

Configure Long-term Archiving

Last Modified on 11/16/2021 5:39 am EST

V9.4

Overview

The long-term archiving feature extracts the closed workflow instances' data from the Cora SeSequence database, and moves it to the archive database. Effective archiving configuration makes sure that only operational data resides in the production database, thus improving the system performance.

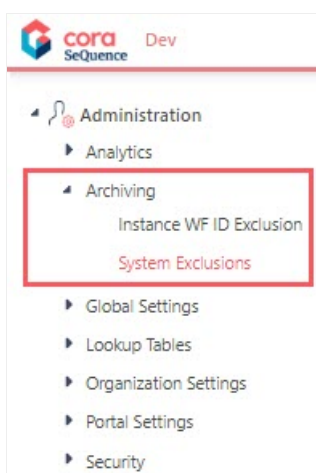
Workflow instances that match the archiving configuration settings are referred to as the "Archiving Population". Archiving is generally performed in the master workflow context.

For example, if archiving is set for 1 year, then the master workflow that has been closed for one year will be archived along with its sub-workflows.

You can configure archiving at different levels, based on different settings defined while setting up the archiving jobs.

- System configuration: This is a general system archiving setting. This time configuration archives all closed workflow instances, except the ones configured differently, and excluded explicitly from the configuration.
- Workflow space configuration: With this configuration, you can have different archiving configurations for different workflow space according to variables and time filters. The archiving is performed per master workflow instance ID.

Both configurations allow you to exclude some workflow instances, templates, and tables from archiving. These *excluded items* are not archived even if the configuration filters define them as part of the archiving population. In the **Administration > Archiving**, you have new settings, **System Exclusions** and **Instance WF ID Exclusion** to add excluded items for system and workflow space configurations respectively.



IMPORTANT

Archiving is irreversible. After archiving, the data is deleted from the Cora SeSequence database.

Configuration steps

All the archiving configuration steps are defined in the Administration site.

	Step	Performed by
1	Define archiving data	<ul style="list-style-type: none">• Architect• Tech Lead• Developer
2	Deploy archive database	<ul style="list-style-type: none">• Cora SeQuence Administrator• DBA
3	Create archive connection string	<ul style="list-style-type: none">• Architect• Tech Lead• Developer
4	Set up archiving jobs <ul style="list-style-type: none">• Archiving Data Collector• Archiving Data Worker• Archiving Files Worker (optional)	<ul style="list-style-type: none">• Architect• Tech Lead• Developer



Watch [this video](#) to view a demonstration of the archiving configuration steps.

The below sections explain each archiving configuration step in detail.

1. Define archiving data

Defining archiving data is same for system and workflow configuration. You define parameters in the JSON code snippet that is added in the Archiving Data Collector job, to determine which data from the Cora SeQuence database will be retrieved for archiving, and/or deleted from the database.

The same JSON code also determines, which configuration will be used for archiving, system or workflow.

For example, no value for workflowSpaceId parameter suggests that system configuration will be used for archiving.

View a [template](#) of the JSON code snippet.

Use this as a sample for reference only, and do not copy paste from this template. Make sure to delete comments from JSON code snippet, before adding to the job.

JSON parameters

You need to set up two main parameters for archiving: `operation` and `filters`.

Parameter	Description
<p><code>operation</code></p>	<p>The operation that needs to be performed. Filters in the JSON are defined per this operation.</p> <p>Value:</p> <ul style="list-style-type: none"> • Archive • Delete
<p><code>filters</code></p>	<p>Array of filters Each configuration setting can have multiple filters for an archiving job.</p>
<p><code>name</code></p>	<p>Name of the archiving configuration.</p>
<p><code>workflowPopulationFilter</code></p>	<p>Array of filters The filters based on which the archiving population is decided.</p>
<p><code>workflowSpaceId</code></p>	<p>Determines which workflow configuration will be considered for archiving. To use system archiving configuration, delete this parameter.</p> <p>Value: Workflow space template GUID</p> <p>The workflow instance determines the archive or delete date.</p>

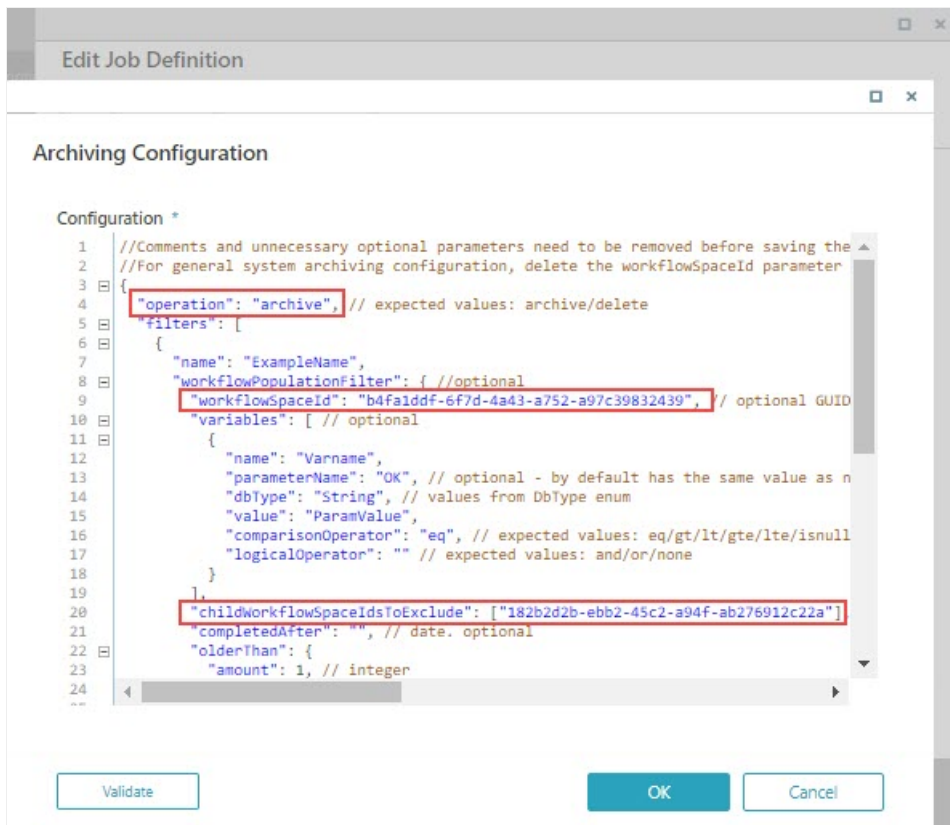
Parameter	Description
<p><code>variables</code></p>	<p>Array of variables For each variable, set these parameters:</p> <ul style="list-style-type: none"> • name: name of variable in <code>tblInstanceWorkflowsDataPropagationClosed</code> • parameterName: by default, same value as name. • dbType: variable datatype for example, "dbType": "String", • value: permanent value of variable to transfer • comparisonOperator: comparison operator to check the variable value <p>Expected values:</p> <ul style="list-style-type: none"> ◦ eq ◦ gt ◦ lt ◦ gte ◦ lte ◦ isnull ◦ isnotnull <ul style="list-style-type: none"> • logicalOperator: <ul style="list-style-type: none"> Expected values: ◦ - OR ◦ & - AND ◦ NONE
<p><code>childWorkflowSpaceIdsToExclude</code></p>	<p>Determines which sub workflows will be excluded from archiving.</p> <p>Example: ["182b2d2b-ebb2-45c2-a94f-ab276912c22a"]</p>
<p><code>completedAfter</code></p>	<p>Sets the date after which data will be archived or deleted.</p>

Parameter	Description
<p><code>olderThan</code></p>	<p>Sets the upper limit date for archiving data. All the data created before this date is archived.</p> <p>Set these parameters within:</p> <ul style="list-style-type: none"> • amount - number • timeUnit - date unit like day (d), month (m), or year (y) <p>For example, amount set as 1, and timeUnit set as 0 archives all the data that is 1 day older than the current date.</p> <div data-bbox="804 573 1415 692" style="background-color: #e1f5fe; padding: 5px;"> <p>NOTE Only one from <code>completedAfter</code> and <code>olderThan</code> are considered while archiving.</p> </div>
<p><code>ignoreOpenChildren</code></p>	<p>Determines whether the open workflows will be archived or not.</p> <p>Value:</p> <ul style="list-style-type: none"> • true • false (default) <p>If set to false, the master workflow is not archived until all sub workflows are closed.</p> <p>If set to true, the master workflow is archived with closed sub workflows only, and does not check for open sub workflows. The open sub workflows remain in the Cora SeQuence database.</p>
<p><code>archiveAsStandAlone</code></p>	<p>Determines which configuration will be used for archiving, system or workflow.</p> <p>Value:</p> <ul style="list-style-type: none"> • true • false (default) <p>If set to true, the master workflow archiving configuration is considered.</p>
<p><code>tablesToExclude</code></p>	<p>Array of tables to be excluded from archiving</p> <p>Set these parameters within:</p> <ul style="list-style-type: none"> • name: table name • schema: table schema
<p><code>tablesToInclude</code></p>	<p>Array of tables to be included for archiving.</p> <p>Set these parameters within:</p> <ul style="list-style-type: none"> • name: table name • schema: table schema

Parameter	Description
<code>enabled</code>	<p>Determines if the JSON configuration is enabled or disabled.</p> <p>Value:</p> <ul style="list-style-type: none"> • true (default) • false
<code>simulationMode</code>	<p>Determines whether, after archiving, the data will be removed from the Cora SeQUENCE database.</p> <p>Value:</p> <ul style="list-style-type: none"> • true • false (default) <p>If set to false, the user can validate the data before deletion from the database.</p> <p>In Simulation mode the data is not written to the file <code>arc.tblInstanceWorkflowClosed</code>, but instead is written to <code>arc.tmpTblInstanceWorkflowsClosed</code>.</p> <p>The simulation can be run only once.</p>
<code>moveFilesToFilesArchiveStorage</code>	<p>Determines whether archived files will be moved to archive file storage.</p> <p>Value:</p> <ul style="list-style-type: none"> • true (default) • false <p>If set to true, the actual files (blobs) are moved to the archive file storage.</p>

Consider the following factors, when you create the JSON code snippet:

- If you want custom data to be archived or deleted, you need to add the custom table's fields to the JSON in the `tablesToInclude` filter.
- If you do not want to archive all the data, you can use the `Variables` filter to specify the column that should be filtered out, its value, data type, and conditions with comparison operator and logical operator, such as AND or OR.



2. Deploy archive database

To use the archive feature in Cora SeSequence, deploy an archive database on an SQL server. It is recommended to deploy the archive database on a separate SQL server than the Cora SeSequence operational database.

Before you deploy the archive database, make sure that:

- The SQL server is setup.
- The latest [PowerShell modules](#) are installed.

Click [here](#) to view a sample PowerShell script. Consider this template as a sample only, and do not copy paste the same.

Procedure

1. Run the [Install-CoraSeSequenceArchiveDatabase](#) function.

NOTE

Database files are installed in default locations, as configured on the SQL server. If required, after installation, you can move the files using the standard procedure.

For details on the archive database structure, see [this article](#).

3. Create the connection string

For archive feature to move collected data to archive database, you need to set up a connection between the Archiving Data Worker job and the archive database.

Prerequisites

Before you create the connection string, make sure that:

- You have the archive connection details, such as:
 - Connection type
 - End points
 - Required credentials
- Configure MSDTC between the operational database and the archive database. For details, see [this article](#).

Procedure

1. Go to **Administration > Global Settings > Connection Strings**, and click **Add New Record**.
2. Enter a name for the connection string. This name will be displayed in the list of connections to choose from, while setting up the archiving job.
3. Select the Provider as System.Data.SqlClient.
4. Enter the connection string.
For example, `Provider=SqlClient Data Provider (System.Data.SqlClient)Persist Security Info=False;User ID=sa;Initial Catalog=enableSP2;password=*;Data Source=1.1.1.1`
5. Click **Add**.

You can also create a connection string while setting up the Archiving Data Worker job.

4. Set up archiving jobs

Data archiving is performed by JES jobs that you need to set up on the Administration site.

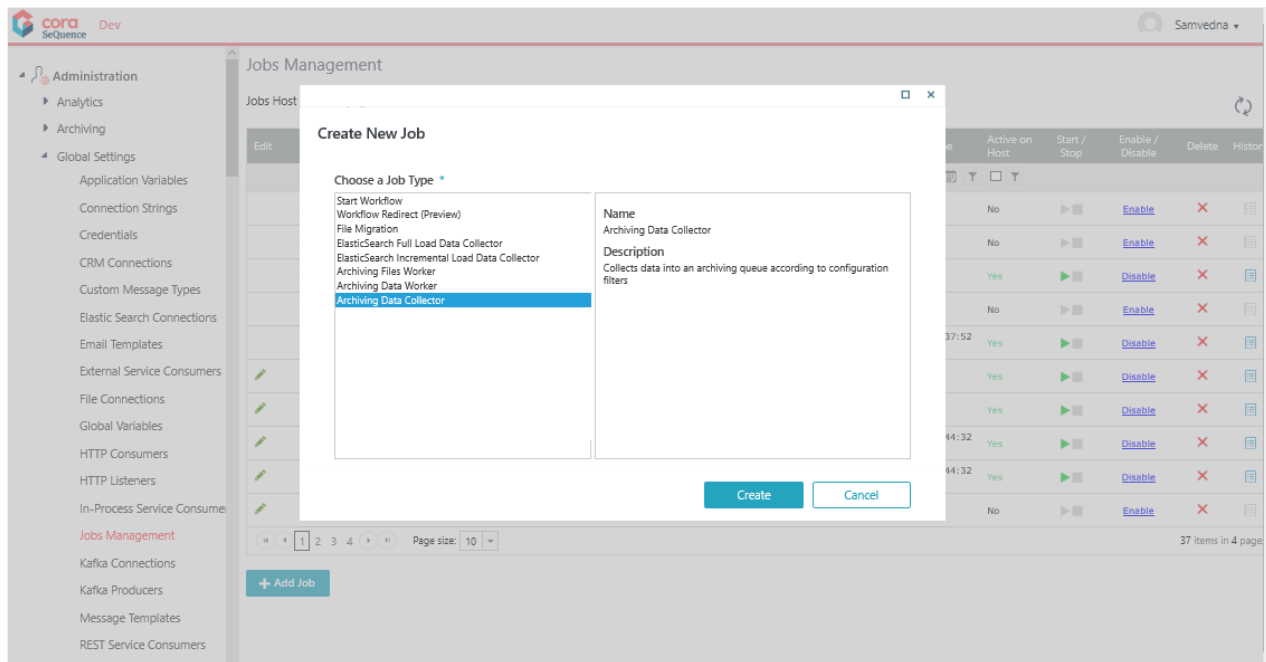
Go to, **Administration > Global Settings > Jobs Management**, and click **Add Job**.

- **Archiving Data Collector:** Collects the relevant data from the Cora SeQuence database based on the JSON, and adds it to the queue.

NOTE

- Depending on the size of your database, this job can take some time. To avoid performance issues, configure it to run during the low-peak hours or weekend.
- Make sure this job is scheduled to run at least one hour *after* the *BRS Closed Instances Separator Thread* is scheduled to move data to *closed* tables.

- **Archiving Data Worker:** Runs right after the Archiving Data Collector job, and makes sure that the collected data is moved to the archive database defined in the connection string.
- **Archiving Files Worker** (optional): Migrates the archived attachments to an external storage location.



Prerequisites

Before you create the archiving jobs:

- Make sure that the *BRS Closed Instances Separator Thread*, to move closed workflow instances to the archive tables, is already scheduled to run, and you know the schedule. For more details, see [this article](#).

Configure Archiving Data Collector job

1. Select **Archiving Data Collector** on Create New Job screen, and click **Create**.
2. On the **Job** tab of job definition screen, define the following:
 - **Name:** Enter a meaningful name for the data collector job.
 - **Job is enabled:** Select this option only after you complete the workflow, or if you want to run the job for testing purposes.
3. Click **Next**.
4. On the **Command** tab, set the following:
 - **Archiving Configuration:** Click the ellipsis to paste the JSON code snippet. Click Validate to validate the JSON code. An alert message is displayed in case of validation failure, make the required changes.

NOTE

Make sure that you delete any comments from the JSON code before adding.

- **Work Batch Size:** Batch Size is *the number of records* that are synced at one time. Depending on the complexity of the cases, you may consider to set a smaller batch size. The default is set to 100. You can enter up to 10000.
 - **Batch Iteration Delay (seconds):** Time in seconds, after which the next batch will be synced. Default is 5 seconds.
5. Click **Next**.
 6. On the **Advanced Options** tab, set the following:
 - **Job Execution Schedule:** Set the job to execute once a day.
 - Select the "If a job is currently running, do not start the next job..." check box, to avoid congestion.

- If not defined, job will run with default options.
7. Click **Finish**.

Configure Archiving Data Worker job

The Archiving Data Worker job moves the queued data, collected by Archiving Data Collector Job, into the archive database. Configuring the data worker job is similar to configuring the data collector job.

1. Select **Archiving Data Worker** on Create New Job screen, and click **Create**.
2. On the **Job** tab of job definition screen, define the following:
 - **Name:** Enter a meaningful name for the data worker job.
 - **Job is enabled:** Select this option only after you complete the workflow, or if you want to run the job for testing purposes.
3. Click **Next**.
4. On the **Command** tab, set the following:
 - **Archiving Connection String:** Add the relevant connection string to connect job to the archive database. Choose from the list, the connection string you have already set up.
OR
Click **Add** to set up, and add a new connection string.
 - **Work Batch Size:** Batch Size is *the number of records* that are synced at one time. Depending on the complexity of the cases, you may consider to set a smaller batch size. The default is set to 100. You can enter up to 10000.
 - **Batch Iteration Delay (seconds):** Time in seconds, after which the next batch will be synced. Default is 5 seconds.
 - **Retry Stuck Data After (hours):** Time in hours, after which the job will retry to push the stuck data to the archive database. Default is 24 hours.
5. Click **Next**.
6. On the **Advanced Options** tab, set the following:
 - **Job Execution Schedule:** Set the job to execute once a day.
 - Select the "If a job is currently running, do not start the next job..." check box, to avoid congestion.
 - If not defined, job will run with default options.
7. Click **Finish**.

Configure Archiving Files Worker job

The Archiving Files Worker job migrates all the archived attachment files to an external storage location.

Configuring the files worker job is similar to configuring the data worker job. Just note the following settings:

- On Create New Job screen select,
 - **Job Type:** Archiving Files Worker
- On the **Command** tab, set the following:
 - **Separate connection for files:** To connect this job to the external storage, for file migration.
 - **File Storage Connection:** Select from the list a connection string to connect this job to the external storage.
 - **Work Batch Size:** Batch Size is *the number of records* that are synced at one time. Depending on the complexity of the cases, you may consider to set a smaller batch size. The default is set to 100. You can enter up to 10000.
 - **Batch Iteration Delay (seconds):** Time in seconds, after which the next batch will be synced.

Default is 5 seconds.

- **Retry Stuck Data After (hours):** Time in hours, after which the job will retry to push the stuck data to the archive database. Default is 24 hours.

For the detailed procedure, see the *Configure Archiving Data Worker job* section above.