Guidelines for Archiving Workflow Related Data Including Tasks and Alerts

IBM Cúram Social Program Management v6.0

Table of Contents

1 Purpose of Document

The purpose of this document is to provide high level guidelines to those who are attempting to archive Workflow, Task and Alert related data from a system.

The document describes the relationships between these tables. It also briefly describes the order in which data has to be archived from these tables.

1.1 Scope

A fully tested archiving process is not provided with IBM Cúram Social Program Management. As such, the purpose of this document is to provide guidance on a best effort basis to assist a customer implementing such a strategy.

This document only describes the Cúram Workflow, Task and Alert tables of version 6.0 of the IBM Cúram Social Program Management product. It does not describe any customizations made by the customer that would have to be taken into account when archiving such data.

Any data archiving should be first completed on a test system and a complete suite of test cases supplied before being applied to a live system.

This document assumes that a list of cases to be archived has already been established and provides guidance in retrieving the dependency records associated with these cases.

2 Workflow, Task and Alert Related Tables

The following is a brief description of the Workflow, Task and Alert related tables present the Cúram product. These tables are those that need to be considered if an archiving strategy is being considered for Workflow, Task and Alert related data.

2.1 Tables, Key Attributes and Descriptions

Table Name	Key Attribute	Description
ProcessInstance	processInstanceID	The run-time data for an enacted Workflow.
ActivityInstance	activityInstanceID	The run-time data for an activity in an enacted Workflow.
ActivityOccurrence	activityInstanceID	A Parallel Activity in a Workflow Process Definition allows a Workflow to perform the same step on multiple data items in parallel. The data items are obtained from a Parallel List WDO and one activity instance is created for each item in the list. The Parallel Activity Occurrence record tracks which item in the list is associated with each Activity Instance.
Task	taskID	A set of instructions to enable a user to perform the manual work required by an activity instance.
TaskWDOOverflow	taskID sequenceNo	The Workflow Engine creates a task whenever a specified activity type is executed (e.g. manual activity). At the time the task is created, the Workflow Engine takes a snapshot of any Workflow Data Object values that are

		used by the Task. This snapshot is stored as XML data in the “wdoSnapshot” attribute of the Task entity. If the XML string exceeds 3900 characters, it is broken up into a sequence of strings (0 based sequence incremented by 1), each with a maximum length of 3900 characters. These strings are then stored as overflow records.
EventWait	eventWaitID	The details of an event which must occur before an activity instance can be completed.
MatchedEvtArchive	eventWaitID	The details of an event which has been raised by the system.
WorkflowDeadline	deadlineID	The length of time that a task or event wait is allowed to remain unprocessed.
Reminders	reminderID activityInstanceID	The Process Definition Tool (PDT) allows the definition of an arbitrary number of Reminder Notifications for any Workflow activity that waits on an event. Any activity that waits on an event can optionally specify a deadline after which the Workflow engine should stop waiting and take some action. Reminders can then be set to send out notifications at certain times in advance of a deadline. This table stored instances of those reminder notifications.
TaskHistory	taskHistoryID	An event in the life cycle of a task. A row on this table stores the old and new values for a particular column value which has changed on the related Task row together with

		the type of change that has occurred.
BizObjAssociation	bizObjAssocID	An association of a Task with a row on a database table (i.e. a person or a case).
TaskAssignment	taskAssignmentID	The assignment of a Task to a User, Work Queue, Organization Unit, Position, or Job.
SuspendedActivity	activityInstanceID processInstanceID	The Activity Instances that have been suspended as a result of the suspension of an associated Process Instance
TransitionInstance	transitionInstanceID	The details of the attempt to navigate between two activity instances (the “from” activity instance and the “to” activity instance).
ProcInstWDOData	processInstanceID wdoName	The Process Definition Tool (PDT) allows the definition of Workflow Data Objects which hold shared data for activities in a Workflow. At Workflow execution time, instances of these Workflow Data Objects are wrapped in an XML string and stored as Process Instance WDO Data records (one record per Workflow Data Object).
ProcInstOverflow	processInstanceID wdoName sequenceNo	The Workflow Engine stores Workflow Data Object values as ProcInstWDOData records. These records contain an XML string describing the Workflow Data Object, its attributes and the current values of those attributes for a given process instance. If the XML string exceeds 3900 characters, it is broken up into a sequence of strings ((0 based sequence incremented by 1), each with a maximum

		length of 3900 characters. These strings are then stored as overflow records.
WDOValuesHistory	wdoValuesHistoryID	An audit log of the process instance WDO data used as input and output for activities in a Workflow process instance.
WorkflowHistory	workflowHistoryID	An audit log taken to record changes to the status of an executing Workflow process instance by an administrator.
FailedMessage	failedMessageID	Workflow recovery data arising from a failure in business application processing.
Iteration	iterationID	The details for each iteration of a loop block in a Workflow process instance.
JoinInstance	joinInstanceID	A count on the number of parallel branches remaining to be synchronized at an instance of a Workflow join (synchronization point) in a Workflow process instance.
Alert	alertID	A piece of information that may be of interest to a user or a set of users. It is not a request of action by the user and therefore there is no action or work to be managed or monitored by a supervisor.
AlertWDOOverflow	alertID sequenceNo	The Workflow Engine may create an alert whenever a Notification is delivered. At the time that the Alert is created, the Workflow Engine takes a snapshot of any Workflow Data Object values that are used by the Notification. This snapshot is stored as XML data in the wdoSnapshot attribute of the Alert entity. If

		the XML string exceeds 3900 characters, it is broken up into a sequence of strings ((0 based sequence incremented by 1), each with a maximum length of 3900 characters. These strings are then stored as overflow records.

3 Relationships between Tables

3.1 Foreign Key Relationships

The following is a brief description of the relationships that exist between the various tables described above.

The starting point for describing the table relationships is the **ProcessInstance** table as it is this table that will act as a starting point for the archiving of the Workflow, Task and Alert related data. Therefore the relationships between the other tables and the ProcessInstance table will be described here.

The unique key on the ProcessInstance table is the '*processInstanceID*' attribute.

Parent Table Name	Child Table Name	Foreign Key Attribute
ProcessInstance	ActivityInstance	processInstanceID
ActivityInstance	ActivityOccurrence	activityInstanceID
ActivityInstance	EventWait	activityInstanceID
ActivityInstance	MatchedEvtArchive	activityInstanceID
ActivityInstance	WorkflowDeadline	activityInstanceID
ActivityInstance	Reminders	activityInstanceID
ActivityInstance	Task	taskID
Task	TaskWDOOverflow	taskID
Task	TaskHistory	taskID
Task	BizObjAssociation	taskID
Task	TaskAssignment	taskID
ProcessInstance	SuspendedActivity	processInstanceID
ProcessInstance	TransitionInstance	processInstanceID
ProcessInstance	ProcInstWDOData	processInstanceID
ProcessInstance	ProcInstOverflow	processInstanceID
ProcessInstance	WDOValuesHistory	processInstanceID

ProcessInstance	WorkflowHistory	processInstanceID
ProcessInstance	FailedMessage	processInstanceID
ProcessInstance	Iteration	processInstanceID
ProcessInstance	JoinInstance	processInstanceID
ProcessInstance	Alert	processInstanceID
ProcessInstance	AlertWDOOverflow	alertID for an alert record which contains a processInstanceID

4 Archiving Data Strategy

4.1 Order of Data Archiving

When considering an arching strategy in the Workflow area there is some vital information that should be at the forefront of any process: The relationships between the tables being archived, the starting point for the archiving, and the order in which to do it. In the previous sections we described the tables involved and the relationships between these tables. In this section we describe the starting point and the order in which the archiving process should be carried out in.

a. **Select the parent records to archive:**

  a. Select all of the process instances on the system that have a status of ‘**COMPLETED**’

  b. Ensure that none of these completed process instances selected are related to the enactment of **synchronous sub-flow** process definitions whose **parent** process instances have not yet completed. These synchronous sub-flow process instances should not be archived until the associated parent process instances have completed. To ensure this, each process instance record selected above should be checked as follows:

    a. Check if the ‘PROCESSTYPE’ attribute of the process instance record has a value of ‘PT2’ (i.e. a synchronous sub-flow Workflow process instance). If it does, retrieve the ‘PARENTACTINSTID’ attribute of the process instance record. The value of this attribute relates to the identifier of the activity instance on the parent process instance that was used to enact the synchronous sub-flow process.

    b. Using this value, retrieve the associated process instance identifier (for the parent process instance).

    c. Using this identifier, retrieve the status of the process instance (this is the status of the parent process instance for the associated synchronous sub-flow process instance). If the status of this process instance is not ‘COMPLETED’, then remove the synchronous sub-flow process instance record from the list of process instances to archive.

  c. Select all of the activity instances and associated task identifiers associated with the list of ‘COMPLETED’ process instances that are left after the above processing has been done.

  d. This now leaves three lists of parent records from which to archive all of the required data from. These are:

    a. List of completed process instances.

   b. List of activity instances associated with these completed process instances.

   c. List of tasks associated with these activity instances.

b. **Archive records associated with selected tasks:**

   a. For each task record selected, archive any associated task WDO overflow (TaskWDOOverflow) records.

   b. For each task record selected, archive any associated task history (TaskHistory) records.

   c. For each task record selected, archive any associated business object association (BizObjAssociation) records.

   d. For each Task record selected, archive any associated Task Assignment (TaskAssignment) records. This step is just a sanity check and should result in 0 records being archived. This is due to the fact that Task Assignment records are deleted when the associated Task is closed.

c. **Archive records associated with selected activity instances:**

   a. For each activity instance selected, archive any associated activity occurrence (ActivityOccurrence) records.

   b. For each activity instance selected, archive any associated event wait (EventWait) records. NOTE: this step is just a sanity check and should result in 0 records being archived. This is due to the fact that event wait records are archived to the MATCHEDEVTARCHIVE table when the event is matched and the associated activity instance is completed.

   c. For each activity instance selected, archive any associated matched event wait (MatchedEvtArchive) records.

   d. For each activity instance selected, archive any associated Workflow deadline (WorkflowDeadline) records. NOTE: this step is just a sanity check and should result in 0 records being archived. This is due to the fact that Workflow deadline records are deleted when the associated task is closed and activity instance is completed.

   e. For each activity instance selected, archive any associated reminder (Reminders) records.

d. **Archive records associated with selected process instances:**

   a. For each process instance record selected, archive any associated suspended activity (SuspendedActivity) records. NOTE: this step is just a sanity check and should result in 0 records being archived. This is due to the fact that there should be no activity instances in a suspended state if the associated process instance is complete.

   b. For each process instance record selected, archive any associated transition instance (TransitionInstance) records.

   c. For each process instance record selected, archive any associated process instance WDO overflow data (ProcInstOverflow) records.
   d. For each process instance record selected, archive any associated process instance WDO data (ProcInstWDOData) records.
   e. For each process instance record selected, archive any associated Workflow WDO values history (WDOValuesHistory) records.
   f. For each process instance record selected, archive any associated Workflow history (WorkflowHistory) records. NOTE: this step may result in the archiving of 0 records if status of none of the process instance records has been changed by the administrator.
   g. For each process instance record selected, archive any associated failed message (FailedMessage) records. NOTE: this step may result in the archiving of 0 records if there have been no business application processing failures for any of the process instances being archived.
   h. For each process instance record selected, archive any associated iteration (Iteration) records.
   i. For each process instance record selected, archive any associated join instance (JoinInstance) records.
   j. For each process instance record selected, archive any associated alert WDO overflow (AlertWDOOverflow) records. NOTE: this step may result in the archiving of 0 records if all of the alerts associated with the process instance have been viewed and subsequently deleted by the relevant users.
   k. For each process instance record selected, archive any associated alert (Alert) records. NOTE: this step may result in the archiving of 0 records if all of the alerts associated with the process instance have been viewed and subsequently deleted by the relevant users.

e. **Archive the parent records described in section “a” above:**
   a. Archive each of the select tasks.
   b. Archive each of the selected activity instances.
   c. Archive each of the selected process instances.