metadata generator process and add all the respective pages and extensions into the PALDB. Below is the set of steps that the process would perform.

1. Select all active records from R_BUS_ROLE.
2. For all active records, get all active records from R_BUSROLE_BUSPROC.
3. For all active records, get all active records from R_BUSPROC_BUSACTV.
4. For all active records, get all active R_BUSACTV_RSRCTRGT.
5. Create Menu XML for all Business Roles and add necessary extensions in the Application.xml. The Menu xml will have all the metadata information. While creating metadata for the role based pages Menu Generator will filter out special characters such as comma (,), period (.), single quote (‘), or double quote (“) as part of the Name attribute. Therefore, if the application has existing extensions in place for role based pages, and if the extensions use the listed special characters, then those need to be removed from the extension metadata Name attribute.
6. Create Business Process Landing page for all Business processes and also add page entries in the Application.xml and CrossReference.xml.

7. Create Home Page extensions in application.xml.
    a. There is one out-of-the-box mobile home page delivered. As only one mobile home page is allowed to be configured, this home page extension is always generated and registered in application.xml.

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

Process Steps	Messages
1. Parameter Validation	Validating Batch Parameters. Validate if the Meta Data Path parameter is empty. If empty then raise an error.
2. Select Records	Select records as per the selection criteria listed in the Selection Criteria section**.**
3. XML creation for Menu Pages	Create Menu Pages for all active roles. The Menu Pages will have all the necessary hierarchy of Business Process/ Business Activities and Business Roles.
4. XML creation for Business Process Landing.	Create Business Process Landing Pages for all Business Processes within every role.
5. Home Page extension	Create Home Page extensions for all Business Roles and Home page combinations.
6. Job complete	Run ended.

Restartability Information

The job has to be rerun. It does not store any checkpoints and cannot be restarted.

Major Input

- Business Role (R_BUS_ROLE)
- Business Process (R_BUS_PROC)
- Business Activities (R_BUS_ACTV)
- Resource Target (R_RSRC_TRGT)
- Business Role Association (R_BUSROLE_BUSPROC , R_BUSPROC_BUSACTV, R_BUSACTV_RSRCTRGT)
- Home Page Configuration (R_HOME_PAGE_CNFG)

Batch Parameters

Parameter	Description	Default Value	Explanation
META_DATA_PATH	The path where page metadata folder is stored. This job creates xml metadata under that folder.	$$AMSROOT$$/ApplicationMetadata/metadata/common	This batch job creates XML as the output and writes to the location specified under this parameter.
GENERATE_HOME_PAGE	Valid values are: true (generate home pages) or false (do not generate home pages)	false	Indicates whether to generate home page. Note that the job always creates an extension for Mobile Home Page irrespective of this parameter value as there is only one Mobile Home Page and it is not associated with any Business Role.

Major Output

- Creates XML metadata extensions for Menu Pages based on Business Role.
- Creates Home Page extensions based on Business Role.
- Creates Home Page extension for Mobile Home Page
- Creates Business Process Landing Page metadata.

Job Return Code

Return Code	Condition
Successful (1)	All records processed and xml metadata created successfully.
Warning (4)	This job does not use this return code.
Non-Fatal Error (8)	This job does not use this return code.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Run time exceptions for unexpected situations
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

- The Business Role Processes are sorted by DISP_ORD
- The Business Process Activities are sorted by DISP_ORD
- The Business Activities Resources are sorted by GROUPING and RSRC_DISP_ORD.

Selection Criteria

This batch job selects the records in the following sequence.

- Select all active Roles from R_BUS_ROLE and for all active roles get all the active Business Role / Business process combination (R_BUSROLE_BUSPROC) records.
- For all active records from the above selection, get all active Business Process and Activity combinations (R_BUSPROC_BUSACTV) and also all the Activity Resource combinations (R_BUSACTV_RSRCTRGT).
    - Additionally, selects Mobile Home Page along with Home Pages associated with selected Business Roles.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All the necessary Menu and Business process landing page XML metadata is created successfully for all roles. The Home pages are also added for each role.	N/A	N/A
Failed (12)	If the batch job fails then the batch job has to be rerun or the System Administrator has to be notified.	Rectify the problem and schedule a new job.	View logs for help on the problem.

2.2.14 Multi-threaded Table Loader

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

Overview

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

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

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

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

Process Details

Logic:

STEP-1 Parameter Validation:

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

STEP-2 Splitting the Input File:

In this step the job will split the input file in to multiple smaller files. If the count of split files is greater than zero the following message will be logged in the job log: "File split done:

count=<number>". If the count is not greater than zero then the job will terminate and issue an error stating "No records found for processing" within the job log.

STEP-3 Child Job processing:

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

STEP-4 Status of Child Job scheduled for processing:

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

STEP-5 Deletion of temporary Files created during processing:

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

Job Not Restartable

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

Major Input

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

Major Output

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

Batch Parameters

Parameter Name	Internal Parameter Name	Description	Default Value
Block Size	BLOCK_SIZE	Number of records in each split segment of the input file. This parameter is not required.	1
Commit Block Size	COMMIT_BLOCK_SIZE	Number of records to commit at a time. This parameter is not required.	The value for the property – XMLExportCommitBlockSize – in ADV30Params.ini
File Input Directory	FILE_INPUT_DIR	The location of the input file. This parameter is not required.	The value for the property – XMLExportFileLocation– in ADV30Params.ini
File List	FILE_LIST	Comma separated list of files to be uploaded in multi-threaded processing. This parameter is required.	None
File Output Directory	FILE_OUTPUT_DIR	Output location for the file segments. This parameter is not required.	The value for the property – XMLExportFileLocation– in ADV30Params.ini
File Prefix	FILE_PREFIX	Prefix used on the filenames for the output file segments. This parameter is not required.	None (blank string)
Log Status Interval	LOG_STATUS_INTERVAL	Logging frequency (in seconds) for controller thread reporting status of child threads to the system log. This parameter is not required.	300 Seconds (5 minutes). Cannot be less than 30 seconds. The system will use the default of 300 if a value less than 30 is specified.
Sleep Interval	SLEEP_INTERVAL	Polling frequency (in seconds) for internal controller thread for checking child processes. This parameter is not required.	5 seconds. A value less than 5 seconds is not used (default value used instead)
System	SMU_CTLG_ID	Catalog id of the System	3

Maintenance Utility Catalog ID		Maintenance Utility job which is spawned as the child process. This parameter is required.	
Stagger Time	STAGGER_TIME	The lag time, in seconds, between the spawning of each child process. This parameter is not required.	30 seconds. Valid values >= 1 second- Default value used otherwise
Thread Count	THREAD_COUNT	Number of threads to use for processing. This parameter is not required.	1

Additional Job Parameters

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

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

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

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

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

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

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

Job Return Code

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

Return Code	Condition
Successful (1)	All of the table records are imported successfully.
Warning (4)	Not applicable for this job
Non-Fatal (8)	This job will issue a non-fatal error for the following: • Error while importing the input record files during the child job processes (System Maintenance Utility). For example, if attempting to import a record that already exists in the target table.
Failed (12)	The job will fail under the following conditions: • Invalid Parameters • Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

There is not any sort criteria.

Selection Criteria

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

Problem Resolution

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Non-Fatal (8)	Job Failed due to Non-fatal condition: Duplicate record being imported to target table	No action is needed. The job will continue and will import all other records (assuming they do not exist on the target table).	N/A
Failed (12)	Job failed due to Fatal conditions. Sample Message: "No input file to process"	In this step, the job can fail under the following conditions: 1) Job Parameter error 2) Encounters any runtime exceptions and 3) Failed during child job. If the Job Parameter input was causing the error, examine the error message in the job log, correct the job parameter. You must then reschedule the job. If the job fails because of runtime exceptions, investigate the exception reported by the process, resolve the error. You must reschedule the job. For child job errors, refer to the run-sheet for the System Maintenance Utility job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. When the reason for termination has been determined, you must reschedule the job.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. When the reason for System Failure has been determined, you must reschedule the job.	N/A

2.2.15 Multi-threaded Transaction Loader

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

Overview

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

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

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

Process Details

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

Process mode 1: Import

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

Process mode 2: Import & Submit

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

Process mode 3: Import & Transaction Other Action Mode

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

Process mode 4: Import & Validate

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

Logic:

STEP-1 Parameter Validation:

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

STEP-2 Splitting the Input File:

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

STEP-3 Child Job processing:

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

STEP-4 Status of Child Job scheduled for processing:

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

STEP-5 Exception Report:

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

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

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

Restartability Information

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

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

Major Input

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

Major Output

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

Job parameters

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

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

			instead)
System Maintenance Utility Catalog ID	SMU_CTLG_ID	Catalog id of the System Maintenance Utility job which is spawned as the child process. This parameter is required.	No Default for this parameter.
Stagger Time	STAGGER_TIME	The lag time, in seconds, between the spawning of each child process. This parameter is not required.	30 seconds. Valid values >= 1 second- Default value used otherwise
System Maintenance Utility Exception File for Import and Submit Mode	S_SMU_EXCEP_REP_FILE_NM	Name of exception file for System Maintenance Utility in Import and Submit Mode. This parameter is not required.	None
System Maintenance Utility Flag to indicate if failed error messages should be recorded in Import Mode	I_SMU_DTL_IMP_MSG_FL	Indicates whether error message details for each failed record should be recorded in the error file in Import Mode	True
Thread Count	THREAD_COUNT	Number of threads to use for processing. This parameter is not required.	1
File Polling Interval(Seconds)	FILE_POLLING_INTERVAL	The time interval taken (in seconds) by this batch job after every polling is committed to the Export-Import folder looking for the input file.	0
File Polling Max Time(Seconds)	FILE_POLLING_MAX_TIME	The maximum time (in seconds) until when the batch process will keep polling to the Export-Import folder looking for the input file.	0

Additional Job Parameters

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

- When these parameters are added they need to be prefixed as follows:
  - **I_SMU_**: Parameters passed to SMU jobs in Import Mode.
  - **S_SMU_**: Parameters passed to SMU jobs in Import and Submit Mode.
  - **O_SMU_**: Parameters passed to SMU jobs in Import and Other Action Mode.

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

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

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

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

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

Job Return Code

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

Return Code	Condition
Successful (1)	All of the Transaction records are imported successfully.
Warning (4)	N/A
Non-Fatal (8)	This job will issue a non-fatal error for the following: ▪ Error while importing records that already exist in the application.
Failed (12)	The job will fail under the following conditions: • Invalid Parameters

Return Code	Condition
	- Run time exceptions for unexpected situations.
Terminated (16)	This return code will be issued when the job is terminated by the user.
System Failure (20)	This return code will be issued when the job is terminated because of database server or network issues.

Sort Criteria

There is not any sort criteria.

Selection Criteria

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

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully	N/A	N/A
Warning (4)	N/A	N/A	N/A
Non-Fatal (8)	Job Failed due to Non-fatal condition: Duplicate record being imported to application	No action is needed. The job will continue and will import all other records (assuming they do not exist within the application).	N/A
Failed (12)	Job failed due to Fatal conditions. Sample Message: "Invalid value received for parameter MODE (expected values 1,2 or 3): aBcD"	In this step, the job can fail under the following conditions: 1) Job Parameter error 2) Encounters any runtime exceptions and 3) Failed during child job. If the Job Parameter input was causing the error,	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.

Possible Return Codes	Condition	Recommendation	Other Instructions
	"No input file to process"	examine the error message in the job log, correct the job parameter, then restart the job. If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job. For child job errors, refer to the runsheet for the System Maintenance Utility job.	
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The job can either be restarted or schedule a new job.	If another instance of the job has already been scheduled and ran successfully, then this job should not be restarted – only a new job should be scheduled.

2.2.16 Password Expire Email Notification

Chain or Job Name	Password Expire Email Notification
Recommended Frequency	The Password Expire Email Notification can be run on demand or daily.
Single Instance Required	No
Can be restarted?	No
Reports generated	No

Overview

The Password Expire Email Notification batch process will send emails to users whose password is about to expire. An email notification will be sent to the user based on the number of days that has been set in the Password Warn Day Count field on the Security Configuration table or the password warn day count value that has been sent through the PSWD_WARN_DY_CT job parameter.

The following table shows the various steps that the Password Expire Email Notification batch process goes through and the messages issued at each step:

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters. • If the User Id is not Valid, then the following message is issued: • "USER_ID+: is not a valid user." • If the Password Warn Day Count (PSWD_WARN_DY_CT) is not a numeric value, then the following message is issued: • "Password warn day count should be a number." • When all parameters are validated successfully, then the following message is issued: • “Parameter Validation Completed Successfully.”
2. Selection of Records	The selection criteria for this process selects records from the Search by Organization (R_SC_USER_INFO) and User Search by Name (R_SC_USER_DIR_INFO) tables based on Password Change Date (PSWD_CHG_DT) and Password Warn Day Count (PSWD_WARN_DY_CT) fields.
3. Process Records	Processing is based on the Password Warn Day Count selected: The Password Expire Email Notification batch process will send emails to users whose password is about to expire.

Process Steps	Messages
	Password Warn Day Count can be specified while scheduling a job and notifies the corresponding users with an email accordingly.
4. Job complete	Job processing complete.

When to Run

This process can be run On Demand, or Daily.

Restartability Information

The job does not maintain any checkpoints. If the job encounters any errors while synchronizing data, then you can view the resulting errors in the job log and schedule a new job after rectifying the issue. The selection logic ensures that the previously processed records are not picked up again for processing.

Major Input

- SCUSER (R_SC_USER_INFO, R_SC_USER_DIR_INFO) - User Information
- SCCNFG (R_SC_SEC_CNFG) - Security Configuration
- The Email notification text can be customized via the Notifications (NOTIF) page by updating the record with a Notification ID of PW_EXP_MSG.

Batch Parameters

Parameter	Description	Default Value	Explanation
EMAIL_FROM_ADD	Email From Address	do-not-reply@cgi.com	From address to send an email (by default it will be do-not-reply@cgi.com unless specified during scheduling a job).
PSWD_WARN_DY_CT	Password Warn Day Count	No default	Password Warn Day Count (number of days before password expiry, that an email notification must be sent)
USER_ID	User ID	No default	The User ID for which the upcoming expiration of password needs verified. Multiple User IDs with comma separated values are also supported. If a User ID

Parameter	Description	Default Value	Explanation
			is not provided the job will run for all users whose password is about to expire.

Major Output

Notification emails will be sent to users.

Note: Generating a PDF report is currently not available with this job.

Job Return Code

Return Code Condition	Return Code Condition
Successful (1)	Job runs successfully under the following conditions: • Successful parameter validation • All of the selected records are processed successfully
Failed (12)	Job will fail under the following conditions: • Parameters are invalid. • Run time exceptions are encountered while processing the records. • Selected records failed to process successfully
Terminated (16)	This return code is issued when the job is terminated by the user.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues.

Sort Criteria

The records from the R_SC_USER_INFO and R_SC_USER_DIR_INFO tables will be sorted on the basis of the Password Change Date (PSWD_CHG_DT) and Password Warns Day Count (PSWD_WARN_DY_CT) fields.

Selection Criteria

The selection criteria for this process is that it will pick up records from the Search by Organization (R_SC_USER_INFO) and User Search by Name (R_SC_USER_DIR_INFO) tables based on Password Change Date (PSWD_CHG_DT) and Password Warns Day Count (PSWD_WARN_DY_CT) fields.

Problem Resolution

If the process was discontinued for any reason, then the job needs to be rescheduled again.

2.2.17 Populate Advantage Metadata

Chain or Job Name	Populate Advantage Metadata
Recommended Frequency	The new job should be run under three circumstances: 1. A new transaction type is added to Transaction Control, 2. A new table with the attachment feature has been added, or 3. A transaction or table's columns are modified which supports the attachment feature.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

The Populate Advantage Metadata job is used to populate the Advantage Type (R_ADV_SYS_TYP) and Advantage Metadata (R_ADV_FLDS_INFO) tables. This new job is used to register all transaction and table fields from all parts of Advantage (e.g. Financial and Human Resource Management. The most common use of the Populate Advantage Metadata Job will be for ECM processing.

For example, the user wants to get all of the possible list of attributes which he/she can use to setup the ECM Mapping table for the Invoice, the user will run the job for the Transaction Type of IN. After the job runs successfully, the Advantage Metadata will be updated with the list of all attributes related to the IN Advantage Transaction Type.

When to Run

The new job should be run under three circumstances:

1. A new transaction type is added to Transaction Control
2. A new table with the attachment feature has been added
3. A transaction or table's columns are modified which supports the attachment feature

Major Input

- Transaction Type (R_DOC_TYP)

Batch Parameters

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

Parameter	**Description**	**Default Value**	**Explanation**
DOC_TYPES	Transaction Types	No Default	Optional. A comma

			separated list of transaction types to be processed for the application. 'All' can be provided to process all Transaction Types from Transaction Type reference page. If left blank, no transaction types will be populated into the Advantage Metadata table.
TBL_NM	Transaction Component Table Name	No Default	Optional. A comma separated list of names of the tables to be processed for the application. If left blank, no tables will be populated into the Advantage Metadata table.

Major Output

- ADVANTAGE Type (R_ADV_SYS_TYP)
- ADVANTAGE Metadata (R_ADV_FLDS_INFO)

Job Return Code

The following table shows the potential job return codes for the Populate Advantage Metadata job.

Return Code	Condition
Successful (1)	All of the selected records are processed successfully.
Warning (4)	When one or more transaction types or tables being processed throw an error.
Non Fatal Error (8)	Not applicable.
Failed (12)	The job will fail under the following conditions: - Parameters are invalid. - Run time exceptions for unexpected situations.
Terminated (16)	This return code is issued when the job is terminated by the user.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues.

Sort Criteria

No sorting criterion is being used.

Selection Criteria

Selection of the records is based on the following criteria:

- In the job parameter, if the parameter value is specified in the TBL_NM job parameter, then fields from the TBL_NM table are used for processing else no Table records are processed.
- In the job parameter, if the parameter value is specified in the DOC_TYPES job parameter, then DOC_TYPES transaction are used for processing. If 'All' is specified in DOC_TYPES, all Transaction Types from the Transaction Type table are taken as the default list of transactions for processing.

Problem Resolution

If the Populate Advantage Meta Data job runs unsuccessfully, correct the problem and submit a new job.

2.2.18 Populate Business Object Metadata

Chain or Job Name	Populate Business Object Metadata
Recommended Frequency	On demand
Single Instance Required	Yes
Can be restarted?	No
Reports generated	No

Overview

The Populate Business Object Metadata batch job is provided to load or update the Application Business Object Registration table for the selected application. The Application Business Object Registration table is a registry of business objects for CGI Advantage applications.

It is recommended that this process be executed while deploying a new release/container that includes any newly introduced business objects or drops business objects that are not required.

Steps for running process:

1. Parameter Validation
2. Delete Records
   a. If any existing application specific business object entries on the Application Business Object Registration table
3. Add application specific business object entries on the Application Business Object Registration table

Process Steps	**Messages**
1. Parameter Validation	• Validating batch parameters • Parameters are valid or invalid depending on the validation. • Batch Parameter validation completed.
2. Delete Records	• Deleting records from Application Business Object Registration table for application • No records found in Application Business Object Registration table to delete for application • Deleted <Count> records from Application Business Object Registration table for application
3. Add Records	If any error while adding record(s) to the Application Business Object Registration table, then the following message will be

Process Steps	Messages
	issued: • Adding records to Application Business Object Registration table has failed Following progression messages are issued: • <Count> records are added to Application Business Object Registration table • Total records added to Application Business Object Registration table: <Count>

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

Major Input

- Application Business Object Registration (IN_APPL_BOS)
- Optimized metadata for application (PalDB files)

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

Parameter	Description	Default Value
Application (APPLICATION)	Maintain User Configurations for application Default value cannot be overridden.	Administration Job: adm Financial Job: fin HRM Job: hrm VSS Job: vss Performance Budgeting Job: pb
Commit block size (COMMIT_SIZE)	Commit block size	100

Major Output

- Application Business Object Registration (IN_APPL_BOS)

Job Return Code

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

Return Code	Condition
Successful (1)	All of the available business objects are added to the Application Business Object Registration table successfully.
Warning (4)	Not Applicable
Non-Fatal Error (8)	Not Applicable
Failed (12)	The job will fail under the following conditions: • Parameters are invalid. • Run time exceptions for unexpected situations.
Terminated (16)	This Return Code is issued when the job is terminated by the user.
System Failure (20)	This Return Code is issued when the job is terminated because of database server or network issues.

Sort Criteria

None

Selection Criteria

The record count is captured from Application Business Object Registration (IN_APPL_BOS) where Application (IN_APPL_BO) matches the Application (APPLICATION) parameter.

If already registered records are found, then the process deletes records from Application Business Object Registration (IN_APPL_BOS) where Application (IN_APPL_BO) matches the Application (APPLICATION) parameter.

Business objects for the selected application are captured from optimized metadata for the application (PalDB files) and a new record is created on Application Business Object Registration (IN_APPL_BOS) with details from optimized metadata.

Problem Resolution

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

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

Possible Return Codes	Condition	Recommendation	Other Instructions
	for application are added on the Application Business Object Registration table successfully.		
Warning (4)	N/A	N/A	N/A
Non Fatal Error (8)	N/A	N/A	NA
Failed (12)	Job failed due to Fatal conditions.	In this step, the job can fail under the following condition. • Encounters any runtime exceptions If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and schedule a new job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated. The new job can be scheduled.	N/A
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated. The new job can be scheduled.	N/A

2.2.19 Populate Transaction Codes

Description

Populate Transaction Codes is used to register all transaction codes for approval processing from the application as defined on the Transaction Control (DCTRL) pages.

When to Run

When a new transaction code is added to Transaction Control.

Major Input

Transaction Control (R_GEN_DOC_CTRL)

Output

Transaction Codes (R_DOC_CD)

Parameters

N/A

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

If the Populate Transaction Codes job runs unsuccessfully, correct the problem and submit a new job.

2.2.20 Populate Workflow Meta Data

Description

Populate Workflow Meta Data is used to register all transaction fields from other applications such as FIN and HRM into the ADMN application. As the name implies, the most common use of the meta data is for approval processing. Information from transaction codes registered on a Transaction Control (DCTRL) reference page is translated into the meta data table (R_WF_XML_DATA). A secondary use of the updates from this program is the use of the Transaction Component Requirements reference page to set maximum record limits.

When to Run

When a new transaction code is added to Transaction Control or when a transaction's fields are modified.

Major Input

- Transaction Control (R_GEN_DOC_CTRL)
- Transaction Type (R_DOC_TYP)

Output

- Approval Meta Data (R_WF_XML_DATA)

Parameters

Parameter	Description	Default Value	Explanation
ALL_DOC_TYPES	All Transaction Types	false	Process all transaction types for the application.
DOC_TYPES	Transaction Types	Null	A comma-separated list of transaction types to be processed for the application.
DOC_CODES	Transaction Codes	Null	A comma-separated list of transaction codes to be processed for the application.
DOC_COMPS	Transaction Components	Null	A comma-separated list of transaction components to be processed for the application.

Sort Sequence

N/A

Selection Criteria

Transaction fields will be selected for processing:

- If ALL_DOC_TYPES is *true* - All fields from all transaction types and all transaction codes associated with the transaction types will be selected.
- Otherwise, if DOC_TYPES is not null - Only those fields from all transaction codes associated with the transaction types specified will be selected.
- Otherwise, if DOC_CODES is not null - Only those fields from the specified transaction codes will be selected.
- Otherwise, if DOC_COMPS is not null - Only those fields from the specified transaction components for all transaction codes associated with the specified components will be selected.

Problem Resolution

If the Populate Workflow Meta Data job runs unsuccessfully, correct the problem and submit a new job.

2.2.21 RLS Existing Integrated Role Users

Job Name	RLS Existing Integrated Role Users
Recommended Frequency	This job runs for the Row Level Security enhancement to insert the existing user data.
Single Instance Required	No
Can be restarted?	No
Reports generated	No

Overview

The RLS Existing Integrated Role Users job is used to insert all existing user records assigned to the integration security role into the BI User Restriction table for the Row Level Security enhancement.

The following table shows the various steps that the RLS Existing Integrated Role Users job goes through at each step:

Process Steps	Messages
1. Selection of Records	Select all existing users assigned to the integration security role.
2. Insertion of Records	• Insert the selected user records with USER_ID, DEPT_CD, UNIT_CD, RSTR_TYP, APP_ID, and USR_ACTV_FL into the BI_USR_RSTR table and IN_AUD_LOG table. • If the user doesn't have access to the application, the record will not be inserted into the BI_USR_RSTR table. • If the records already exist in the BI_USR_RSTR table, those will not be inserted.
3. Job Complete	Job processing complete.

When to Run

This job should be run to insert all existing user records into the BI User Restriction table, when a site upgrades to the Row Level Security enhancement.

Restartability Information

Schedule a new job if it fails.

Major Input

None

Batch Parameters

None

Major Output

Inserts all existing integrated user records into the BI User Restriction table, if the records do not exist.

Job Return Code

Return Code	Condition
Successful (1)	Inserts all the user records with valid integration security role into the BI User Restriction table successfully.
Failed (12)	The job may fail if records are not inserted successfully.

Sort Criteria

None

Problem Resolution

If the job fails, submit a new job.

2.2.22 Server Data Object (SDO) Cache Report

Job Name	SDO Cache Report
Recommended Frequency	On Demand
Single Instance Required	No
Can be restarted?	No
Reports generated	SDO Cache Report

Overview

The Server Data Object (SDO) Cache Report provides a list of Data Objects that are cached using the Server Data Object Caching Cache feature. It also lists cache related properties configured for data objects. This report is an alternative to the time-consuming manual method of using the VLS Console to research Server Data Objects. For more details on the Server Data Object Caching feature, please refer to the "Server Data Object Caching" section in the *CGI Advantage System Administration Guide*.

The generated report categorizes Data Objects into three groups:

- Data Objects with all rows stored in cache
- Data Objects with a finite number of rows stored in cache
- Data Objects with no rows stored in cache

For Data Objects maintained in cache, the report also lists values for cache properties, specifically *CacheSize* and *ExpirationMillis***.**

When to Run

As needed.

Major Input

Metadata from VLS console.

Output

SDO Cache Report

Job Return Code

Return Code	**Condition**
Successful (1)	The report was generated successfully.

Return Code	Condition
Warning (4)	This job does not issue this return code.
Non Fatal Error (8)	This return code is issued under the following condition: • No eligible records found in the SDO Cache.
Failed (12)	The job will fail under the following condition: • Run time exceptions for unexpected situations.
Terminated (16)	This return code is issued when the job is terminated by the user.
System Failure (20)	This return code is issued when the job is terminated because of database server or network issues.

Sort Criteria

For each group of Data Objects, sort them by the Data Object Name.

Selection Criteria

- Select records from SDO Cache where Object Type = Data Object
- Categorize the selected records based on criteria and generate the report based on these categories:

Category	Criteria
Data Objects with all rows stored in the SDO cache.	CacheSize = -1
Data Objects with a finite number of rows stored in the SDO Cache.	CacheSize > 0
Data Objects with no rows stored in the SDO Cache.	CacheSize = 0

Troubleshooting

If the SDO Cache Report job runs unsuccessfully, contact the System Administrator to correct the problem and submit a new job.

2.2.23 System Maintenance Utility

Description

System Maintenance Utility is the primary utility that can be used to maintain table data and transactions in the system. System Maintenance Utility provides the following features:

- Maintenance of table data
  - Exporting table records
  - Inserting table records
  - Update table records
  - Overlay table records
  - Delete table records
  - Purge table records
  - CSV Overlay table records
- Maintenance of transactions
  - Archiving transactions
  - Archiving historical transactions
  - Unarchiving transactions
  - Exporting transactions
  - Importing transactions
  - Uploading transactions to the Transaction Catalog
  - Performing transaction custom actions
  - Marking transactions for processing
  - Holding transactions from processing
  - Discarding transactions
  - Editing transactions
  - Deactivating transactions
  - Activating transactions
  - Validating transactions
  - Submitting transactions
  - Printing transactions
  - Rejecting All Pending transactions
  - Approve transactions
- Statistics for the actions that it performs
- Restart capability for failed jobs during processing
- Chunking database commits for certain action such as import of transactions
- Provision of specifying maximum errors allowed

- Capability of supporting additional reference types such as:
  - Instream Reference where the actual attachment is embedded in the Foundation XML as an encoded data field.
  - ECM Reference where the Foundation XML passes the identifiers needed to allow Advantage to access the attachment via the ABI ECM adapter.

An input parameter file can be used to specify parameters that are required to run a SysManUtil process. The location of the input parameter file is defined in the params file since it must reside in the export file location. However, the name of the parameter file is itself a parameter to SysManUtil. This input parameter file is used for specifying more than one action in single run.

SysManUtil can also be setup as a standalone job in most cases. This is useful to run an online batch job to perform actions such as submitting transactions, deleting table data etc. Only a limited number of parameters need to be setup as part of this process. The standalone job is not an option for printing transactions.

To perform different kind of jobs, a custom parameter page has been set up. This custom parameter page is used to manage parameters for all kinds of jobs run by SysManUtil.

A batch job can be run using SysManUtil in any of the ways discussed above. The Job return status is set to “Successful” if the batch job runs successfully and is set to “Failed” if an exception is encountered.

Please refer to the *“Working with System Maintenance Utility*” s*ection in the CGI Advantage – System Processing User Guide* available on the User Guide Downloads page in the online Help for more information on the features, maintenance of tables, maintenance of transactions, and setting up parameters.

When to Run

The System Maintenance Utility can be run on demand.

Major Input

- Input Parameter File
- Single job parameters
- Custom parameters (IN_SYS_MAN_PARM)

Other Input

Input directories and files (for actions such as unarchive, import, overlay)

Output

- Log file for the job
- Output directories and files (for actions such as archive, export)

Parameters

Common parameters for running SysManUtil (both Input parameter file and Custom parameters)

Job	Parameter	Description	Default Value
SysManUtil	ACTN_CD	Action Code for the job. Based on the action selected, the parameters discussed below are shown to the user only if the custom parameters page is used.	No Default
	DOC_S_ACTN_CD	Secured transaction sub action to be processed for the job. This is enabled only if **Transaction Other Actions** is selected for action code.	No Default
	PARM_FILE	Input Parameter File Name for running SysManUtil. If a parameter file is specified, then all other parameters are taken from the parameter file, and all other job parameters are ignored. If Printing Transactions via SMU this value is required.	No Default
	GENERATE_STATS	When this parameter is specified, SysManUtil writes statistics to the log at the end of the run.	True
	USER_ID	When this parameter is specified, User ID will be considered the Execution User ID for that single job execution. The parameter is optional. The USER_ID, when supplied through a parameter file, can be specified as ‘Global User ID’ or ‘Action Specific User ID’. If defined as ‘Global User ID’, the USER_ID parameter must be the first parameter specified within the parameter file, and must be physically located just prior to the first **ACTN_CD parameter within the file. If defined as ‘Action Specific User ID’, the USER_ID parameter must be physically located after the **ACTN_CD parameter and prior to the _PARAM_LINE_ parameter	No Default

Job	Parameter	Description	Default Value
		within the file. If USER_ID parameter is not specified, then the submitter's User ID will be considered the Execution User ID for that job execution.	
	Listener Class Name (LISTENER_NAME)	Optional. This is the name of the Java class implementing the AMSSysManUtilListener interface or any interface extending the AMSSysManUtilListener. Syntax for providing the listener class name: <Package Name>.<Class Name>.	No Default

Custom parameters for processing Transaction actions using SysManUtil.

Job	Parameter	Description	Default Value
SysManUtil	DOC_TYP	The **Transaction Type** for the transactions on which action is performed. The parameter is optional. The "*" notation can be used to represent wild cards.	No Default
	DOC_DEPT_CD	The **Transaction Department Code** for the transactions on which action is performed. The parameter is optional. The "*" notation can be used to represent wild cards.	No Default
	DOC_VERS_NO	The **Transaction Version Number** for the transactions on which action is performed. The parameter is optional. Primitive search criteria using ">" or "<" can be used.	No Default
	DOC_PHASE_CD	The **Phase Code** of the transactions on which action is performed. This is an optional parameter.	No Default

Job	Parameter	Description	Default Value
	DOC_CREA_USID	The **ID** of the user who created the transaction. This is an optional parameter. The "*" notation can be used to represent wild cards.	No Default
	DOC_CD	The **Transaction Code** for the transactions on which action is performed. The parameter is optional. The "*" notation can be used to represent wild cards.	No Default
	DOC_ID	The **Transaction Id** for the transactions on which action is performed. The parameter is optional. The "*" notation can be used to represent wild cards.	No Default
	DOC_UNIT_CD	The **Unit Code** of the transactions on which action is performed. This is an optional parameter. The "*" notation can be used to represent wild cards.	No Default
	DOC_STA_CD	Indicates the status of a transaction on which action is performed. If this parameter is not specified, the default status of the transaction selected is *Held*. The following transaction status values can be specified *Ready* (Value: 2) *Held* (Value: 1) *Rejected* (Value: 3)	No Default

Job	Parameter	Description	Default Value
	CLEANUP_PERIOD	The **Fiscal Year/ Lag Days** works as selection criteria to perform the transaction actions of discard along with or without the existing selection criteria For the **Financial application**, transaction selection would be based on the **Fiscal Year**. The system would select the records where the **Fiscal Year** provided in the parameter is equivalent to the **Fiscal Year** (Current Fiscal Year) from the Transaction Header (doc_hdr) table. For the **HRM, VSS** and **Admin applications**, transaction selection would be based on the **Lag Days**. The system would select the records where the difference between the Current Date [APPCTRL date] and the Transaction last Modified date [doc_appl_last_dt] is greater than the number of **Lag Days** specified by the user.	
	USE_HIERCL_XML	The Use Hierarchical XML flag instructs the SMU Job to produce or consume hierarchical XML. If this flag is checked, the XML components are arranged in a hierarchical order.	This parameter is set to *false* (not checked) by default.

Custom parameters for processing Table actions such as Export, CSV overlay, and Purge using SysManUtil

Job	Parameter	Description	Default Value
SysManUtil	TBL_NM	Name of the table that is used for the export, CSV Overlay or purge of a table.	No Default

Job	Parameter	Description	Default Value
	FM_KEY	This is an optional parameter that you can specify to restrict the data being exported to the file for table export. When you specify a value for this parameter, data is exported beginning from this record. The format for the key is a 'where' clause that identifies a record using the primary key. For example, if table x has two primary keys A and B, the format for the key is: A=1;B=2 Please note that character values need not be enclosed in quotes (for example, VEND_CUST_CD=DELL) and the following formats must be used to specify data values: • Just the date – MM*/DD/YYYY* (for example, *06/28/2001*) • Just the time – *HH24:MI:SS* (for example, *17:06:03*) • Date and time – YYYY-MM-DD HH24:MI:SS (for example *2002-06-18 14:08:55*)	No Default

Job	Parameter	Description	Default Value
	TO_KEY	This is an optional parameter that you can specify to restrict the data being exported to the file for table export. When you specify a value for this parameter, data is exported up to this record. The format for the key is a 'where' clause that identifies a record using the primary key. For example, if table x has two primary keys A and B, the format for the key is: A=1;B=2 Please note that character values need not be enclosed in quotes (for example, VEND_CUST_CD=DELL) and the following formats must be used to specify data values: • Just the date – MM*/DD/YYYY* (for example, *06/28/2001*) • Just the time – *HH24:MI:SS* (for example, *17:06:03*) • Date and time – YYYY-MM-DD HH24:MI:SS (for example *2002-06-18 14:08:55*)	No Default
	EXCL_ATT	This is an optional parameter that you can specify to restrict the specific attributes being exported to the file for 'Transaction Export' and 'Table Export'. When you specify the value for this field the mentioned attributes get excluded from Export action. The value of this parameter needs to be given in a specific format. The attributes which need to be excluded from the export action should be given in a semi colon delimited format. For example if we want to exclude attributes DOC_LAST_DT, TBL_LAST_DT, DOC_LAST_USID,	No Default

Job	Parameter	Description	Default Value
		AMS_ROW_VERS_NO then it should be given as DOC_LAST_DT; TBL_LAST_DT; DOC_LAST_USID; AMS_ROW_VERS_NO	
	EXCL_NON_PERSIST_ATT	This parameter restricts exporting the non persistent fields from the Data Object for 'Transaction Export' and 'Table Export'. When this parameter is set to true, SysManUtil will export the Data Object without non persistent fields and vice versa. This parameter is specifically for Export.	This parameter is set to false by default.
	EXP_FILE_MODE	This parameter is optional on the table export action. The value REPLACE indicates that the file to be written will replace any existing file. The value APPEND will also replace any existing file initially, but allows records selected by more than one _PARAM_LINE_ parameter group to write to the same file. In other words, when REPLACE is used: the second, third, fourth, etc. _PARAM_LINE_ for a specific output file will cause the file to be overwritten each time. When APPEND is used: the second, third, fourth, etc. _PARAM_LINE_ for a specific output file will append to the file created by the first _PARAM_LINE_. If the parameter is not provided, the default setting is REPLACE. If provided, the EXP_FILE_MODE and FILE_NM parameters must be physically located after the **ACTN_CD parameter and prior to the first _PARAM_LINE_ parameter.	REPLACE

Job	Parameter	Description	Default Value
	EXP_FILE_TYP	This parameter is optional on the table export action. This parameter defines the type of file used for the table export. The valid selections are 1 (XML) and 2 (CSV). If not provided, the default setting is 1 (XML). This parameter is ignored on any action other than table export.	No Default
	EXP_ATT_FL	When this parameter is set to true, SysManUtil will export the object attachments along with each data object exported. This parameter is set to false by default.	This parameter is set to false by default.
	DTL_IMP_MSG_FL	Indicates whether error message details for each failed, to be processed record should be recorded in the error file.	This parameter is set to false by default.
	EDITS_FL	Indicates whether the records from the XML file are inserted with or without business rules applied. By default, this flag is set to true to indicate that records are inserted with business rules enabled.	This parameter is set to true by default.
	EXP_ATT_TYP	Indicates if the user wants to export the attachment in Zip file, in any directory or in the XML Stream. (The XML Stream produces the InStream reference attachment which stores the attachment binary encoded data information.)	No Default
	EXP_ECM_CNT	Allows the user to choose the value for the export attachment content. If *Attachment* is chosen, then the original attachment (physical file) will be exported. If *ECM Reference* is chosen, then the attachment contains an ECM reference to the file stored on the ECM software. The default value is	This parameter is set to *Attachment* by default.

Job	Parameter	Description	Default Value
		Attachment. Please note that the Export With Attachments (EXP_ATT_FL) flag must be checked in order for attachments to be exported.	
	USE_HIERCL_XML	The Use Hierarchical XML flag instructs the SMU Job to produce or consume hierarchical XML. If this flag is checked, the XML components are arranged in a hierarchical order.	This parameter is set to *false* (not checked) by default.

Custom parameters for processing Archive, Unarchive, Import, Export, and Overlay actions using SysManUtil

Job	Parameter	Description	Default Value
SysManUtil	MAX_ERRORS	Indicates the maximum number of logical errors that can occur while inserting records before the process is terminated. If a value of *–1* is specified, the processing does not stop irrespective of the number of errors encountered.	No Default
	FILE_NM	Name of the XML file to use for archive/unarchive/import/export/overlay actions and CSV file is used for CSV Overlay action. The file is located in the directory specified by the configuration parameter *XMLImportFileLocation* defined in ADV30Params.ini.	No Default

Job	Parameter	Description	Default Value
	APPLY_OVERRIDES	Indicates that when loading the transaction while importing, the override level in the transaction needs to be set to the override level of the user importing the transactions into the system (in this case, "sa" or "system administrator"). The override level of the user is set to facilitate the transaction being submitted without someone having to open the transaction manually and apply overrides from the online system. However, if this transaction is subsequently edited from the online system, the override level is removed.	False
	ERROR_FILE_NM	Specifies the name of the file to which the records that failed to be inserted while importing are written. The file containing the errors is located in the directory specified by the configuration parameter *XMLImportErrorFileLocation* defined in ADV30Params.ini.	No Default
	BYPASS_APPROVAL	Indicates that when loading the transaction, the **Bypass Approval Indicator** in the transaction has to be set. The Bypass Approval Indicator allows the transaction to be submitted without going through approvals processing. If this transaction is subsequently edited from the online system, the Bypass Approval Indicator is removed.	False
	EDITS_FL	Indicates whether the records from the XML/CSV file are updated with or without business rules applied.	True

Job	Parameter	Description	Default Value
	OVERRIDE_LVL	Indicates that when loading the transaction while importing, the override level in the transaction needs to be set to the Override Level defined by the parameter value, The Override Level parameter will only apply to the DOCIMPORT action and will only apply if the existing Apply Overrides parameter is set to "True". Valid values are: 0(blank online), 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10. If the Apply Override parameter is set to true and the Override Level value is zero or the parameter is not specified in the parameter file, then the Override Level of the user importing the transactions will be used.	No Default
	EXCL_ATT	This parameter allows restricting export of the given attributes from Transaction. When the value for this field is specified, SysManUtil excludes those specified attributes from Export action. The value of this parameter needs to be given in a specific format. The attributes which need to be excluded from the export action should be given in a semi colon delimited format. For example to exclude attributes DOC_LAST_DT, DOC_LAST_USID the format should be given as DOC_LAST_DT;DOC_LAST_USID This parameter has no default value. This parameter is specifically for Export.	No Default

Job	Parameter	Description	Default Value
	EXCL_NON_PERSIST_ATT	This parameter allows restricting export of the non persistent fields from the Transaction. When this parameter is set to true, SysManUtil will export the Transaction without non persistent fields and vice versa. This parameter is specifically for Export.	This parameter is set to false by default.
	PARM_FIL_VERBOSE	Optional field. Controls whether a parameter file's information is displayed in the job log. If set to 'N' a message is not generated. If set to 'Y' or any other value, parameter line information in the parameter file will be displayed in the job log. This parameter is exceptional such that it can be specified outside of the parameter file and still be used.	This parameter is set to Y by default.
	EXP_ATT_FL	When this parameter is set to true, SysManUtil will export the object attachments along with each data object exported. This parameter is set to false by default.	This parameter is set to false by default.
	DTL_IMP_MSG_FL	Indicates whether error message details for each failed, to be processed record should be recorded in the error file.	This parameter is set to false by default.
	EXP_ATT_TYP	Indicates if the user wants to export the attachment in Zip File, in any directory or in the XML Stream. (The XML Stream produces the InStream reference attachment which stores the attachment binary encoded data information.)	No Default

Job	Parameter	Description	Default Value
	EXP_ECM_CNT	Allows the user to choose the value for the export attachment content. If *Attachment* is chosen, then the original attachment (physical file) will be exported. If *ECM Reference* is chosen, then the attachment contains an ECM reference to the file stored on the ECM software. The default value is *Attachment*. Please note that the Export With Attachments flag (EXP_ATT_FL) must be checked in order for attachments to be exported.	This parameter is set to *Attachment* by default.
	USE_HIERCL_XML	The Use Hierarchical XML flag instructs the SMU Job to produce or consume hierarchical XML. If this flag is checked, the XML components are arranged in a hierarchical order.	This parameter is set to *false* (not checked) by default.

Custom parameters for processing Transaction Validate and Submit actions using SysManUtil

Job	Parameter	Description	Default Value
SysManUtil	EXCEP_REP_IND	Specifies the type of exception report to generate. Four types of exception reports can be generated: • DETAILED • FAILED_DOCS • PROCESSED_DOCS • FAILED_LINES	No Default
	EXCEP_REP_FILE_NM	The file name of the exception report. This parameter is required when the **Exception Report Indicator** (above) is specified. The file will be written to the directory specified by the configuration parameter *XMLExportFileLocation* defined in ADV30Params.ini.	No Default

Job	Parameter	Description	Default Value
	DOCSUB_PROG_CTR_SZ	Optional field and relevant for Action Code of Submit Transaction. Regardless of the setting of the Generate Statistics parameter, generates progression statistics of processed transactions after processing specified number of transactions. If not specified, or if value is not a positive integer, then SMU will utilize the default setting of 250. This parameter is exceptional such that it can be specified outside of the parameter file and still be used.	250
	EXP_SEV_FL	Option field. This is an Exception Severity Flag. When this flag is set to false, all errors (Info, Warning, Error, and Severe) will be displayed on the HTML and PDF exception report. When the flag set to true, only hard errors (Error and Severe) will be displayed on the HTML and PDF exception report.	No Default

Additional custom parameters for processing Transaction Approve

Job	Parameter	Description	Default Value
SysManUtil	APRV_ID	Specifies the Approver ID to be used to select and Approve the worklist records.	No Default
	APRV_ROL	Specifies the Approval Roles to select the records from, using the Approver ID.	No Default
	APRV_LVL	Specifies the Approval level for which the records should be selected and approved.	No Default

Custom parameters for processing all actions except Archive, Unarchive using SysManUtil

Job	Parameter	Description	Default Value
SysManUtil	RESTART_FL	Stores information about the job so that it can be restarted if required.	True
	COMMIT_BLOCK	Indicates the number of records that can be inserted before the changes have to be committed to the database.	No Default
	FY / LAG_DAYS	Relevant when Action Code is Discard Transaction and Transaction Phase Code is Draft/Conflict Draft or Action Code is Reject All Pending Transaction. For Financial application this parameter is Fiscal Year. When Fiscal Year is specified then System selects transactions with that Fiscal Year. **If parameter file is being used to specify this parameter in Advantage Financial application then this parameter should be listed as FY.** For other applications this parameter is Lag Days. When Lag Days is specified then System selects transactions where difference between the Application Control date and the Transaction last Modified date is greater than Lag Days parameter value (while calculating Application Control date is excluded and Transaction last Modified date is included). **If parameter file is being used to specify this parameter for applications other than Advantage Financial then this parameter should be listed as LAG_DAYS.**	No Default

Sort Criteria

None

Selection Criteria

- Active Status
- Job Expiry Date

Troubleshooting

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

SMU Transaction Upload Job

Job Name	SMU Transaction Upload Job
Recommended Frequency	On Demand or as part of the chain job
Single Instance Required	No
Can be Restarted?	Optionally based on the Save Restart Information parameter
Reports Generated	No

Overview

System Maintenance Utility is the primary utility that is used to load transactions into the Transaction Catalog. This job can be run by providing the necessary information either through an input parameter file or by setting parameters on the SMU Parameter page. The job first validates the parameters and then loads the transactions from the XML file into the Transaction Catalog based on the provided parameters.

Restart Information

The job can be restarted only when the Save Restart Information Flag parameter is set to *True*. If the parameter is set to *False*, then the job cannot be restarted. See the “Problem Resolution” section for more details when a job ends with a Return Code other than Successful.

Major Input

- SMU job parameter file
- XML File

Batch Parameters

The job uses the following parameters when an input parameter file is supplied.

Parameter	Description	Default Value
Parameter File (PARM_FILE)	Optional field. Input Parameter File Name for running SysManUtil. If a parameter file is specified, then other parameters are taken from the parameter file, and most other job parameters are ignored.	
Verbose Messages for Parameter File (PARM_FIL_VERBOSE)	Optional field. Controls whether a parameter file's information is displayed in the job log. If set to "N" a message is not generated. If set to "Y" or any other value, parameter line information in the parameter file will be displayed in the job log. This parameter is exceptional such that it can be specified outside of the parameter file and still be used.	Y

The following parameters can be entered on the Custom Parameter page when the Input parameter file is supplied to this job.

Parameter	Description	Default Value
Action code (ACTN_CD)	Required field. The Action code that instructs the program to perform the Transaction Import action.	171
File Name (FILE_NM)	Required field. The File Name that defines what XML file will be uploaded.	No Default
Transaction Status Code (DOC_STA_CD)	Required field that will be used by the program when loading transactions. Valid values are 1 - Held and 2 - Ready. Transactions loaded as Held will not be selected by another process searching for transactions to submit. Ready transactions will be selected by such a program. Held is often used when some user action will be required.	2
Apply Overrides? (APPLY_OVERRIDES)	Optional field. If set to True the override level of all imported transactions is set to the override level of the user submitting the job. The override level of the user is set to facilitate the transaction being submitted without someone having to open the transaction manually and apply overrides from the online system. Note that if a	False

Parameter	Description	Default Value
	transaction is subsequently edited by a user online, the override level is removed.	
OVERRIDE_LVL	Indicates that when loading the transaction while importing, the override level in the transaction needs to be set to the Override Level defined by the parameter value, The Override Level parameter will only apply to the DOCIMPORT action and will only apply if the existing Apply Overrides parameter is set to “True”. Valid values are: 0(blank online), 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10. If the Apply Override parameter is set to true and the Override Level value is zero or the parameter is not specified in the parameter file, then the Override Level of the user importing the transactions will be used.	No Default
Bypass Approvals? (BYPASS_APPROVAL)	Optional field. If set to True the Bypass Approval Indicator of all imported transactions is set to True. The Bypass Approval Indicator allows the transaction to be submitted without going through approval processing (workflow). Note that if a transaction is subsequently edited by a user online, the Bypass Approval Indicator is removed.	False
Generate Statistics? (GENERATE_STATS)	Optional field. If set to True, SysManUtil writes statistics to the job log at the end of the run.	True
Maximum Errors Allowed (MAX_ERRORS)	Optional field. Indicates the maximum number of logical errors that can occur while inserting records before the process is terminated. If a value of *1* is specified, the processing does not stop irrespective of the number of errors encountered.	No Default
Detailed Import Message Reporting?	Optional field. Indicates whether Detailed Import Message Reporting is required or not.	No Default
Error File Name (ERROR_FILE_NM)	Optional field. Specifies the name of the file to which the records that failed to be inserted while importing are written.	No Default

Parameter	Description	Default Value
Save Restart info? (RESTART_FL)	Optional field. Stores information about the job so that it can be restarted if required.	True
Commit Block Size (COMMIT_BLOCK)	Required field. Controls how many records are committed by the application at one time. The size should be compatible with technical capabilities and performance guidelines.	1
Verbose Messages for Parameter File (PARM_FIL_VERBOSE)	Optional field. Controls whether a parameter file's information is displayed in the job log. If set to "N" a message is not generated. If set to "Y" or any other value, parameter line information in the parameter file will be displayed in the job log.	Y
USER_ID	When this parameter is specified, User ID will be considered the Execution User ID for that single job execution. The parameter is optional. The USER_ID, when supplied through a parameter file, can be specified as 'Global User ID' or 'Action Specific User ID'. If defined as 'Global User ID', the USER_ID parameter must be the first parameter specified within the parameter file, and must be physically located just prior to the first **ACTN_CD parameter within the file. If defined as 'Action Specific User ID', the USER_ID parameter must be physically located after the **ACTN_CD parameter and prior to the _PARAM_LINE_ parameter within the file. If USER_ID parameter is not specified, then the submitter's User ID will be considered the Execution User ID for that job execution.	No Default

Major Output

- Transactions in Draft Phase on Transaction Catalog.

Batch Return Codes

The following table shows the potential job return codes for the SMU Transaction Upload job. Please note that these return codes are applicable for each input file (data file). If the parameter file is supplied as an input file, then these return codes are applicable for each parm line on the parameter file and the job will end with the highest return code across all of the parm lines.

Return Code	Condition
Successful (1)	All of the records are loaded into the Transaction Catalog successfully.
Warning (4)	This return code will be issued under the following conditions: • The input file is empty • Some records fail to load into the Transaction Catalog.
Non Fatal Error (8)	All records fail to load into the Transaction Catalog.
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • The input file is not found in the specified directory • If the error messages exceed the maximum number specified in the parameter • Runtime exceptions encountered for any unexpected situations When the job ends with a return code of Failed, subsequent jobs in the chain are set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain 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 Sequence

N/A

Selection Criteria

N/A

Problem Resolution

If the input file for this job is supplied and the job ends with the return code other than successful, then there is no need to back out any updates. Review the reason why the job ended with that return code, resolve it and schedule a new job using the same Parameter File (PARM_FILE) to reattempt loading the records to the Transaction Catalog.

Considerations when SMU is part of a chain job

When the XML file designated by the File Name (FILE_NM) parameter is created in a prior job in the chain and this job ends with the return code other than successful, updates made by the prior job in the chain may need to be backed out if the issues cannot be resolved immediately and the records successfully loaded to the Transaction Catalog. For more details on this, refer to the respective chain job's run sheet.

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All parameters are valid and all the transactions are loaded successfully.	N/A	N/A
Warning (4)	The input file is empty. **Sample message:** No records found on the input file.	This scenario will arise when the input file (File Name (FILE_NM)) contains no transaction records to be loaded to the transaction catalog. Schedule a new job with an input file that has records to be uploaded in to the Transaction Catalog.	
	This return code will be issued when the job fails to load some of the transactions. **Sample Message:** Unable to load all of the transactions into the catalog.	Analyze the reason why records failed to load to the Transaction Catalog. When the current run is a re-execution of a previous run (the same input file is used in a new job) this error may be seen because the records that previously loaded successfully will not load again (duplicate entry on the Transaction Catalog is not allowed). If the records failed to load to the Transaction Catalog due to any other reason, then analyze the reason, resolve it and schedule a new schedule a new job using the same input parameter file name.	
Non Fatal Error (8)	This return code will be issued when the job failed to load all of the	Analyze the reason that records failed to load to the Transaction Catalog and	

Possible Return Codes	Condition	Recommendation	Other Instructions
	transactions.	schedule a new job using the same Parameter File (PARM_FILE).	
Failed (12)	This return code will be issued when the parameters are not valid. **Sample Message:** Parameter file could not be located/read	Make sure that the parameter file exists in the specified folder and schedule a new job.	
	This return code will be issued when the input file is not found. **Sample Message:** Input file xxxx (xxxx being the file name) not found on the yyyy (yyy being the path).	Make sure that the input file exists in the specified folder and schedule a new job.	
	This return code will be issued when the number of error messages exceeds the maximum number specified on the batch parameter. **Sample Message:** No of messages exceeds the maximum number (xxx).	Resolve the error and schedule a new job using the same Parameter File (PARM_FILE).	Alternatively the Maximum errors allowed value can be changed to a higher number.
	Failed because of runtime exceptions for an unexpected situation.	Resolve the error and schedule a new job using the same Parameter File (PARM_FILE) provided no new instance of this chain has been executed.	
Terminated (16)	Job is terminated manually by the user.	Reason for the termination needs to be investigated before scheduling a new job.	

Possible Return Codes	Condition	Recommendation	Other Instructions
System Failure (20)	Job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated before scheduling a new job.	

SMU Transaction Submit Job

Job Name	SMU Transaction Submit Job
Recommended Frequency	On Demand or as part of the chain job.
Single Instance Required	No
Can be Restarted?	Optionally based on the Save Restart Information parameter.
Reports Generated	Optionally based on the Exception Report parameter in a text format.

Overview

System Maintenance Utility is the primary utility used in CGI Advantage to submit transactions. This job can be run by providing the necessary information either through an input parameter file or by setting parameters on the SMU Parameter page. The job first validates the provided parameters and then submits transactions based on the job parameters.

Restart Information

The job can be restarted only when the Save Restart Information Flag parameter is set to ‘True’. If the parameter is set to False, then the job cannot be restarted. See the “Problem Resolution” section for more details when a job ends with a Return Code other than Successful.

Major Input

- SMU job parameter file
- Draft Transactions in the Transaction Catalog

Batch Parameters

Parameter	Description	Default Value
Parameter File	Required field. Parameter file that will be used to define the parameters for	

Parameter	Description	Default Value
(PARM_FILE)	submitting the transactions.	
Exception Report File (EXCEP_REP_FILE_NM)	Optional field. Defines what file the program is to use to create an exception report.	
Verbose Messages for Parameter File (PARM_FIL_VERBOSE)	Optional field. Controls whether a parameter file's information is displayed in the job log. If set to "N" a message is not generated. If set to "Y" or any other value, parameter line information in the parameter file will be displayed in the job log.	Y

The following parameters are available on the Custom Parameter page when the Input parameter file is not supplied to this job.

Parameter	Description	Default Value
Action code (ACTN_CD)	Required. Action code which instructs the program to perform a certain action on its records.	162
Transaction Code (DOC_CD)	Optional Field. When entered, the values entered are used in the selection logic for which transactions will be submitted.	
Transaction Creation User Id. (DOC_CREA_USID)	Optional Field.	
Transaction department code for the transactions that have to be submitted (DOC_DEPT_CD)	Optional Field. When entered, the values entered are used in the selection logic for which transactions will be submitted.	
Transaction Id (DOC_ID)	Optional Field. When entered, the values entered are used in the selection logic for which transactions will be submitted.	
Transaction Status (DOC_STA)	Optional Field. When entered, the values entered are used in the selection logic for which transactions will be submitted.	
Transaction version number (DOC_VERS_NO)	Optional Field. When entered, the values entered are used in the selection logic for which transactions will be	

Parameter	Description	Default Value
	submitted.	
Exception Report File Name (EXCEP_REP_FILE_NM)	Required field. Defines what file the program is to use to create an exception report.	
Exception Report Indicator (EXCEP_REP_IND)	Required field. Defines the level of detail of the exception report. Valid values are: 1 = Detailed 2 = Failed Transactions 3 = Processed Transactions 4 = Failed Transaction Lines 5 = Transaction Status	
Save Restart Information? (RESTART_FL)	Stores information for the job such that the job can be restarted if required. In the case of a transaction submit action, if the job is being restarted only those transactions that satisfy the search criteria greater than the checkpoint stored will be selected and submitted. Please note that new transactions that enter the system between the time the job failed and was restarted will also be submitted if they meet the selection criteria specified. This parameter is set to true by default.	
Progression Counter Size (DOCSUB_PROG_CTR_SZ)	The progression counter size controls the interval at which progression messages are generated in the job log during transaction submit processing. These intermediate messages are displayed periodically based on the parameter's value until the action completes. The parameter's value should be positive if entered. If not specified or incorrectly specified, then processing uses a default value of 250.	250

Major Output

- Transactions in Pending / Final / Rejected on Transaction Catalog.

Batch Return Codes

The following table shows the potential job return codes for the SMU Transaction Submit job.

Return Code	Condition
Successful (1)	All of the transactions submitted successfully.
Warning (4)	N/A
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Parameter file is not found • Runtime exceptions encountered for any unexpected situations When the job ends with a return code of failed, subsequent jobs in the chain are set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain 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 Sequence

N/A

Selection Criteria

N/A

Problem Resolution

If the job encounters a runtime exception (failure, termination, etc.) no special recovery steps are necessary since no source table updates occur at this step – the job is simply performing a submit action on existing transactions. When an input file is supplied as a parameter to this job and the job ends with any return code other than successful, a new job can be scheduled using the same Parameter File (PARM_FILE) after resolving the issues.

Considerations when SMU is part of a chain job

When all transactions are not submitted due to a failure of the SMU submit step, and the parameter file name created by the chain job is not unique, and another instance of the chain job has been run, then the un-submitted transactions should be submitted manually. This is because the parameter file created/used in the first run would have been overwritten by the second chain job and will include transaction records created by the second job.

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the transactions submitted successfully.	N/A	
Warning (4)	N/A	The job doesn't issue this return code.	
Non Fatal Error (8)	N/A	The job doesn't issue this return code.	
Failed (12)	This return code will be issued when the input parameter is not found in the specified directory. **Sample Message:** Parameter file could not be located/read.	Make sure that the parameter file exits in the specified folder and schedule a new job.	
	Failed because of runtime exceptions for an unexpected situation.	Resolve the error and schedule a new job using the same Parameter File (PARM_FILE) provided no new instance of this chain has been executed.	
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)	Job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated before scheduling a new job.	

SMU Transaction Print Job

Job Name	SMU Transaction Print Job
Recommended Frequency	On Demand or as part of the chain job.
Single Instance Required	No
Can be Restarted?	Yes
Reports Generated	Optionally based on the Exception Report parameter in a text format.

Overview

System Maintenance Utility is the primary utility used in CGI Advantage to print transactions. This job can be run by providing the necessary information through an input parameter file. The job first validates the provided parameters and then prints transactions based on the job parameters.

Restart Information

If the job fails due to any reason other than an invalid or incorrect parameter it can be restarted after the error has been resolved. The restarted job will continue printing transactions in the parameter file that were not yet printed in the previous run. If there is any error printing any transaction, the Print Job will not continue and hence the user must restart the job after the problem has been resolved to complete the Print Job.

If the job fails due to a problem with the parameter file, correct the parameter value and schedule a new instance of the job. Any updates made to the parameter file will not be recognized if the job is restarted.

Major Input

- SMU job parameter file
- Transactions in the Transaction Catalog

Batch Parameters

Parameter	Description	Default Value
Parameter File (PARM_FILE)	Required field. Parameter file that will be used to define the parameters for printing the transactions and the transactions to be printed.	
Exception Report File (EXCEP_REP_FILE_NM)	Optional field. Defines what file the program is to use to create an exception	

Parameter	Description	Default Value
	report.	
Verbose Messages for Parameter File (PARM_FIL_VERBOSE)	Optional field. Controls whether a parameter file's information is displayed in the job log. If set to "N" a message is not generated. If set to "Y" or any other value, parameter line information in the parameter file will be displayed in the job log.	Y

The Custom Parameter page is never applicable to the Printing Transactions via SMU job.

Major Output

- Printed Transactions
- Updated Transactions (with Last Print Date populated)

Batch Return Codes

The following table shows the potential job return codes for the SMU Transaction Print job.

Return Code	Condition
Successful (1)	All of the transactions sent to printer successfully.
Warning (4)	N/A
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Parameter file is not found • Runtime exceptions encountered for any unexpected situations When the job ends with a return code of failed, subsequent jobs in the chain are set to inactive.
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of Terminated, subsequent jobs in the chain 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 Sequence

N/A

Selection Criteria

N/A

Problem Resolution

If the job encounters a runtime exception (failure, termination, etc.) no special recovery steps are necessary since no source table updates occur at this step – the job is simply performing a print action on existing transactions. When an input file is supplied as a parameter to this job and the job ends with any return code other than successful, a new job can be scheduled using the same Parameter File (PARM_FILE) after resolving the issues.

Considerations when SMU is part of a chain job

When all transactions are not printed due to a failure of the SMU print step, and the parameter file name created by the chain job is not unique, and another instance of the chain job has been run, then the un-printed transactions can be printed manually or the job can simply be rerun. A rerun of the job will result in transactions that were printed in the first run being reprinted in the second run.

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

Possible Return Codes	**Condition**	**Recommendation**	**Other Instructions**
Successful (1)	All of the transactions printed successfully.	N/A	
Warning (4)	N/A	The job does not issue this return code.	
Non Fatal Error (8)	N/A	The job does not issue this return code.	
Failed (12)	This return code will be issued when the input parameter is not found in the specified directory. **Sample Message:** Parameter file could not be located/read.	Make sure that the parameter file exits in the specified folder and schedule a new job.	

Possible Return Codes	Condition	Recommendation	Other Instructions
	Failed because of runtime exceptions for an unexpected situation.	Resolve the error and schedule a new job using the same Parameter File (PARM_FILE).	
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)	Job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated before scheduling a new job.	

SMU Transaction Approve Job

Job Name	SMU Transaction Approve Job
Recommended Frequency	On Demand or as part of the chain job.
Single Instance Required	No
Can be Restarted?	Optionally based on the Save Restart Information parameter.
Reports Generated	Optionally based on the Exception Report parameter in a text format.

Overview

System Maintenance Utility is the primary utility used in CGI Advantage to approve transactions. This job can be run by providing the necessary information either through an input parameter file or by setting parameters on the SMU Parameter page. The job first validates the key parameters and then approves transactions selected based on the job parameters.

Restart Information

The job can be restarted only when the **Save Restart Information Flag** parameter is set to *True*. If the parameter is set to *False*, then the job cannot be restarted. See the “Problem Resolution” section for more details when a job ends with a Return Code other than Successful.

Major Input

- SMU job parameter file

- Pending Transactions in the User/Role worklist

Batch Parameters

Parameter	Description	Default Value
Parameter File (PARM_FILE)	Required field. Parameter file that will be used to define the parameters for approving the transactions.	
Exception Report File (EXCEP_REP_FILE_NM)	Optional field. Defines what file the program is to use to create an exception report.	
Verbose Messages for Parameter File (PARM_FIL_VERBOSE)	Optional field. Controls whether a parameter file's information is displayed in the job log. If set to *N* a message is not generated. If set to *Y* or any other value, parameter line information in the parameter file will be displayed in the job log.	Y

The following parameters are available on the Custom Parameter page when the Input parameter file is not supplied to this job.

Parameter	Description	Default Value
Action code (ACTN_CD)	Required. Action code which instructs the program to perform a certain action on its records.	181
Approver ID (APRV_ID)	Required. User ID which will be used to select and approve the pending records from the worklist.	
Approval Role (APRV_ROL)	Optional Field. When entered, records from the specified Approval Role worklist will be selected for approval by the job. If the parameter is left empty, the job will select the records from all the worklists assigned to the user.	
Transaction Code (DOC_CD)	Optional Field. When entered, the values are used in the selection logic for which transactions will be selected for approval.	
Transaction Creation User ID	Optional Field.	

Parameter	Description	Default Value
(DOC_CREA_USID)		
Transaction Department Code for the transactions that have to be submitted (DOC_DEPT_CD)	Optional Field. When entered, the values entered are used in the selection logic for which transactions will selected for approval.	
Transaction ID (DOC_ID)	Optional Field. When entered, the values are used in the selection logic for which transactions will be submitted.	
Transaction Status (DOC_STA)	Optional Field. When entered, the values entered are used in the selection logic for which transactions will be submitted.	
Transaction Version Number (DOC_VERS_NO)	Optional Field. When entered, the values are used in the selection logic for which transactions will be submitted.	
Approval Level (APRV_LVL)	Required. The parameter will be used to select and approve the transaction at the specified Approval Level.	
Apply Overrides (APPLY_OVERRIDES)	Optional Field. The parameter will be used to decide if overrides have to be applied on the transaction before applying the approval.	
Exception Report Indicator (EXCEP_REP_IND)	Optional field. Defines the level of detail of the exception report. Valid values are: • 1 = Detailed • 2 = Failed Transactions • 3 = Processed Transactions • 4 = Failed Transaction Lines • 5 = Transaction Status	
Exception Report File Name (EXCEP_REP_FILE_NM)	Required field if EXCEP_REP_IND is specified. Defines what file the program is to use to create an Exception Report.	
Save Restart Information? (RESTART_FL)	Stores information for the job such that the job can be restarted if required. In the case of a transaction approve action, if the job is being restarted for only those transactions that satisfy the search criteria greater than the	

Parameter	Description	Default Value
	checkpoint stored will be selected and approved. Please note that new transactions that enter the system between the time the job failed and was restarted will also be approved if they meet the selection criteria specified. This parameter is set to *True* by default.	
Progression Counter Size (DOCSUB_PROG_CTR_SZ)	The Progression Counter Size controls the interval at which progression messages are generated in the job log during transaction approval processing. These intermediate messages are displayed periodically based on the parameter's value until the action completes. The parameter's value should be positive if entered. If not specified or incorrectly specified, then processing uses a default value of 250.	250

Major Output

- Transactions in Pending / Final / Rejected state on the Transaction Catalog.

Batch Return Codes

The following table shows the potential job return codes for the SMU Transaction Submit job.

Return Code	Condition
Successful (1)	All of the transactions approved successfully.
Warning (4)	N/A
Non Fatal Error (8)	N/A
Failed (12)	The job will fail under the following conditions: • Parameters are invalid • Parameter file is not found • Runtime exceptions encountered for any unexpected situations When the job ends with a return code of *Failed*, subsequent jobs in the chain are set to *Inactive*.
Terminated (16)	This return code will be issued when the job is terminated by the user. When the job ends with a return code of *Terminated*, subsequent jobs in the chain are set to *Inactive*.

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

Sort Sequence

N/A

Selection Criteria

N/A

Problem Resolution

If the job encounters a runtime exception (failure, termination, etc.) no special recovery steps are necessary since no source table updates occur at this step – the job is simply performing an approve action on existing transactions. When an input file is supplied as a parameter to this job and the job ends with any return code other than *Successful*, a new job can be scheduled using the same Parameter File (PARM_FILE) after resolving the issues.

Considerations when SMU is part of a chain job

When all transactions are not approved due to a failure of the SMU approve step, and the parameter file name created by the chain job is not unique, and another instance of the chain job has been run, then the un-approved transactions should be approved manually. This is because the parameter file created/used in the first run would have been overwritten by the second chain job and will include transaction records created by the second job.

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

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the transactions approved successfully.	N/A	
Warning (4)	N/A	The job doesn’t issue this return code.	
Non Fatal Error (8)	N/A	The job doesn’t issue this return code.	
Failed (12)	This return code will be issued when the input	Make sure that the parameter file exits in the specified folder	

Possible Return Codes	Condition	Recommendation	Other Instructions
	parameter is not found in the specified directory. **Sample Message:** Parameter file could not be located/read.	and schedule a new job.	
	Failed because of runtime exceptions for an unexpected situation.	Resolve the error and schedule a new job using the same Parameter File (PARM_FILE) provided no new instance of this chain has been executed.	
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)	Job is terminated because of database server or network issues.	Reason for the System Failure needs to be investigated before scheduling a new job.	

2.2.24 Workflow Delegate to Alternate Approver

Chain or Job Name	Workflow Delegate To Alternate Approver
Recommended Frequency	This job can be run on demand.
Single Instance Required	No
Can be restarted?	No
Reports generated	No

Overview

The Workflow Delegate To Alternate Approver job is used to route transactions that are submitted into the manager's queue during the time the Out of Office is effective into the alternate approver's queue.

The following table shows the various steps that the Workflow Delegate To Alternate Approver job goes through and the messages issued at each step:

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters. • If Begin date is not provided, then the following message is issued: • "Begin date is empty; therefore, Start date not entered, defaulting to current date" • If End date is not provided, then the following message is issued: • "End date is empty; therefore, Start date not entered, defaulting to current date" • If the End date is less than begin date, then the following message is issued: • "End date cannot be before the start date"; • If the End date is less than the application system, then the following message is issued: • "End date cannot be before the application system date."
2. Selection of Records	Select all the out of office users if the logged in user is an alternate approval manager.
3. Process Records	Update the ALT_ASSIGNEE column value with logged in user in the WF_APRV_WRK_LST table if the Out of Office manager chose Alternate approval manager as logged in user.
4. Job Complete	Job processing complete.

When to Run

The job can be run on demand.

Restartability Information

Schedule a new job if it fails.

Major Input

End Date value specified as System Date (Optional).

Batch Parameters

Parameter	Description	Default Value	Explanation
BEGIN_DT	Begin Date	Empty	A required date to begin alternate routing.
END_DT	End Date	Empty	An optional date to stop alternative routing.

Major Output

Update the ALT_ASSIGNEE column value on the WF_APRV_WRK_LST table with the Alternate Approver for Out of Office mangers.

Job Return Code

Return Code	Condition
Successful (1)	This Return Code is issued under the following condition: • The report validating successfully. • On successfully validating all the user records and performing the update action.
Warning (4)	This Return Code is issued under the following condition: • While running the Batch Job, if no eligible records are found, the system shall complete the Batch Job and the Job shall have a return code of *Warning*.
Non-Fatal Error (8)	While running the Batch Job, if only a partial number of records are processed by the system, the system shall complete the Batch Job and the Job shall have a return code of *Non-Fatal Error*.
Failed (12)	The job may fail under the following conditions:

	• Parameters are invalid. • Runtime exceptions for unexpected situations.
Terminated (16)	This Return Code is issued when the job is terminated by the user.
System Failure (20)	This Return Code is issued when the job is terminated because of database server or network issues.

Sort Criteria

None

Problem Resolution

The following table shows the possible return codes and recommendations for each processing step. For further assistance, contact the System Administrator.

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the parameters are validated successfully.	N/A	N/A
Warning (4)	Job ended with a Warning because No eligible records are found.	N/A	N/A
Non-Fatal Error (8)	This Return Code is issued While running the Batch Job, if only a partial number of records are processed by the system.	N/A	N/A
Failed (12)	Job failed due to following conditions. • Parameters are invalid. • Runtime exceptions for unexpected situations.	If the job fails because of the runtime exceptions, investigate the exception reported by the process, resolve the error and restart the job.	N/A
Terminated (16)	Job is terminated manually by the user.	The reason for the termination needs to be investigated before rescheduling the job.	N/A

Possible Return Codes	**Condition**	**Recommendation**	**Other Instructions**
System Failure (20)	When the job is terminated because of database server or network issues.	The reason for the System Failure needs to be investigated before rescheduling the job.	N/A

2.2.25 Worklist Details Synchronization

Chain or Job Name	Worklist Details Synchronization
Recommended Frequency	Whenever there is a change in the Worklist Details Configuration for the Transaction data maintained on the Worklist Details table, this job should be run to synchronize the Worklist Details entries with the updated configurations.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Configuration for Worklist Details

Overview

The Worklist Details Synchronization batch job is used to synchronize the data on Worklist Details as per setup on the Configure Worklist Details page. The Worklist Details record is an extension of a Worklist entry to display Transaction specific data as specified on the Worklist Details Configuration page. It helps the User to sort Worklist items by enabling sort on Transaction specific data on Worklist Details records. For example: An Accounts Payable manager should be able to sort Invoice transactions as per their Scheduled Payment Date so that he/she may be able to prioritize approval of Invoice transactions with an earlier Scheduled Payment Date. Thus, the Scheduled Payment Date field can be displayed on the Worklist Details page for Worklist items of Invoice Transactions with appropriate setup on the Configure Worklist Details page. The manager can then sort on the Scheduled Payment Date field on the Worklist Details page.

Whenever any configuration changes are made on this Configure Worklist Details page then the records are marked as dirty behind the scenes. This batch job has to be run in order for the configuration changes to take effect and reset the configuration records as non-dirty (which means configurations changes were applied). Synchronization includes inserting, updating, or deleting Worklist Details record data to be in synch with the configuration setup on this page.

Note: The only Configure Worklist Details records selected are those that have the Transaction Codes for the transactions pertaining to the Advantage application from where this batch process is run. For example, if the batch process is run from the Advantage Financial application then Configure Worklist Details records for PO Transactions are picked up and ones for UDOC that belong to the Advantage Administration application are ignored.

The following example explains the types of configurations possible on the Configure Worklist Details page and how they get applied when the batch job is run:

All of the following configuration is set up for Transaction Code = PO.

On 10/1/2011 UserA wants to see field F1 values on the Worklist Details page for all Worklist items with the Transaction Code = PO. This is the first setup for PO. Note in real time, field F1 can be Expiration Date or Effective Begin Date from PO Header, whichever field UserA chooses to view for PO Transactions.

UserA sets up the following configuration record on the Configure Worklist Details page:

In the explanation that follows Transaction Component and Transaction Field are termed as New fields and Old Transaction Component and Old Transaction Field are termed as Old fields.

Transaction Component	Transaction Field	Old Transaction Component	Old Transaction Field	
PO_DOC_HDR	F1			This is termed as INSERT configuration and it is represented by this state: New fields are populated and Old field are blank. Record is marked as dirty.

When the batch job is run during the nightly cycle on 10/1/2011, the job picks up all dirty configuration records to process, that is, the above INSERT configuration. It adds Worklist Details records with data copied from the Worklist records where Transaction Code = PO and populates F1 field value on it. F1 field value is obtained from the PO_DOC_HDR table for the transaction listed on the Worklist item for which the Worklist Details record is being inserted. Then it copies over New field values into Old field values on the configuration record to leave it in following state:

Transaction Component	Transaction Field	Old Transaction Component	Old Transaction Field	
PO_DOC_HDR	F1	PO_DOC_HDR	F1	This now becomes the APPLIED (non dirty) configuration because existing Worklist Details records were synchronized by the batch job to show F1 for PO. This state is represented by Old fields and New fields having the same values. Any new Worklist entries created will also show field F1 on PO Worklist Details records because APPLIED configuration is applied for newly created Worklist Details entries

				that are created when new Worklist items are created.

On 10/2/2011 UserA added configuration records to see fields F2, F3, F4 and F5 for PO Worklist Details records. And the batch process ran that did the synchronization and left the configuration table in this state:

Transaction Component	Transaction Field	Old Transaction Component	Old Transaction Field
PO_DOC_HDR	F1	PO_DOC_HDR	F1
PO_DOC_HDR	F2	PO_DOC_HDR	F2
PO_DOC_HDR	F3	PO_DOC_HDR	F3
PO_DOC_HDR	F4	PO_DOC_HDR	F4
PO_DOC_HDR	F5	PO_DOC_HDR	F5

On 10/3/2011 UserA wants to see field F6 from the Accounting line of the PO; however, only 5 fields are allowed per Transaction Code. UserA then decides to view F6 from the PO Accounting line instead of getting to view F1 from PO Header. UserA also wants to no longer see F3 on the Worklist Details records. UserA then does the following setup for the same:

Transaction Component	Transaction Field	Old Transaction Component	Old Transaction Field	
PO_DOC_ACTG	F6	PO_DOC_HDR	F1	New fields changed. This is termed as UPDATE configuration and it is represented by this state: New fields are populated and Old field are populated and they are different. Record is marked as dirty.
PO_DOC_HDR	F2	PO_DOC_HDR	F2	No change

		PO_DOC_HDR	F3	When UserA selected 'Delete' link, New field values are blanked out. This is termed as DELETE configuration and it is represented by this state: New fields are blank and Old field are populated. Record is marked as dirty. Note the record is actually deleted by the batch job after it does synchronization.
PO_DOC_HDR	F4	PO_DOC_HDR	F4	No change
PO_DOC_HDR	F5	PO_DOC_HDR	F5	No change

When the batch job is run during the nightly cycle on 10/3/2011, the job picks up all dirty configuration records to process, that is, the UPDATE and DELETE configuration records. It picks them up for processing one by one in the order as they appear on the page.

For UPDATE configuration it selects all Worklist Details records for PO and updates the field that stored F1 to instead store F6 data.

Note it is possible that PO has multiple Accounting lines, in which case F6 value from the first Accounting line from the sorted Accounting lines by their primary keys is used. The same rule is applied when multiple Transaction component lines are found.

For DELETE configuration the job wipes out F3 field values from Worklist Details records for the PO.

The job then copies over New field values into Old field values on the configuration record to leave it in following state (note: configuration record for F3 is deleted).

Transaction Component	**Transaction Field**	**Old Transaction Component**	**Old Transaction Field**	
PO_DOC_ACTG	F6	PO_DOC_ACTG	F6	This now becomes the APPLIED (non dirty) configuration because existing Worklist Details records were

				synchronized by the batch job to show F6 for the PO. All of the details explained earlier for APPLIED configuration holds true here also.
PO_DOC_HDR	F2	PO_DOC_HDR	F2	No change
PO_DOC_HDR	F4	PO_DOC_HDR	F4	No change
PO_DOC_HDR	F5	PO_DOC_HDR	F5	No change

After synchronization, the batch process always leaves the configuration records in APPLIED state, that is, New fields equals Old fields.

If the batch process had to process configuration records for a Transaction Code with all of them defined as DELETE configuration (meaning no fields need to be additionally displayed on the Worklist Details records), then the job deletes all of the Worklist Details records for that Transaction Code.

The batch process generates a report listing the Worklist Details Configuration records that were selected for processing.

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

Process Steps	Messages
1. Parameter Validation	• Validating Batch Parameters • Parameter validation successful/failed. If the parameter is invalid, the invalid value is displayed in the log.
2. Initialize job resources	• Initializing Job resources
3. Worklist record processing	• Fetching records from Worklist Details Configuration. • Updating Worklist Details • Configuration records processed: <RecordCount>
4. Create Report	• Rendering report started • Rendering report completed
5. Job complete	• Run ended

Restartability Information

The job will not maintain any checkpoints. If the job encounters any errors while synchronizing data, users can view the resulting errors in the job log and re-schedule a new job after rectifying the issue. The selection logic will ensure the previously processed records will not be picked up again for processing.

Major Input

- Worklist Details Configuration (R_WF_DET_CNFG)
- Worklist (WF_APRV_WRK_LST)
- Worklist Details (WF_APRV_WRK_LST_DET)

Batch Parameters

Parameter	Description	Default Value	Explanation
CLIENT_NM	Client Name (Optional)	No Default	Optional name to appear on report header
COMMIT_SIZE	Commit Size (Optional)	500	Optional performance parameter
DOC_CD	Transaction Code	No Default	Transaction Code for which the batch job should process the changed configuration (marked as dirty) on the Configure Worklist Details page. The value can be a single Transaction Code, list of comma separated values of Transaction Codes, or left blank to select/process all of the Worklist Details Configuration records for all Transaction Codes on the Configure Worklist Details page.
MODE	Run Mode	1	(1) Report Mode - When the job is run in Report mode, the job will only report the list of Worklist Details Configuration records that were selected. No updates happen in this mode. This mode can be used just to know which Worklist Details Configuration records would be picked up for processing if the job was to be run in Update mode with same set of batch parameter values. (2) Update Mode - In this mode the job will update the Worklist Details table as per setup on the Worklist Details Configuration records that were selected for processing. It will also generate the report that lists the Worklist Details

Parameter	Description	Default Value	Explanation
			Configuration records that were picked up for processing.
RUN_TYP	Run Type	1	(1) Incremental - The job will only select Worklist Details Configuration records that were changed (marked as dirty) and if Mode = Update then process them. The selected set typically includes the set of User changes on the Configure Worklist Details page since the last job was run. This mode should be used during the regular frequent runs. (2) Full - the Transaction Code parameter value is ignored. All Worklist Details Configuration records from the Configure Worklist Details page are selected and if Mode=Update then processed as well. This rebuilds the Worklist Details table from scratch with all of the configurations setup on the Configure Worklist Details page. This Run Type does not consider whether the record was marked as dirty or not (recently changed since last job run or not), to summarize it processes dirty records as well as APPLIED configurations. This Run Type should be rarely used and can be used in cases where the Worklist Details table needs to be rebuilt from scratch or all Worklist Details configuration records irrespective of dirty/APPLIED status need to be processed.

Major Output

- Updates on Worklist Details and Worklist Details Configuration

- Report on selected Configuration for Worklist Details

Job Return Code

Return Code	Condition
Successful (1)	All records processed and report generated successfully.
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

The Worklist Details Configuration records are sorted by,

- Transaction Code (DOC_CD)
- Row Number (ROW_NO)

Selection Criteria

The batch process selects Worklist Details Configuration records for reporting/processing.

The selection criteria are based on values provided in the batch parameters:

- Run type (RUN_TYP)
- Transaction Code (DOC_CD)

If Run Type is ‘Incremental’ and Transaction Codes are provided in the Transaction Code batch parameter then the batch job selects all of the Dirty records for the Transaction codes specified in the Transaction Code batch parameter.

If the Transaction Code batch parameter value is blank, then it selects all Dirty records from the Worklist Details Configuration for processing.

If Run Type is ‘FULL’ then it selects all records existing in the Worklist Details Configuration table regardless of the records being Dirty or Non-Dirty and ignoring the Transaction Code batch parameter value as well.

Problem Resolution

Possible Return Codes	Condition	Recommendation	Other Instructions
Successful (1)	All of the Worklist Details Configuration and Worklist Details	N/A	N/A

Possible Return Codes	Condition	Recommendation	Other Instructions
	records are processed and the report is generated successfully.		
Failed (12)	Worklist Details Configuration and Worklist Details records not updated as expected.	Rectify the problem and schedule a new job.	View logs for help on the problem. Alternatively, the job can be rescheduled with a different set of parameters.

2.3 Administration Utilities Report Processes

The Advantage Administration Utilities report run sheet included in this section is:

- Separation of Duties Audit Report

2.3.1 Separation of Duties Audit Report

Chain or Job Name	Separation of Duties Audit Report
Recommended Frequency	Monthly and anytime there are major changes to the mapping of users to security roles, and so forth.
Single Instance Required	Yes
Can be restarted?	No
Reports generated	Process Report is Generated

Overview

The Separation of Duties Audit Report is a report that shows which users are failing to abide by the Restricted Security Role Group reference page. This page is used to setup the combination/group of security roles that cannot be assigned to the same user. Users that have these restricted security roles assigned to them will show up in this report.

Process Steps	**Messages**
1. Parameter Validation	• There are no Batch Parameters, so there are no parameters to validate • Batch Parameter validation completed
2. Selection of Records	• Selecting eligible records • Reports Output folder will be mapped. It will issue an error if the directory is not found. • If the selection returns 0 records, then the following message is issued: “0 rows of log table processed”. • Number of records (count) processed are displayed to show how many records should be displayed in the report • Render the report • At the end, the following message is issued: Run Ended.

Major Input

- The Restricted Security Role Group reference page, which defines the groups of Security Role IDs that cannot be assigned to the same user. The database table names are R_SC_SEP_DUTY_GRP and R_SC_SEP_DUTY_DET.
- No Parameters are needed for this job.

Major Output

The Separation of Duties report is the only output for this job.

Sort Criteria

The sort order for this job is by user ID and home department code. So the report will be ordered in this fashion.

Selection Criteria

It selects all of the users that are assigned for any combination of Security Role IDs that are specified on Restricted Security Role Group.

Problem Resolution

If the job fails, verify setup and schedule a new job.