

Scheduler Guide Axiom Software Version 2018.1



Kaufman Hall® is a trademark of Kaufman, Hall & Associates, LLC. Microsoft®, Excel®, Windows®, and SQL Server® are trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

This document is Kaufman, Hall & Associates, LLC Confidential Information. This document may not be distributed, copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable format without the express written consent of Kaufman, Hall & Associates, LLC.

Copyright © 2018 Kaufman, Hall & Associates, LLC. All rights reserved. Updated: 4/27/2018

Kaufman Hall Axiom Software 10260 SW Greenburg Road, Suite 710 Portland, Oregon 97223 877.691.9969 (Toll-Free) 503.977.0234 (Local) www.kaufmanhall.com

# **Table of Contents**

Chapter 1: Introduction	1
Chapter 2: Scheduler Overview	2
About Scheduler	2
The Scheduler dialog	4
Chapter 3: Scheduler Setup	7
Enabling Scheduler services	7
Configuring Scheduler system jobs	8
Chapter 4: Scheduler Job Setup	11
Managing jobs and tasks	11
Defining scheduling rules for a job	14
Setting up email notification for jobs	17
Creating event handlers for a job	
Using job variables	20
Processing tasks in parallel	23
Using iterative task processing	25
Using iterative task processing	28
Conditionally processing tasks in a job	32
Processing priority for scheduled jobs	34
Running a job	35
Job properties	36
Chapter 5: Scheduler Administration	42
Managing scheduled jobs	42
Viewing job results	44
Managing Scheduler servers	45
Managing event handlers	47
Managing remote data connections	49
Chapter 6: Advanced Scheduler Features	51
Using file watcher to trigger imports	51
Using RunEvent to execute a job	55
Chapter 7: Scheduler Task Reference	62
Task Control properties	63
Active Directory Import task	65

Collect Worksheets task	68
Create Plan Files task	71
Echo task	72
Execute Command Adapter task	73
Execute SQL Command task	73
Export ETL Package task	74
File Processing task	74
Import ETL Package task	77
Process Plan Files task	78
Process Document List task	88
Process Template List task	94
Purge System Data task	96
Raise Event task	98
SMTP Message Delivery task	99
Start Process task	99
Update Indexes and Constraints task	100
Index	101

# Introduction

Axiom Software's Scheduler is used to schedule tasks for automated execution and perform those tasks using server-side processing. Various tasks can be scheduled, such as processing plan files, file processing, importing data, etc.

#### Intended audience

This guide is intended for administrators and other power users who are responsible for scheduling tasks for automated execution.

### What is covered in this guide?

This guide covers the following aspects of Scheduler setup and management:

- Performing initial Scheduler setup
- Creating jobs and tasks and scheduling them for execution
- Using advanced Scheduler features such as variables and event handlers
- Reference for all job and task properties

#### What is not covered in this guide?

The following related topics are not covered in this guide:

- Security setup for Scheduler. For more information, see the System Administration Guide.
- Explanation of certain features that can be scheduled. For example, if you want to schedule a data import, this guide assumes that you already know what a data import is and that the import package has already been created. This guide only explains how to set up that import for execution in Scheduler.

All documentation for Axiom Software can also be accessed using the Axiom Software Help Files.

#### Axiom Software Client versions

This guide discusses functionality that is available in the Axiom Excel Client and the Axiom Windows Client. Screenshots of features may show either client. The Axiom Software functionality is virtually identical and applies to both clients.



# Scheduler Overview

Using Scheduler, you can schedule certain Axiom Software tasks to be processed on a Scheduler server at a specific date and time. For example, you can schedule a data import, or a refresh of plan files.

Processing tasks using Scheduler has advantages over manual processing, such as:

- Leverages the server's processing power and frees up your computer's resources.
- Enables recurring scheduling of ongoing tasks.
- Allows tasks to be scheduled during "off hours," during periods of low network and system activity.
- Allows tasks to be performed in batch, including enforcing task dependencies.

Scheduler processes tasks using jobs. Each job is a scheduled unit that can contain one or more tasks. The tasks in a job can be processed sequentially or concurrently as appropriate.

Only system administrators and users with the **Scheduled Jobs User** security permission can access Scheduler.

## **About Scheduler**

This section contains conceptual information about the Scheduler feature in Axiom Software.

Scheduler jobs and tasks

Scheduler processes tasks by using jobs. Each job is a scheduled unit that can contain one or more tasks.

When you create a job, you define the following basic properties:

- The schedule of the job, including recurrence (if any)
- The priority of the job
- The notification options for the job
- The tasks to perform for the job

The tasks define the actual activities to be performed by the job, such as importing data or processing plan files. Some tasks correspond to existing features that can also be processed manually (such as Process Plan Files), while other tasks are Scheduler-specific and can only be processed via Scheduler. Each task has a unique set of options that are specific to that task and to the activity to be performed.

The tasks in a job can be processed sequentially or concurrently as appropriate. Tasks can be dependent on other tasks in the job (for example, if Task 1 has errors, do not execute Task 2).

For more information on the available task types, see Scheduler Task Reference.

### Job execution and security

When a job is executed by Scheduler, it is run using a particular user identity. In order for a job to be valid, the user must be an active user defined in Axiom Software security, and the user must have the appropriate security permissions to perform the tasks in the job. The user identity for a job is determined as follows:

- If a job is a system job, then it is run as the system-managed identity of Scheduler Service ("System") instead of a user identity.
- If a job is run by clicking Run Now, then it is run as the user who placed the job on the schedule.
- If a job is run by an active scheduling rule, then it is run as the *job owner*. The job owner is the user who last saved the job.
- If a job is run via an event handler, then the job may be run as either the job owner, or the job requester (the user who triggered the event handler). This is specified when configuring the event handler.

### System jobs

Scheduler can execute user-created jobs and system jobs. Generally speaking, a system job is a job that needs to be run to facilitate system functionality, and should not depend on any particular user rights.

System jobs have two defining characteristics:

- System jobs are run under the system-managed identity of "Scheduler Service" instead of a user identity. This identity has full rights to the system as necessary to perform system tasks.
- For on-premise systems, system jobs are run by the default System Scheduler service. This service
  is created and started automatically on the Axiom Application Server, and does not require a
  separate installation. This service is exclusively for running system jobs. In Scheduler, the System
  Scheduler service is listed on the Servers tab (Service > Servers) using the following naming
  convention: <ServerName>-System.

Cloud Service systems do not have a separate System Scheduler service. System jobs are run using the available Scheduler services for the cloud system.

Because system jobs may run frequently and produce many results, these jobs are hidden by default in the **Result History** tab. You can clear the **Hide system jobs** check box at the bottom of the dialog to see the history for these jobs (or simply open the job to view the history).

There are two types of system jobs in Axiom Software:

- System jobs that are automatically created by Axiom Software to support certain system
  functionality. Some system jobs are created on initial startup and are intended to run on an
  ongoing basis, such as the system job to manage SMTP email delivery. Other system jobs are
  created on-demand in response to system events, using Scheduler event handlers. Only
  administrators can edit these system jobs.
- Jobs that are created manually by a user and then flagged as a system job. When a job is designated as a system job, the job runs on the dedicated System Scheduler service, using the Scheduler Service identity. For example, you might choose to run the Active Directory Import task as a system job, to avoid any interruptions due to user changes.

To designate a job as a system job, enable **Mark as System Job** in the **General** job properties. The following rules apply to manually-created system jobs:

- Only system administrators can designate a job as a system job.
- The job cannot contain any tasks that are designated as "non-system" tasks. Non-system tasks are any tasks that might involve spreadsheet processing, such as Process Plan Files.

# The Scheduler dialog

The Scheduler dialog is used to create and manage Scheduler jobs.

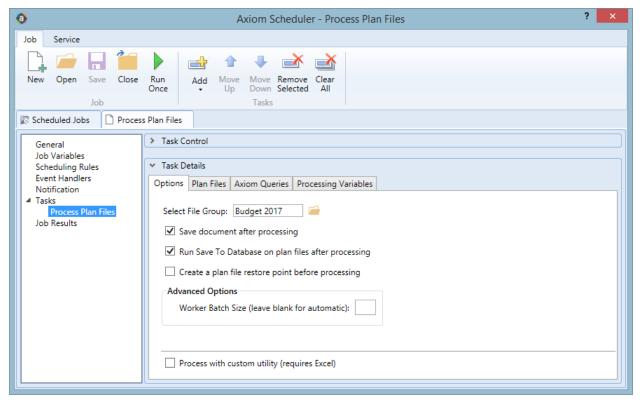
To access Scheduler:

• On the Axiom tab, in the Administration group, click Manage > Scheduler.

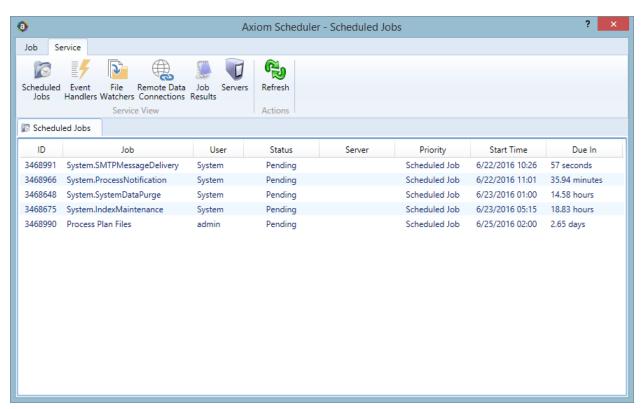
The top section of the **Scheduler** dialog contains a ribbon-style toolbar with two tabs: **Job** and **Service**.

- On the Job tab, you can create, run, and edit jobs.
- On the Service tab, you can manage scheduled jobs, view job results, and manage Scheduler servers.

As you perform actions on the **Job** and **Service** tabs, additional tabs are opened in the navigation pane of the dialog. For example, clicking the **Scheduled Jobs** button on the **Service** tab opens the **Scheduled Jobs** tab in the navigation pane. You can move between any open tab in the navigation pane, regardless of which tab is selected in the ribbon. The ribbon updates to show the related commands for the selected item.



Example Job tab



Example Service tab

When you right-click a tab in the dialog's navigation pane, you can close or save items as follows:

- For all items, you can Close, Close All, or Close All But This.
- For jobs, you can Save or Save As. Selecting Save As allows you to save a copy of the job to the Scheduler Jobs Library in the Axiom Software file system.

The Scheduler Jobs Library is also accessible via Axiom Explorer.

# Scheduler Setup

If this is a new Axiom Software installation, certain setup activities need to be performed to enable Scheduler functionality for each system.

# **Enabling Scheduler services**

All Axiom Software systems must have one or more Scheduler services available to execute Scheduler jobs. These Scheduler services must be explicitly enabled within the Scheduler dialog before they are available for use.

- For on-premise systems, these are the Scheduler services that you manually installed as part of the Axiom Software server installation.
- For Cloud Service systems, these Scheduler services were created for you when your cloud system was created.

#### To enable Scheduler services:

1. In the Scheduler dialog, on the Service tab, click Servers.

You should see one or more servers listed in this tab, representing your manually installed services or the services created for your cloud system. These are the servers that you must enable for processing.

You may also see the following items in the **Servers** tab:

- AppServerName-System: For on-premise systems, this is the system Scheduler service that was installed automatically as part of your Axiom Application Server installation. This service should already be enabled. It does not require any special setup.
- ServerName-CIS: For Cloud Service systems, this is the Cloud Integration Service that was manually installed to facilitate connections between your local environment and your cloud system. For more information on this service, see the Cloud Service Technical Guide.
- 2. Select each server in the list. In the **Configuration details** section at the bottom of the dialog, select the **Schedule Processing Enabled** check box, and then click **Update**.

**NOTE:** The **Configuration details** section also contains several configuration settings for the Scheduler service. We recommend leaving the default settings unless you are instructed to change them by Kaufman Hall Software Support. For more information, see Managing Scheduler servers.

These Scheduler servers are now available to process Scheduler jobs.

# Configuring Scheduler system jobs

Scheduler includes several built-in system jobs that should be configured as part of your system setup activities. These jobs are created automatically by the system Scheduler service when it is started. These jobs control email notifications for Scheduler and other system processes, and perform periodic database cleanup.

**NOTE:** Only administrators can edit these system jobs.

To configure the Scheduler system jobs:

1. In the Scheduler dialog, on the Service tab, click Scheduled Jobs.

You should see the following scheduled jobs where the name starts with **System**:

- System.SMTPMessageDelivery
- System.ProcessNotification
- System.SystemDataPurge
- System.IndexMaintenance
- 2. Double-click a job to open it for editing.
  - In the Tasks section, select the task name to access the task settings, and then edit the Task
     Details as necessary. See the following section for more details on the specific task settings.
  - In the **Notifications** section, configure the email notifications for the job as desired. By default, the jobs are configured to send notifications on error, however, you must specify a valid email address to receive these notifications (such as the email address of a system administrator).
- 3. Save the modified job. The changes will apply the next time the job is processed.

The changes will apply the next time the job is executed.

**NOTE:** You can also access these jobs in Axiom Explorer, in  $\Axiom\Axiom\$  System\Scheduler Working Folder.

If one of these jobs is removed from the schedule or deleted, you can restore it by right-clicking any system job or event handler and selecting **Create system jobs and event handlers**. Any unscheduled or deleted system jobs will be restored.

System job overview

The following table provides a brief description of the purpose of each job and its task settings.

Job	Information
SMTPMessageDelivery	<ul> <li>Description: This job enables email notifications for Scheduler jobs and other system processes, such as process management notifications.</li> <li>Email notifications are stored in the database until the SMTP Message Delivery task picks them up and sends them.</li> </ul>
	<ul> <li>Configuration: This task is not operational until you specify the name of a valid SMTP server in your environment. Once you have completed the SMTP server settings, make sure to disable Test Mode so that email notifications will be sent. For more details on the task settings, see SMTP Message Delivery task.</li> </ul>
	<ul> <li>Schedule: By default, the job is set to execute continuously (once per minute). We recommend leaving it at this frequency to ensure timely delivery of notifications, however, you can set the job to any desired frequency.</li> </ul>
ProcessNotification	<ul> <li>Description: This job enables reminder notifications for process management. These are notifications to remind users of upcoming due dates and overdue tasks.</li> </ul>
	Configuration: This task does not have any editable settings.
	<ul> <li>Schedule: By default, this job is set to execute once per hour. You should not change the frequency of this job.</li> </ul>
SystemDataPurge	<ul> <li>Description: This job purges old data from the system, such as old Scheduler job results, old email notifications, old temp table data, and old audit data. This task should be run regularly to help keep old and unnecessary data from impacting system performance.</li> <li>Configuration: This task is configured to delete items that are older than a specified number of days (by default, 15 days). You can change the default settings for this task if desired. For more details on the task</li> </ul>
	<ul> <li>settings, see Purge System Data task.</li> <li>Schedule: By default, this job is set to run every ten minutes from midnight to 4:00 AM. The task is operational as soon as it is created by the Scheduler service, using the default settings.</li> </ul>

Job	Information
IndexMaintenance	<ul> <li>Description: This job maintains the indexes on your Axiom databases (system database and audit database). It reports the current index fragmentation and then reorganizes or rebuilds the indexes as needed, depending on the current level of fragmentation. It also updates table statistics.</li> </ul>
	<ul> <li>Configuration: Any changes made to the SQL statements in this system job should only be made with the guidance of Kaufman Hall Software Support and/or your SQL database administrator.</li> </ul>
	<ul> <li>Schedule: By default, this task is set to run once per day, at 5:15 AM.         The task is operational as soon as it is created by the Scheduler service, using the default settings.     </li> </ul>

### Email notifications for system jobs

By default, most system jobs are configured to send notifications on error, however, you must edit the job to specify a valid email address for the notifications (such as the email address of a system administrator). In the **Notifications** section for the job, edit the email notification settings as desired. The default setting of {CurrentUser.EmailAddress} cannot be used because the job is run by the system and therefore does not resolve to a user email address.

**NOTE:** It is recommended to leave the SMTPMessageDelivery job at the default notification behavior of None. If this job experiences an issue attempting to send email, it likely will be unable to send you an email notification about this error.

For more information on notification settings, see Job properties.



# Scheduler Job Setup

To perform Axiom Software tasks using Scheduler, you must create jobs. Each job can execute one or more tasks. This section discusses how to set up jobs, including how to schedule jobs for future execution and how to be notified when a job has been completed.

# Managing jobs and tasks

Using the **Axiom Scheduler** dialog, administrators can create and edit Scheduler jobs. To access this dialog:

• On the Axiom tab, in the Administration group, click Manage > Scheduler.

This section discusses how to create, edit, and delete jobs and tasks, not how to manage the Scheduler queue once jobs have been placed on the schedule. If you need to stop or reschedule a scheduled job, see Managing scheduled jobs.

Scheduler jobs are saved as XML files and are stored in the Axiom Software file system at \Axiom\Scheduler Jobs Library.

## Creating a Scheduler job

You can create a new Scheduler job to perform one or more tasks.

#### To create a new job:

1. In the Scheduler dialog, on the Job tab, click New.

A new tab appears in the navigation pane, labeled **New Job**. The left-hand side of the job lists sections for which you can define various job settings. When you click a section name, the settings for that section display in the right-hand side of the job.

2. In the General section, define general job settings as desired.

For detailed information on the available settings for a job, see Job properties.

3. In the Scheduling Rules section, specify scheduling details for the job.

You can schedule the job for future execution, for one time or on a recurring basis.

**NOTE:** If you are always going to run the job manually, and do not need to schedule it for future execution, then you do not need to define scheduling rules.

For more information, see Defining scheduling rules for a job.

4. In the **Notification** section, specify email notification options for the job.

You can send email notifications every time the job completes, or only when the job experiences errors. By default, the job is configured to notify on completion.

For more information, see Setting up email notification for jobs.

- 5. In the **Tasks** section, add one or more tasks to the job.
  - a. On the **Job** tab of the ribbon, in the **Tasks** group, click **Add**. This brings up a list of available tasks. Select the task that you want to add.

The task is added to the **Tasks** section, and the settings for the task display in the right-hand side of the job.

b. Complete the settings for the task as desired.

The Task Control section of the task contains standard task settings, and the Task Details section contains settings unique to the task type. For more information, see Task Control properties.

If a required setting is not completed, the setting is highlighted in red and error text appears in the bottom of the dialog. Make sure to complete all required settings for the task before saving.

Repeat this process until you have added all desired tasks to the job. Tasks are processed in the order listed. If you need to change task order, select a task and then click **Move Up** or **Move Down**.

- 6. In the Job tab of the ribbon, click Save.
- 7. At the bottom of the Save As dialog, in the File name box, type a name for the job, and then click OK.

The job is saved as an XML file in the Scheduler Jobs Library.

If the job was saved with an active scheduling rule, Axiom Software determines the next scheduled date of execution and schedules the job. You can view the job in the Scheduled Jobs list (on the Service tab of the ribbon, click Scheduled Jobs).

#### Advanced job settings

This procedure covers the basic steps of creating a job. Jobs also support the following advanced options:

- Event handlers: You can create event handlers for the purposes of running the job using the RunEvent function. This allows users to trigger job execution from within an Axiom file.
- **Job variables**: You can create job variables and then use those variables within certain job settings. You can then dynamically pass in values for those variables when using the RunEvent function to execute the job.

For more information, see Creating event handlers for a job, Using job variables, and Using RunEvent to execute a job.

### Editing a job

You can edit a job at any time to change job settings, add or remove tasks, change scheduling rules, or change notification options.

This section describes the general process of opening a job for editing. For more details on the impacts of editing scheduling rules, see Defining scheduling rules for a job.

#### To edit a job:

1. In the Scheduler dialog, in the Job tab, click Open.

The Axiom Explorer dialog opens, showing the Scheduler Jobs Library only.

2. Select the job and then click Open.

The job opens in the **Scheduler** dialog. Make sure the job is the active tab in the navigation pane (the most recently opened tab is the active tab by default).

3. Edit the job and task properties as desired.

For detailed information on the available settings for a job, see Job properties. For detailed information on task settings, see Task Control properties.

4. In the Job tab of the ribbon, click Save.

#### Deleting a job

Deleting a job removes any scheduled executions of the job from the scheduled jobs list.

#### To delete a job:

1. In the Scheduler dialog, in the Job tab, click Open.

The Axiom Explorer dialog opens, showing the Scheduler Jobs Library only.

2. Navigate to the job that you want to delete, then right-click the job and select **Delete**.

You can also delete Scheduler jobs from Axiom Explorer or the Explorer task pane.

# Defining scheduling rules for a job

Once a job has been created, you can run it on demand, or you can schedule it for future execution. Jobs can be scheduled to be run one time, or on a recurring basis. To schedule a job, you define scheduling rules for the job.

You can add, edit, and remove the scheduling rules for a job at any time using the **Scheduling Rules** section of the job properties. You can also flag a rule as active or inactive. If a job has no scheduling rules, or if all of its scheduling rules are inactive, then it will not be run unless it is run manually by a user.

If a job is saved with an active scheduling rule, then Axiom Software determines the next scheduled instance of the job and places it in the scheduled jobs list. Once that instance has been processed, the next scheduled instance is determined and scheduled, and so on. Each time the job is run using an active scheduling rule, it is run as the current job owner (unless the job is a system job, in which case it is run as the Scheduler Service System identity).

If a job has multiple active scheduling rules, Axiom Software evaluates all of the rules and schedules a single instance of the job, for the earliest time allowed by the rules. Multiple scheduling rules do not result in multiple scheduled instances of the job.

**NOTE:** If a time zone is listed on the Scheduling Rules section of the job, then the defined rules will be evaluated in the context of that listed time zone. Otherwise, scheduling rules are evaluated in the context of the local time zone.

## Adding a Scheduling rule

You can add a scheduling rule to a job to schedule it for future execution, either one time or on a recurring basis.

If you only plan to run the job manually on demand, then you do not need to create a scheduling rule.

To add a scheduling rule to a job:

- 1. In the **Scheduler** dialog, open a job to edit or create a new job.
- 2. In the left-hand side of the job, select **Scheduling Rules**.
  - By default, this area is empty. You must add a rule in order to define scheduling for the job.
- 3. On the Job tab of the Scheduler ribbon, in the Scheduling Rules section, click Add.
  - A new row appears in the right-hand side of the job. By default, the new row is active, but does not have start / end dates or any specific recurrence settings.
- 4. Complete the following settings within the row as needed:

Item	Description
Active	If you want the job to be placed on the schedule as soon as you save the job with the new scheduling rule, then you should leave this option checked.
	However, if you just want to save your schedule settings but you are not ready to begin scheduling the job, then you can clear the <b>Active</b> check box for the rule. The job will not be scheduled until it is saved with an active scheduling rule.
Starting On Ending On	Optional. These dates specify the time frame for the scheduling rule. The starting date defines the earliest point in time that the job can be scheduled, and the ending date defines the latest point in time that the job can be scheduled.
	If these dates are not defined (left blank), then the job will be perpetually scheduled according to the rule settings, as long as the rule is active.
	If you want to schedule a one-time job, then set the starting / ending dates to the same date and time.
	NOTE: Your system locale determines the format of dates.
Day of Week	Specify the day(s) of the week that you want the job to be run:
	$\bullet \ \ \star$ (Default): The job will be run on all days within the start / end range.
	<ul> <li>0-6: The job will be run on the specified day or days, where 0 is Sunday and 6 is Saturday. Use a comma or a hyphen to separate multiple days (hyphen if the days are contiguous, commas if not).</li> </ul>
	For example, you can enter 1, 3, 5 for Monday, Wednesday, and Friday, or enter 1–5 for Monday through Friday.
Hours	Specify the time of day (hours) that you want the job to be run, in relation to the specified days:
	<ul> <li>* (Default): The job will be run on all hours.</li> </ul>
	<ul> <li>0-23: The job will be run on the specified hour or hours, where 0 is midnight and 23 is 11:00 PM. Use a comma or a hyphen to separate multiple hours (hyphen if the hours are contiguous, commas if not).</li> </ul>
	For example, you can enter 0, 12 to run at midnight and noon, or enter 0–12 to run every hour from midnight to noon.

Item	Description
Minutes	Specify the time of day (minutes) that you want the job to be run, in relation to the specified hours:
	<ul> <li>* (Default): The job will be run on all minutes (essentially the job is run continuously, once per minute).</li> </ul>
	<ul> <li>0-59: The job will be run on the specified minute or minutes of the hour, where 0 is the first minute of the hour and 59 is the last minute of the hour. Use a comma or a hyphen to separate multiple minutes (hyphen if the hours are contiguous, commas if not).</li> </ul>
	For example, you can enter $0,30$ to run at the top of the hour and the half hour, or enter $0-30$ to run every minute from the top of the hour to the half hour.
	<b>NOTE:</b> If you specify an hour, then in most cases you should also specify a minute (such as 0 to run the job at the top of the specified hour). If you enter an hour but leave the minutes at the default asterisk, then the job will run every minute in that hour.

If the Active check box for the rule is selected when the job is saved, then Axiom Software will calculate the date and time of the first scheduled execution and will place the job on the schedule.

## Editing a scheduling rule

You can edit a scheduling rule at any time, to toggle between active and inactive, and to change the start / end dates and recurrence settings.

#### **NOTES:**

- If a pending instance of this job is currently on the schedule, and you edit the scheduling rule, the pending instance will be updated to match the new schedule.
- If you inactivate a scheduling rule, any currently scheduled instances of the job will be automatically removed from the schedule.

#### To edit a scheduling rule:

- 1. In the **Scheduler** dialog, open a job to edit or create a new one.
- 2. In the left-hand pane of the job, select **Scheduling Rules**.
  - The defined rules display in the right-hand pane of the job.
- 3. Make any desired changes directly within the scheduling rules grid.

### Deleting a scheduling rule

You can delete a scheduling rule at any time. If a job has no active scheduling rules, it will not be processed unless it is manually run.

To delete a scheduling rule:

- 1. In the Scheduler dialog, open a job to edit.
- 2. Select the Scheduling Rules section of the job, and then select the rule that you want to delete.
- On the Job tab of the Scheduler ribbon, in the Scheduling Rules group, click Remove Selected.
   Alternatively, if you want to delete all scheduling rules for the job, click Clear All.

Any jobs in the scheduled jobs queue that were related to the deleted rule(s) are also deleted.

### Scheduling rule examples

The following are some example schedules and the rules used to achieve them:

Schedule	Start/End	Day of Week	Hours	Minutes
Weekdays at 11:00 PM	<optional></optional>	1,2,3,4,5	23	0
Every 15 minutes	<optional></optional>	*	*	0,15,30,45
Mondays at 11:30 PM	<optional></optional>	1	23	30
One time (6/30/2018)	Start: 06/30/2018 00:00	*	13	30
at 1:30 PM	End: 06/30/2018 00:00			
Every Wednesday in	Start: 07/01/2018 00:00	3	12	0
July at noon	End: 08/01/2018 00:00			
Continuous	<optional></optional>	*	*	*

# Setting up email notification for jobs

Scheduler can be configured to send an email notification when a job completes, or when a job has errors. In order for an email to be sent for a particular job, the following must be set up:

- The job must be configured to send a notification on completion or error. The notification settings must include valid To and From email addresses (or use system variables).
- The SMTP Message Delivery system job must be configured with a valid SMTP server for your environment. For more information, see Configuring Scheduler system jobs.

When a job creates an email notification, the notification is stored in the database and waits for the SMTP Message Delivery task to pick it up and send it.

**NOTE:** By default, all new Scheduler jobs are configured to send notification on completion, to the user who created the job. You only need to edit these settings if you want to change the default settings.

To configure a job to send email notifications:

- 1. In the Scheduler dialog, open a job to edit or create a new one.
- 2. In the left-hand side of the job, select Notification.
- 3. In the Job Notification Level section, select one of the following:
  - **Send all email notifications**: (Default) An email notification is always sent when the job is executed, regardless of the job status (success, failure, aborted, etc.).
  - Send email notification only when the job has errors: An email notification is only sent if the job experiences errors. If the job completes successfully with no errors, no email notification is sent.
  - **None**: No email notifications are sent for this job. The only way to check the status of the job execution is to check the job history.
- 4. In the Notification Message Content section, complete the following for the notification email:

Item	Description
То	The email address(es) to receive the notification email. Separate multiple addresses with a semicolon.
	By default, this is set to notify the user who executed the job, using the system variable {CurrentUser.EmailAddress}.
From	The email address that the message is sent from. This can be something like axiomscheduler@company.com, so that the recipient can easily tell that the message has been generated by Scheduler.
	By default, this is set to the Scheduler "from" email address as defined in the system configuration settings, using the system variable {Scheduler.FromEmailAddress}.
	<b>NOTE:</b> For installations that are using subsystems, the system variable {Scheduler.FromEmailAddress} may resolve to a subsystem administrator email address instead of the Scheduler "from" email address. For more information, see Using job variables.
Subject	The subject of the message. By default, this is set to "Axiom Scheduler Notification."
User Message	Optional body text for the email. This text is included in addition to the Scheduler auto-generated text regarding the job status.

You can use job variables in the notification settings. For more information, see Using job variables.

When this job is executed, it will generate an email notification according to the defined settings, and save that notification to the database to await delivery.

# Creating event handlers for a job

You can create user-defined event handlers in a job, for the purposes of automatically triggering the job for execution when the event name is called by another feature. Axiom Software supports several features that can be used to raise an event:

- The RunEvent function and command
- · File Group triggers
- The Raise Event Scheduler task

Event handlers are defined by name. Multiple jobs can have an event handler with the same name. When that event handler is called, it will affect all jobs that contain the event handler with the matching name.

To create an event handler in a job:

- 1. In the Scheduler dialog, open a job to edit or create a new job.
- 2. In the left-hand side of the job, select Event Handlers.
- 3. On the Job tab of the ribbon, in the Event Handlers group, click Add.
  - A new event handler is added to the job.
- 4. Double-click the **Event Name** field so that the field becomes editable, and then type the desired event handler name.
  - For example, if the event handler will be used to trigger Process Plan Files jobs, you might name the event handler ProcessPlanFiles.
  - This event name is the name that will be used in features such as RunEvent to trigger this job for execution.
- 5. In the Execute As field, select one of the following to determine the user identity that will be used to run the job when it is executed via the event handler:
  - Owner: The job will be run under the identity of the job owner.
    - The job owner is the user who last saved the job. If you are not sure who the current job owner is, you can check the Job Variables tab. The current job owner is listed in the System defined values section.
  - **Requester**: The job will be run under the identity of the user who triggered the event handler.

By default, event handlers are set to run as the **Owner**. You should carefully consider this option as it may affect whether the job can be run and how the job is run.

For example, if the event handler is set to **Requester**, but the user who triggered the job does not have access to the file group specified for a Process Plan Files task, then the task will fail.

This may be the desired outcome—you may want the job to be dependent on the user's rights, and therefore you should specify **Requester**. On the other hand, you may want the job to run in the same way every time, regardless of the user that triggers the job. In that case you should specify **Owner**.

By default, the event handler is set to **Active**, which means it will be found by any process that triggers the event handler. If you want to temporarily exclude this job from event handler processing, you can clear the Active check box.

### Deleting an event handler

If you no longer need an event handler, you can delete it from the job. Select the event handler and then click **Remove Selected**. You can also **Clear All** to remove all event handlers from the job.

User-defined event handlers display along with the system event handlers in the Event Handlers tab (Service > Event Handlers). If you right-click a user-defined event handler in this location and select Remove event handler, it does not delete the event handler from any jobs that use it, but it does set the event handler to inactive.

# Using job variables

You can use job variables within a Scheduler job, to define the value of the variable when the job is run. Job variables are managed in the **Job Variables** section of the job.

There are two types of variables:

- **User-defined variables:** You can create a variable and then use it within any job or task setting that supports variables, but only within that particular job. The primary use for user-defined variables is to run a job via RunEvent (either the function or the command), and pass in a variable value at that time.
- **System variables:** Axiom Software provides a number of system variables that can be used within relevant job and task settings. For example, instead of specifying a "hard-coded" email address for the job notification, you can use a system variable to specify that whoever ran the job should receive the notification.

In all cases, to use a variable within a job or task, enter the variable name into the desired setting, enclosed in curly brackets. For example: {variable}

**TIP:** If you want to use a variable in a job, you can right-click the variable and then select **Copy variable name to clipboard**. Navigate to the setting where you want to use the variable, and then paste it into the setting (the curly brackets are added automatically).

At the job level, variables can be used in any of the **Notification Message Content** settings in the **Notification** tab. At the task level, in general, variables can be used in any task field that accepts typed user input.

When the job is run, the variable values used for the job display in the job results under **Job Values**, and also in the email notification (if applicable).

#### User-defined variables

User-defined variables are created in the **Job Variables** tab. You define the name of the variable (without brackets), and if desired, define a default value for the variable.

When the job is run, the user-defined variable will be replaced with a value as follows:

- If the job was scheduled using RunEvent (function or command), and RunEvent sent a name / value pair that matches the name of the job variable, that value is used.
- If the job was scheduled as a result of a file group trigger, and the trigger has a defined variable that matches the name of the job variable, that value is used. Multiple values are returned as a comma-separated list.
- If the job contains a Process Document List task or a Process Plan Files task with a defined postprocessing variable that matches the name of a job variable, that value is used after that task has been processed.
- Otherwise, the default value defined in the Job Variables tab is used.

If the value is blank, then the job or task setting using the variable will be evaluated as blank. If the setting cannot be blank, then an error will result when the job is executed.

#### System variables

The available system variables are listed at the bottom of the **Job Variables** tab. Most of these variables relate to user names and addresses, for use within the job notification settings.

When the job is run, the system variable is replaced with the applicable system value.

The following values are available:

Variable	Description
{CurrentUser.EmailAddress} {CurrentUser.LoginName}	Returns the current user's email address, login name, or full name.
{CurrentUser.FullName}	The current user is the user identity under which the job is currently being run. Generally, this is the user who executed the job. If the job was executed via an event handler and the event handler is set to owner, then the current user will be the job owner.
{JobOwner.EmailAddress} {JobOwner.LoginName}	Returns the job owner's email address, login name, or full name.
{JobOwner.FullName}	The job owner is the user who last saved the job.
{Scheduler.ConfiguredFromEmailAddress}	Returns the system's default "from" address, as defined in the system configuration settings.
{Scheduler.FromEmailAddress}	<ul> <li>This returns a value as follows:</li> <li>If the current user belongs to a subsystem, this returns the subsystem administrator's email address.</li> <li>If the current user does not belong to a subsystem, this returns the default configured "from" address.</li> </ul>
{CurrentSubsystem.AdminEmailAddress}	Returns the email address of the subsystem administrator for the subsystem that the current user belongs to.
	<ul> <li>If the subsystem has multiple administrators, the email is sent to the first administrator.</li> <li>If the user belongs to multiple subsystems, the first returned subsystem for the user will be used. No specific logic is applied to determine the "correct" subsystem for any particular job.</li> <li>If the user does not belong to a subsystem, then no email address is returned.</li> </ul>

Variable	Description
{EventHandler.EventName}	Returns the name of the event handler that caused the job to be scheduled, if applicable. Otherwise the variable returns blank.
{NotificationAddress}	Returns the notification address defined for the plan codes that triggered a Scheduler job.
	This variable only applies when the job is executed as a result of a file group trigger, and only if the optional notification address settings are defined for the file group.  Otherwise, no email address is returned.
{Task.CurrentIterationValue}	Returns the current iteration value and the
{Task.IterationNumber}	current iteration number. These variables only apply when using the <b>Iteration</b> feature for a task.
	For more information, see Using iterative task processing.

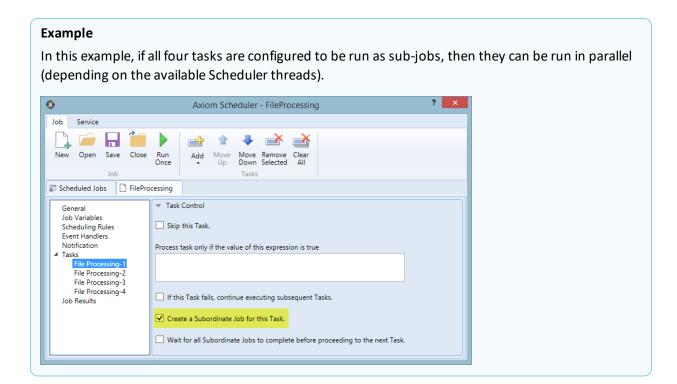
# Processing tasks in parallel

Each Scheduler job can have multiple tasks. By default, each task in the job is processed sequentially, in the order that the tasks are listed in the job.

If desired, you can configure tasks so that they are processed concurrently (in parallel) instead of sequentially. If appropriate, this may speed up the processing of the job.

In order to process tasks in parallel, the tasks must be configured to run as subordinate jobs (sub-jobs). To do this, edit the following settings in the **Task Control** section for each task:

- Select Create a Subordinate Job for this Task.
- Ensure that the following setting is *not* selected: Wait for all Subordinate Jobs to complete before proceeding to the next Task.



When a task is configured as described above, a sub-job is created for the task. The sub-job joins the Scheduler queue and is eligible for processing according to the normal Scheduler processing rules. For more information, see Processing priority for scheduled jobs.

For example, imagine that you have a job with four tasks, and these tasks are not dependent on each other. If you use the default settings, Scheduler takes the first task in the list and starts processing. The second task is not started until the first task is complete, and so on.

If instead you configure each task as a sub-job, then when the "parent" job is processed, it will create four sub-jobs. If two Scheduler threads are available for processing, then two of the sub-jobs are processed at the same time. If four Scheduler threads are available, then all four sub-jobs are processed at the same time. Once all of the sub-jobs are complete, the parent job is completed, and its status reflects the overall status of all of the sub-jobs.

If tasks are dependent on each other, then you should not process them as sub-jobs, or you should use the **Wait** setting as appropriate. For example, imagine that the first four tasks in the job can be run in any order, but the fifth task must be processed last. In that case, you can configure the first four tasks to run as sub-jobs, but on the fourth task you must enable **Wait for all Subordinate Jobs to complete before proceeding to the next Task**. This will cause Scheduler to wait for all sub-jobs to finish before it proceeds to the fifth, final task.

#### Example In this example, the file processing tasks are configured as sub-jobs so that they can be run in parallel. The last file processing task is configured to wait, so that all of the file processing tasks will be finished before the file collection task begins. Axiom Scheduler - FileProcessing Job Service Open Save Close Scheduled Jobs | FileProcessing ▼ Task Control General Job Variables Skip this Task. Scheduling Rules Event Handlers Notification Process task only if the value of this expression is true File Processing-1 File Processing-2 File Processing-3 File Processing-4 If this Task fails, continue executing subsequent Tasks Collect Worksheets Job Results ✓ Create a Subordinate Job for this Task. ✓ Wait for all Subordinate Jobs to complete before proceeding to the next Task.

# Using iterative task processing

You can configure a Scheduler task to use iterative processing, so that the task is repeated multiple times using a designated list of values. Each iteration of the task uses a different item in the list, until all items have been processed.

When you enable iterative processing for a task, you define the list of values by specifying a table column and an optional filter. The task will then be processed for each unique item in the table column. You can reference the column values within the task properties by using a built-in Scheduler job variable. As each iteration of the task is processed, the variable is replaced with the column value for the current iteration. Using this approach, the task can dynamically change for each iteration.

For example, you may have an import that you want to perform for four different entities in your organization. The import configuration is exactly the same except that the source file or query is different for each entity. If the import uses entity as a variable, then you can set up a single import task and configure it to iterate over the list of entities. Each iteration uses a different entity name or code, which you can pass into the import variables so that the import uses the correct source file or query for the current entity.

### Enabling iterative processing

Iterative processing is enabled in the Task Control properties of the task. Select the task within the Scheduler job, then click **Task Control** to expand that section. Any task can use iterative processing,

though it is more useful for certain task types such as Import ETL Package.

Complete the following properties in the Iteration section of the Task Control properties.

Item	Description
Iterate this Task	Specifies whether iterative processing is enabled for the task. If enabled, then the task will be performed N times, where N is the number of unique items in the specified iteration column. Job variables can be used to apply the current iteration value and iteration number to the task.
Create a Subordinate Job for each iteration	Specifies whether each iteration is processed as a separate subordinate job. By default, this is disabled, which means that all iterations are processed sequentially within the overall subordinate job created to process the iterations.
	If enabled, then each iteration is processed as a separate subordinate job, enabling concurrent execution of multiple iterations. This option should only be enabled if the order of iteration processing is not important.
Column	The column that contains the values to iterate over. Use Table.Column syntax to specify the column. Multiple-level lookups can be used.
	For example, if you specify Dept.Region, then the task will be processed once for each unique region value in the column (after applying any filter to limit the list of values).
Group By	Optional. By default, the group by column is the same as the iteration column, so that the task is processed once for each unique value in the iteration column. However, if needed, you can specify a different grouping level.
	You can use any column or columns that would be valid as the "sum by" level for an Axiom query, where the primary table is the table specified for the iteration column.
Order By	Optional. By default, the values are sorted based on the iteration column, in ascending order. You can specify a different sort column, or use the same sort column but change the order to descending.
	The sort order is ascending unless the keyword <code>desc</code> is used to specify descending order. For example:
	Dept.Dept desc
Filter	Optional. A filter criteria statement to limit the list of values for the iterative processing. You can use any filter that is valid against the source table (the table of the iteration column).

When iterative processing is enabled for a task, the iterations are always processed within a subordinate job. Therefore, enabling the Task Control option of Create a Subordinate Job for this Task is unnecessary.

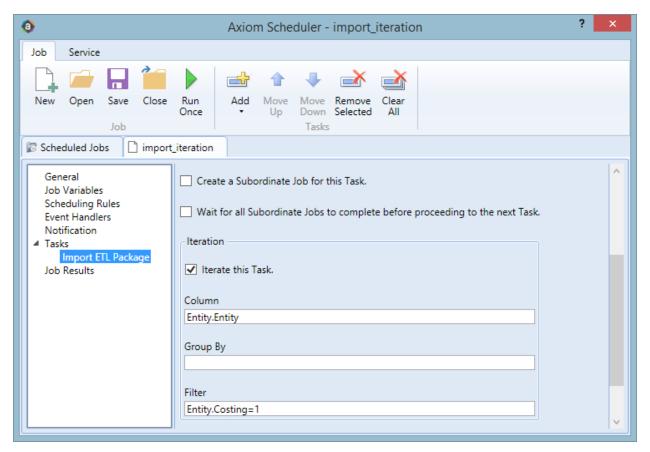
If your job has multiple tasks, and you want the tasks after the iterative task to wait for all iterations to complete before executing, then you must enable the following Task Control option for the iterative task: Wait for all Subordinate Jobs to complete before proceeding to the next Task.

### Configuring the task to change for each iteration

In order for the Scheduler task to apply the current iteration value to each iteration, you must use the built-in iteration variables within the task. These variables are job variables, and can be used like any other job variable. The following variables are available:

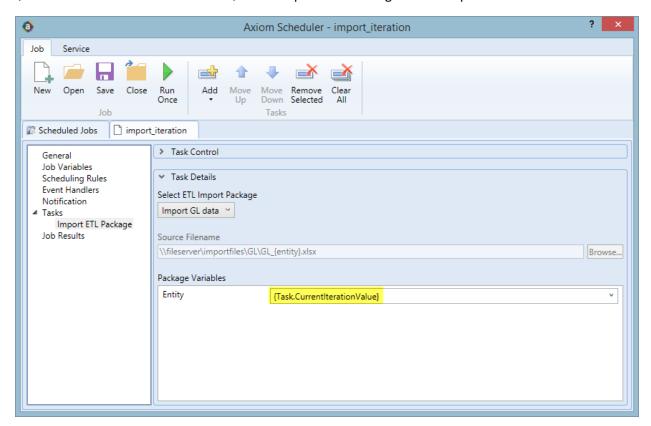
Variable	Description
{Task.CurrentIterationValue}	Returns the current value from the iteration list.
{Task.IterationNumber}	Returns the number of the current iteration.

To continue the previous example, imagine that you are setting up an import for iterative processing by entity. To define the list of entities, you set up the Iteration settings in the Task Control section like the following:



This example will iterate over the list of entities in the Entities column, limited to only those entities where the Costing column is set to True. If this resolves to 4 entities, then the task will be processed 4 times, once for each entity.

The import is configured with a variable {Entity}, which it uses to process the correct entity source file. In order to pass the current task iteration value to the import variable, you can use the job variable {Task.CurrentIterationValue} in the import task settings. For example:



When the first iteration is performed, the {Task.CurrentIterationValue} will be resolved as Entity\_1, so the import will be processed using Entity\_1 as the value for the {Entity} import variable. For the second iteration, the value Entity\_2 will be used, and so on. Using this approach, the import will be processed for all entities in the iteration column.

# Using iterative task processing

You can configure a Scheduler task to use iterative processing, so that the task is repeated multiple times using a designated list of values. Each iteration of the task uses a different item in the list, until all items have been processed.

When you enable iterative processing for a task, you define the list of values by specifying a table column and an optional filter. The task will then be processed for each unique item in the table column. You can reference the column values within the task properties by using a built-in Scheduler job variable. As each

iteration of the task is processed, the variable is replaced with the column value for the current iteration. Using this approach, the task can dynamically change for each iteration.

For example, you may have an import that you want to perform for four different entities in your organization. The import configuration is exactly the same except that the source file or query is different for each entity. If the import uses entity as a variable, then you can set up a single import task and configure it to iterate over the list of entities. Each iteration uses a different entity name or code, which you can pass into the import variables so that the import uses the correct source file or query for the current entity.

### Enabling iterative processing

Iterative processing is enabled in the Task Control properties of the task. Select the task within the Scheduler job, then click **Task Control** to expand that section. Any task can use iterative processing, though it is more useful for certain task types such as Import ETL Package.

Complete the following properties in the Iteration section of the Task Control properties.

Item	Description
Iterate this Task	Specifies whether iterative processing is enabled for the task. If enabled, then the task will be performed N times, where N is the number of unique items in the specified iteration column. Job variables can be used to apply the current iteration value and iteration number to the task.
Create a Subordinate Job for each iteration	Specifies whether each iteration is processed as a separate subordinate job. By default, this is disabled, which means that all iterations are processed sequentially within the overall subordinate job created to process the iterations.
	If enabled, then each iteration is processed as a separate subordinate job, enabling concurrent execution of multiple iterations. This option should only be enabled if the order of iteration processing is not important.
Column	The column that contains the values to iterate over. Use Table.Column syntax to specify the column. Multiple-level lookups can be used.
	For example, if you specify <code>Dept.Region</code> , then the task will be processed once for each unique region value in the column (after applying any filter to limit the list of values).

Item	Description
Group By	Optional. By default, the group by column is the same as the iteration column, so that the task is processed once for each unique value in the iteration column. However, if needed, you can specify a different grouping level.
	You can use any column or columns that would be valid as the "sum by" level for an Axiom query, where the primary table is the table specified for the iteration column.
Order By	Optional. By default, the values are sorted based on the iteration column, in ascending order. You can specify a different sort column, or use the same sort column but change the order to descending.
	The sort order is ascending unless the keyword <code>desc</code> is used to specify descending order. For example:
	Dept.Dept desc
Filter	Optional. A filter criteria statement to limit the list of values for the iterative processing. You can use any filter that is valid against the source table (the table of the iteration column).

When iterative processing is enabled for a task, the iterations are always processed within a subordinate job. Therefore, enabling the Task Control option of Create a Subordinate Job for this Task is unnecessary.

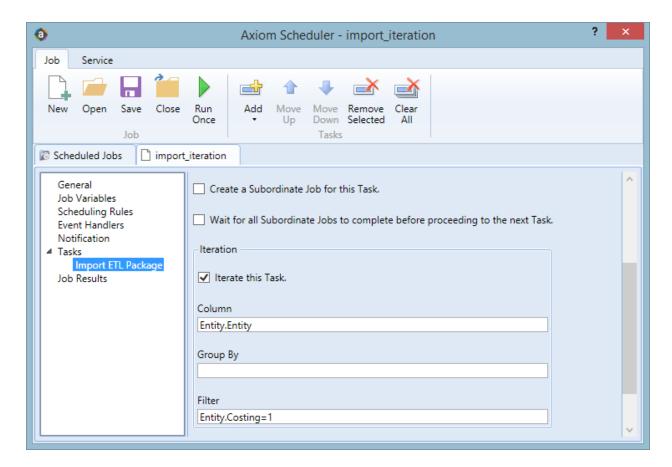
If your job has multiple tasks, and you want the tasks after the iterative task to wait for all iterations to complete before executing, then you must enable the following Task Control option for the iterative task: Wait for all Subordinate Jobs to complete before proceeding to the next Task.

### Configuring the task to change for each iteration

In order for the Scheduler task to apply the current iteration value to each iteration, you must use the built-in iteration variables within the task. These variables are job variables, and can be used like any other job variable. The following variables are available:

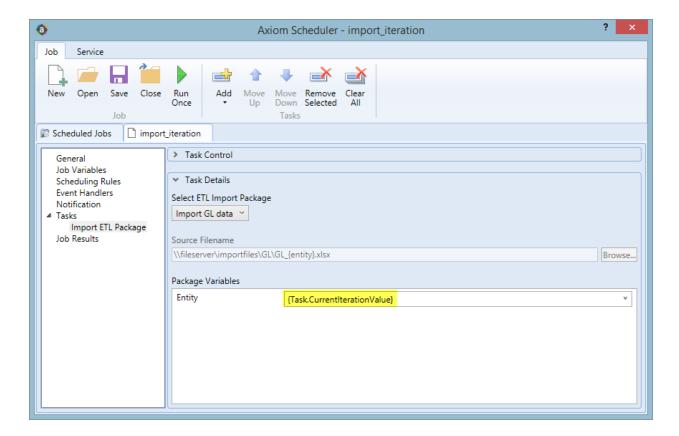
Variable	Description
{Task.CurrentIterationValue}	Returns the current value from the iteration list.
{Task.IterationNumber}	Returns the number of the current iteration.

To continue the previous example, imagine that you are setting up an import for iterative processing by entity. To define the list of entities, you set up the Iteration settings in the Task Control section like the following:



This example will iterate over the list of entities in the Entities column, limited to only those entities where the Costing column is set to True. If this resolves to 4 entities, then the task will be processed 4 times, once for each entity.

The import is configured with a variable {Entity}, which it uses to process the correct entity source file. In order to pass the current task iteration value to the import variable, you can use the job variable {Task.CurrentIterationValue} in the import task settings. For example:

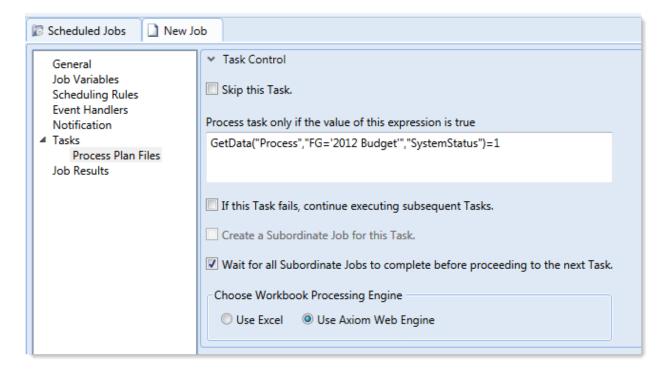


When the first iteration is performed, the {Task.CurrentIterationValue} will be resolved as Entity\_1, so the import will be processed using Entity\_1 as the value for the {Entity} import variable. For the second iteration, the value Entity\_2 will be used, and so on. Using this approach, the import will be processed for all entities in the iteration column.

# Conditionally processing tasks in a job

You can configure a task so that it is only processed if a particular condition is met. This feature is configured in the task settings, in the Task Control section, under Process task only if the value of this expression is true.

To enable conditional processing, you must specify a logical expression that will resolve to either true or false when the job is executed. If true, then the task is processed as normal. If false, then the task is skipped.



The logical expression is evaluated using an IF function on the Scheduler server as follows:

```
=IF(Expression, 1, 0)
```

You can enter any expression that would be valid in an IF function. You can use Excel functions, Axiom functions, and Scheduler job variables in the expression. If you use a job variable, it must be placed in quotation marks unless you expect the variable value to be resolved and evaluated as a number.

If the task is not processed because the condition resolves to false, this is not considered a failed task. If there are other tasks in the job, they will be processed. If you want an entire job to be conditional, you can do either of the following:

- Repeat the condition in each individual task settings. Keep in mind that the condition will be evaluated for each individual task, which means that if it is possible for the condition to change in between tasks, some tasks might be processed while others aren't.
- Use the condition on a Raise Event task that then triggers another job for processing. For more information, see Raise Event task.

### Examples

The following are some example expressions for conditional processing:

```
GetData("Process", "FG='2012 Budget'", "SystemStatus") =1
```

If this GetData function returns 1, the expression resolves to true and the task is processed. If not, it is false and the task is skipped.

```
AND("{EventHandler.EventName}"="ProcessPlanFiles", {Dept}=1000)
```

If this job was triggered for execution by the ProcessPlanFiles event handler, and if the job variable Dept resolves to 1000, then this expression is true and the task is processed. Note that in the first part of the expression, the event handler variable will return a string value so it must be placed in double quotation marks. In the second part of the expression, the department variable will return a number so it is not placed in quotation marks.

```
AND (Day (Now ()) \leq 7, Weekday (Now ()) = 2)
```

This expression will return true if it is the first Monday of the month, otherwise it will return false.

### Processing priority for scheduled jobs

Once a job reaches its start time, it is eligible to be processed by Scheduler, and joins the processing queue.

Scheduler has a configured number of threads that are used to process jobs. As a Scheduler thread becomes available, it takes the next job in the processing queue. The priority of jobs in the processing queue is determined by the combination of the job's priority category, and its **Priority Elevation** setting.

Each job has a priority category, based on how the job execution was initiated. The priority categories are as follows:

- 1. Manual: The job was executed manually.
- 2. Event Handler: The job was executed by a Scheduler event handler.
- 3. Scheduled Job: The scheduled instance of the job results from an active scheduling rule.
- 4. Subordinate Job: The job was generated as a subordinate job, from a currently executing job.

Manual jobs are highest priority and are processed first, and subordinate jobs are lowest priority and are processed last. Then within each category, jobs are processed according to their **Priority Elevation** setting.

For example, imagine that Scheduler has 2 available threads and the following jobs are eligible to be processed:

Priority Category	Priority Elevation
Manual	Default
Event Handler	Default
Scheduled	Default
Scheduled	Elevated
	Event Handler Scheduled

- Scheduler will first execute jobs A and B, because those are the highest priority jobs based on their priority category.
- When the next thread becomes available, Scheduler will execute job D. Although job C may have entered the queue first, and the two jobs have the same priority category, job D's priority elevation is set to **Elevated** so it takes precedence within the category. If instead both jobs were set to **Default**, then job C would be executed first if it entered the queue before job D.
- When the next thread becomes available, Scheduler will execute job C.

**NOTE:** If a job's **Priority Elevation** is set to **Interrupt**, then it is run as soon as it is eligible, regardless of its priority category and regardless of whether any Scheduler threads are currently available to process the job. If no Scheduler threads are available, a new one is created to process the job, even if this temporarily exceeds the number of configured threads for the server.

## Running a job

If a job is saved with an active scheduling rule, then the job is automatically placed on the schedule to be run according to that rule. Each time the job is run according to the rule, it is run as the current job owner (unless it is a system job, in which case it is run as the Scheduler Server System identity).

However, you can also choose to run a job manually. If you run a job manually, the job is added to the **Scheduled Jobs** list with a start time of now, to be processed in accordance with its job priority settings. The job will be run using your user identity (again, unless it is a system job).

Running a job manually does not impact any scheduled executions of the job as determined by scheduling rules. For example, if a job is scheduled to be run at 10:00 PM tonight, and you run the job manually at 2:00 PM, the job will still be run as scheduled at 10:00 PM, unless you edit or inactivate the scheduling rule.

#### To manually run a job:

- 1. In the Scheduler dialog, in the Job tab, click Open.
  - The Axiom Explorer dialog opens, showing the Scheduler Jobs Library only.
- 2. Select the job and then click **Open**.
  - The job opens in the **Scheduler** dialog. Make sure the job is the active tab in the navigation pane (the most recently opened tab is the active tab by default).
- 3. In the Job tab of the ribbon, click Run Once.

A confirmation message informs you that the job has been placed on the schedule.

**NOTE:** When you click **Run Once**, any unsaved changes to the job are automatically saved. This save will designate you as the job owner (if you are not already the job owner).

# Job properties

This topic is a reference for the settings that can be defined for a Scheduler job.

### General

This section defines general settings for the job.

Item	Description
Description	Optional. The description of the job.
	The job description can also be edited in Axiom Explorer, in the Scheduler Jobs Library.
Job Restart Behavior	Specifies whether and how the job should be restarted if it is interrupted prior to completion. Select one of the following:
	<ul> <li>Do not reschedule this job. In this case, you must manually reschedule the job if it needs to be run before its next scheduled execution.</li> </ul>
	<ul> <li>Restart the job from the first task. The entire job is run again, even if some of the tasks were completed successfully before the job was interrupted.</li> </ul>
	<ul> <li>Resume the job beginning with the first uncompleted task. (Default) The job resumes and only the uncompleted tasks are run.</li> </ul>
	A job would be interrupted if the Scheduler server processing it was restarted, or if the Scheduler service on the server was stopped or restarted, or if the Scheduler server was disabled from the Servers tab (Service > Servers) of the Scheduler dialog.
Job Results	Specifies whether historical job results are purged when the job is run.
Cleanup	To purge job results:
	<ol> <li>Select Purge historical job results whenever this job runs.</li> </ol>
	<ol> <li>In Number of days to keep results for this job, specify the number of days to keep when purging results. By default this is set to 0, which means all job results will be purged except the result for the current job execution.</li> </ol>
	A day is counted as 24 hours from the time the cleanup task is executed. So if you specify 1 day, and the task is run at 11:00 PM on Tuesday, then all results prior to 11:00 PM Monday are purged.
	If this option is not selected, then historical job results remain in the database until the system's <b>Purge System Data</b> task is run.

### Item Description **Priority Elevation** Specifies the priority of the job in the scheduled jobs queue, within the job's priority category. Select one of the following: • Default: (Default) This job is run on a "first come, first served" basis. The total number of jobs that can be run at one time is determined by the configured number of Scheduler threads for the installation. • Reduced: The job is designated as a low priority job, and remains at the bottom of the queue until other jobs with Default and Elevated priority have been run. • Elevated: The job is designated as a high priority job, and is moved to the top of the queue to be run before Default and Reduced priority jobs. • Interrupt: The job is run immediately, regardless of any jobs currently waiting in the scheduled jobs queue, and regardless of whether any Scheduler threads are currently available to run it. If no Scheduler threads are currently available, a new thread is created, even if this exceeds the configured thread limit for the installation. Job execution order also depends on the priority category of a specific job execution. See Processing priority for scheduled jobs. Specifies whether the job is run as a system job. Only administrators can edit Mark as System Job this check box. If this check box is selected, the job is run under the "Scheduler Service" system identity instead of a user identity, and the job is run by the system Scheduler server which operates on the Axiom Application Server. Generally, this check box should only be selected for system "support" tasks that should not depend on individual user rights. This check box is not available if the job contains non-system tasks (generally, spreadsheet-related tasks). For more information, see System jobs.

### Item Description Put the system If this option is selected, then the system will be placed into administrator-only in 'admin only' mode at the start of the job, and then placed back into full access mode when mode during this all tasks are completed (including any sub-jobs). This is the same behavior as iob going to Manage > Security > System Access and selecting Administrators Only. **NOTES:** • You should make sure that any jobs using admin-only mode do not overlap. For example, imagine that job A starts and places the system in admin-only mode. While job A is still running, job B starts and finishes. If job B also uses admin-only mode, then when job B finishes the system will be placed back into full access mode, meaning the remainder of job A will be processed in full access mode. Any job using admin-only mode must be run by an administrator. Generally speaking, any job set to run using admin-only mode should be run at a time when no end users will be logged into the system and no other Scheduler jobs will be running.

#### Job Variables

This tab has two sections for job variables:

 In the Job values section at the top of the tab, you can manage user-defined variables for use in the current job.

To add or remove variables, use the Add, Remove Selected, or Clear All commands in the Job Variables group of the Job tab. This group is only available when you have selected the Job Variables section in the left-hand side of the job.

When creating user-defined variables, do not add curly brackets to the variable name. Curly brackets are only required when you use the variable in a job or task setting.

• In the **System defined values** section at the bottom of the tab, you can view the system variables available for use in the job.

You can right-click any variable in this section (user-defined or system-defined) and select **Copy variable** name to clipboard. You can then navigate to the setting where you want to use the variable, and then paste it. The variable will be pasted with the necessary curly brackets.

For more information, see Using job variables.

### Scheduling Rules

Each row in this section defines a scheduling rule for the job. Jobs will be automatically scheduled according to the settings in this section.

To add or remove scheduling rules, use the Add, Remove Selected, or Clear All commands in the Scheduling Rules group of the Job tab. This group is only available when you have selected the Scheduling Rules section in the left-hand side of the job.

For more information, see Defining scheduling rules for a job.

Item	Description
Active	Specifies whether the scheduling rule is active. If this check box is not selected, then the rule is ignored for purposes of scheduling the job.
Starting On	Optional. Specifies the earliest date and time for the scheduling rule to take effect.
	If you want the job to run one time only, set <b>Starting On</b> and <b>Ending On</b> dates to the same date/time.
Ending On	Optional. Specifies the expiration date and time for the scheduling rule. Once this date is past, no further executions will be scheduled for this rule.
Day of Week	Specifies the day(s) of the week that you want the job to be run:
	<ul> <li>* (Default): The job will be run on all days within the start / end range.</li> </ul>
	<ul> <li>0-6: The job will be run on the specified day or days, where 0 is Sunday and 6 is Saturday. Use a comma or a hyphen to separate multiple days (hyphen if the days are contiguous, commas if not).</li> </ul>
Hours	Specifies the time of day (hours) that you want the job to be run, in relation to the specified days:
	<ul> <li>* (Default): The job will be run on all hours.</li> </ul>
	<ul> <li>0-23: The job will be run on the specified hour or hours, where 0 is midnight and 23 is 11:00 PM. Use a comma or a hyphen to separate multiple hours (hyphen if the hours are contiguous, commas if not).</li> </ul>
Minutes	Specifies the time of day (minutes) that you want the job to be run, in relation to the specified hours:
	<ul> <li>* (Default): The job will be run on all minutes (essentially the job is run continuously, once per minute).</li> </ul>
	<ul> <li>0-59: The job will be run on the specified minute or minutes of the hour, where 0 is the first minute of the hour and 59 is the last minute of the hour. Use a comma or a hyphen to separate multiple minutes (hyphen if the hours are contiguous, commas if not).</li> </ul>

#### Event Handlers

If an event handler is associated with the job, it is listed here. There are two types of event handlers:

- System event handlers, for completing system-triggered tasks. See Managing event handlers.
- User-defined event handlers, for running jobs via RunEvent. See Creating event handlers for a job.

To add or remove event handlers, use the Add, Remove Selected, or Clear All commands in the Event Handlers group of the Job tab. This group is only available when you have selected the Event Handlers section in the left-hand side of the job.

Item	Description
Active	Specifies whether the event handler is active or not within the current job. If inactive, then actions that trigger the event handler will ignore this job.
Event Name	The name of the event handler.
	Multiple jobs can have an event handler with the same name; all those jobs will be affected when the event handler is triggered.
Execute As	<ul> <li>The user identity under which the job will be run when the event handler is triggered.</li> <li>Owner: For system-managed event handlers, the owner is the system Scheduler identity. For user-defined event handlers, the owner is the user who last saved the job.</li> <li>Requester: For all event handlers, the requester is the user who caused the event handler to be triggered.</li> </ul>

### Notification

This section defines email notification settings for the job. For more information, see Setting up email notification for jobs.

Job variables can be used in this section. For more information, see Using job variables.

Item	Description
Job Notification Level	Specifies when email notifications are sent for the job. Select one of the following:
	Send all email notifications (Default)
	<ul> <li>Send email notification only when the job has errors</li> </ul>
	None
	If anything other than <b>None</b> is selected, then you must specify <b>To</b> , <b>From</b> , and <b>Subject</b> .
То	The email address(es) to receive the notification email. Separate multiple addresses with a semicolon.
From	The email address to use as the "From" address for the notification email.
Subject	The subject text for the notification email.
User Message	Optional. The body text for the notification email.
	Text entered here will be appended to the body text generated by Scheduler.

#### Tasks

This section defines the tasks in the job. In the ribbon, task commands are available on the **Job** tab, in the **Tasks** group.

- To add a task, click Add.
- To change the order of tasks, select a task and then click Move Up or Move Down.
- To delete a task, select the task and then click Remove Selected.
- To delete all tasks, click Clear All.
- To copy a task, right-click the task and then click **Copy**. You can copy the task within the same job, or to another open job in the Scheduler window. Right-click any task (or the **Tasks** section header) and then select **Paste**. The job is pasted underneath the job you right-clicked (or at the end of the list if you right-clicked the **Tasks** section header).
- To rename a task, double-click the task name to make it editable, and then type the new name. For example, if you have a job with multiple File Processing tasks, then you may want to edit the name of each task so that you know which file each task relates to at a glance. (You can also right-click and select Rename.)

Tasks are processed in the order they are listed in the job. By default, when you add a new task to a job, it is placed at the bottom of the list. Make sure to move the new job if it should not be processed last.

Tasks can be processed concurrently instead of sequentially if they are configured to be run as a subordinate job within the parent job.

Each task type has its own unique settings in addition to the standard task settings. For more information, see Scheduler Task Reference.

#### Job Results

Displays historical results for the job. This section is blank if the job has never been run.

Job results may be purged periodically by using the **Job Results Cleanup** option for the job, or by the system **Purge System Data** task.

**NOTE:** Users with the **Scheduled Jobs User** security permission only see results for jobs that they executed. Administrators see results for all executions.

For more information on job results, see Viewing job results.



# Scheduler Administration

Administrators can review the job schedule and job results, as well as perform Scheduler administration activities such as managing system jobs and configuring Scheduler servers.

## Managing scheduled jobs

The **Scheduled Jobs** tab of the Scheduler dialog displays all jobs that are scheduled to be processed or are in process. This includes jobs that were manually executed, and jobs that were saved with an active scheduling rule (including system jobs).

By default, this list displays when you first open Scheduler. If you have changed the view in the dialog, you can return to it by clicking the following:

• On the Service tab, in the Service View group, click Scheduled Jobs.

If a job has a scheduling rule with a recurring schedule, only the first scheduled execution appears in the list. For example, if you have a job that is scheduled to run once a month for a year, you will not see all twelve scheduled executions in the list—you will only see the first scheduled execution. Once that instance has been run, the scheduling rule is re-evaluated and the next scheduled execution appears in the list.

### In this tab, you can:

- **Stop a scheduled execution.** If you do not want a scheduled instance to be run, right-click the item and select **Remove from Schedule**. This not only removes the scheduled instance, it also inactivates the associated scheduling rule for the job (if applicable). You will need to reactivate the scheduling rule if you want future scheduled instances of the job to proceed.
- Abort an in-process job. If a job is already running and you want to stop it, right-click the item and select Abort Working Job. This will immediately abort the job regardless of what it is currently doing. Keep in mind that depending on the type of job and its stage in the process, this may result in side-effects such as files remaining locked or incomplete processing. No notifications will be sent for the aborted job. Generally, this action should be reserved for situations where a job has gotten "stuck" or was started in error.
- **Reschedule an execution.** To change the scheduled start date/time of a scheduled instance, right-click the item and then select **Reschedule Start Time**. The **Start Time** field in the grid is now

editable, and you can type in a new date and/or time.

• Refresh the list. On the Service tab, in the Actions group, click Refresh. New scheduled jobs will be added to the list, and Status and Due In will be updated appropriately.

Once a job reaches its start time, it is eligible to be processed by Scheduler, and joins the processing queue. For more information on how Scheduler determines the order of jobs to process, see Processing priority for scheduled jobs.

**NOTE:** Users with the **Scheduled Jobs User** security permission can only modify jobs that they placed on the schedule. Jobs scheduled by other users are visible, but are grayed out and unavailable for editing. Administrators can view and edit all scheduled jobs.

### Scheduled job information

The following information displays for each scheduled job:

Item	Description
ID	The system-generated ID for the job execution. Each scheduled execution of a job has a unique ID. Job results are listed by execution ID.
Job	The job name. Each job name corresponds to either a user-defined job in the Scheduler Jobs Library, or a system-managed job name generated by an Axiom Software event handler.
User	The user identity under which the job is being run. System jobs are identified as "System."
Status	Job status is either <b>Pending</b> (waiting to be executed) or <b>Working</b> (currently being executed).
Server	If a job is currently <b>Working</b> , then the server executing the job is listed here. Otherwise, this column is blank.
Priority	The priority category for the job:
	1. Manual: The job was executed manually.
	2. Event Handler: The job was executed by a Scheduler event handler.
	<ol> <li>Scheduled Job: The scheduled instance of the job results from an active scheduling rule.</li> </ol>
	<ol> <li>Subordinate Job: The job was generated as a subordinate job, from a currently executing job.</li> </ol>
	The priority category determines how jobs are evaluated for processing order, in conjunction with the job's <b>Priority Elevation</b> setting. Manual jobs are highest priority, and subordinate jobs are lowest priority.

Item	Description
Start Time	The time the job is scheduled to start. If the job was manually executed or generated for a system job, the start time is the time the execution was initiated.
Due In	The length of time until the job starts. For example, if the job is scheduled to run at noon and it is currently 11:50 AM, then <b>Due In</b> is 10 minutes.

## Viewing job results

Once a job has been run, you can view the results in the Scheduler dialog. Job results can tell you:

- Whether the job processed successfully or failed
- The process steps performed by the job, if it was successful
- The error message for the job, if it failed
- The date/time when the job started processing and how long it took to process
- The Scheduler server that processed the job

**NOTE:** Users with the **Scheduled Jobs User** security permission can only see job results for jobs that they executed. Administrators can see job results for all jobs.

To view results for all jobs:

• On the Service tab, in the Service View group, click Job Results.

The **Result History** tab opens, listing a summary of results for all jobs that have started or finished. If you want to see more specific details for a particular job execution, double-click it. This opens the related job to the **Job Results** section, where you can view more details such as the specific error message for a failed execution.

You can also open a job directly, and go to the Job Results section to view results for that job only.

### System job results

By default, system job results are hidden in the **Result History** tab. System jobs such as the SMTP message delivery job run frequently, and can easily fill up the result history screen, making it difficult to find results for user-initiated jobs.

If you want to view results for system jobs, you can do one of the following:

- At the bottom of the Result History tab, clear the Hide system jobs check box.
- Open the system job directly, and view its job results within the job. For example, you can go to
  the Scheduled Jobs tab and double-click the System.SystemDataPurge job to view all results for
  that job.

### Purging job results

Over time, the job result history can get quite large, so it is recommended to purge the result history periodically. There are two ways to purge result history:

- Each job can be configured to purge its own prior result history when it is run (Job Results Cleanup). This setting is located in the General section of the job properties. For more information, see Job properties.
- The Purge System Data system job purges result history whenever it is run (by default, once per hour). For more information, see Configuring Scheduler system jobs.

When you purge job results, you specify a number of days of history to be kept. For example, you may always want to keep 5 days' worth of job history, and purge any results older than that.

## Managing Scheduler servers

On the Servers tab of the Scheduler dialog (Service > Servers), you can view the status of all Scheduler servers for the current system, enable or disable the servers, and change server configuration settings.

**NOTE**: Only administrators can manage Scheduler servers. Users with the **Scheduled Jobs User** permission cannot see the **Servers** tab in the Scheduler dialog.

This tab lists all Scheduler servers for this Axiom Software installation, including system Scheduler Services for on-premise systems and Cloud Integration Services for Cloud Service systems. For each server, the following properties are listed:

Item	Description
Server	The server name, including the following appended codes where appropriate:
	-System indicates the system Scheduler Service
	-CIS indicates a Cloud Integration Service
Version	The Axiom Software version installed on the server. This should match the version installed on the application server.
Service Status	The status of the service on the server, such as stopped or started.
Processor Status	The status of the server, such as running or stopped.
Processor Enabled	Whether the server is enabled for processing.
Has Embedded App Server	Whether the server uses an embedded application server. This is determined during the installation.

This tab can be useful in troubleshooting—for example, if scheduled jobs are not executing, you can check this tab to see if the Scheduler service has been stopped on a server or if the server is down.

### Enabling or disabling server processing

When a new Scheduler server is installed, you must manually enable it within the Scheduler dialog before it will begin processing jobs. Later, if you want to temporarily prevent a Scheduler server from processing jobs, you can disable processing for that server.

**NOTE:** The system Scheduler service for on-premise systems is enabled automatically. You do not need to manually enable this service.

To enable or disable server processing:

• Select the server in the list. In the Configuration details section for that server, select or clear the Schedule Processing Enabled check box, and then click Update.

If a server is disabled, it will not process any new tasks. If the server is currently processing a job when it is disabled, it will finish the current task and then stop processing. The job's settings determine whether the stopped job is rescheduled for processing.

### Configuring Scheduler server settings

Each Scheduler server has its own configuration settings that determine behavior such as how many threads are available to process jobs at any one time.

**NOTE:** We recommend leaving the default Scheduler server configuration settings unless Axiom Software Support has instructed you to change them, based on your particular system and server environment.

To view or edit the configuration settings for a server, select the server in the list. The settings for that server display in the **Configuration details** section. If you edit a setting, click **Update** to apply the change.

Item	Description
Max Worker Threads	The maximum number of threads that the Scheduler server has available to process jobs simultaneously. The default is 4.
	As a guideline, the Scheduler service can typically handle 2 threads per CPU on the Scheduler server. For example, a 4 CPU server could be set to 8 threads. Using a greater number of threads may degrade performance.
	NOTES:
	<ul> <li>This setting is inapplicable to the system Scheduler service. It creates threads as needed to process system jobs.</li> </ul>
	<ul> <li>A Scheduler server will temporarily exceed the configured number of threads if a job's priority elevation is set to Interrupt and no threads are currently available to process it.</li> </ul>

Item	Description
Server Poll Rate	The rate at which the Scheduler server checks for new jobs to process, in seconds. The default is 20 seconds.
Retry Rate	The rate at which the Scheduler server will resubmit commands to the application server, in the event the original connection failed. The default is 5 seconds.

### Removing server registration

If a Scheduler server is no longer valid, or if the Scheduler software was installed on a server in error, you can remove the server registration for Scheduler.

To remove the registration for a Scheduler server:

Right-click the server and then select Remove Server Registration.

The server is removed from the list. Registration can only be removed if no historical job results exist for the server.

If the service is restarted on the machine and the Scheduler installation is pointed to this Axiom Software installation, then the server will be re-registered.

## Managing event handlers

Event handlers are used to trigger Scheduler jobs based on an event. Scheduler has a number of built-in system-managed event handlers, to support certain features. Scheduler also supports user-defined event handlers.

In the Scheduler dialog, on the Event Handlers tab (Service > Event Handlers), you can view all active event handlers. This includes system-managed event handlers and user-defined event handlers. If an event handler in a job is inactive, it is not listed here.

#### **NOTES:**

- User-defined event handlers cannot be created in this tab; they can only be created within a Scheduler job. For more information, see Creating event handlers for a job.
- The Event Handlers tab is only visible to administrators. Non-admin users with the Scheduled
  Job Users permission can only manage event handlers within the jobs they have access to.

For each event handler, the following information is displayed.

- Event Name: The name of the event handler.
- **Job**: The job associated with the event handler. For system event handlers, the job is a system job. For user-defined event handlers, the job is the job where the event handler was created. Multiple jobs can contain a user-defined event handler with the same name (in which case the event handler is listed multiple times).
- **User**: The user identity that will be used to execute jobs triggered by the event handler. If the event handler is configured to run as the requester, then "(Requester)" is listed here. If the event handler is configured to run as the owner, then the owner name is listed here (either a specific user name, or "System").

If you want to change how the event handler runs, you can double-click the event handler name to open and edit the associated job. Go to the **Event Handlers** section and then edit the **Execute As** setting for the event handler.

### Removing event handlers

You can remove an event handler by right-clicking the event handler and selecting **Remove event** handler.

**IMPORTANT:** You should only remove a system event handler if the event handler has become corrupted and you need to restore a new copy of it. If you remove a system event handler and then do not restore a new copy, the associated system functionality will become unavailable.

For user-defined event handlers, removing the event handler inactivates it within the job; it does not delete the event handler. If you want to delete the event handler entirely, you must edit the job to delete it.

### Creating system jobs and system event handlers

If any system jobs or event handlers are accidentally or deliberately removed from the system, you can restore new copies by right-clicking any event handler and selecting **Create system jobs and event handlers**. Any missing event handlers or system jobs will be re-created. You must refresh the Event Handler tab to see the re-created event handlers.

### System event handlers

Axiom Software uses the following system event handlers.

Event Handler	Description	Job Name
RefreshPlanFiles	Creates a Process Plan Files job when a	PlanFileRefreshHandler
	user selects to run the <b>Process Plan Files</b> utility (from the file group menu) on the Axiom server instead of locally.	The job is performed as the user who executed the utility. The job is not processed on the system Scheduler server.
WorkflowInititation	Handles background processes when a	AxiomWorkflowEventHandler
	workflow is activated. Generates the initial set of workflow tasks. Supports the legacy workflow feature (is only present in systems where Workflow is enabled).	Operates as a system job.
WorkflowUpdate	Handles background processes for plan	AxiomWorkflowEventHandler
	file progression in a workflow. Moves the plan file to the appropriate stage and generates the new workflow tasks.  Supports the legacy workflow feature (is only present in systems where Workflow is enabled).	Operates as a system job.

## Managing remote data connections

Remote data connections provide a way for the Axiom Cloud Service to communicate with the client's local network environment. This connection can be used to support the following features:

- Importing data to the cloud from a local file or database
- Exporting data from the cloud to a local database.
- Authenticating users for access into Axiom Software (Windows Authentication)
- Saving output files for file processing to a local file system (snapshot or export)

For a full-process discussion of how to set up and use remote data connections with the Axiom Cloud Service, see the separate *Cloud Service Technical Guide*.

Remote data connections are managed in the **Remote Data Connections** screen of the **Scheduler** dialog. The following is required to create a remote data connection:

- The Axiom Cloud Integration Service must be already installed on a server within your local network environment, and the server must be enabled for use (on the Server screen of the Scheduler dialog).
- You must have the connection string to your system's namespace on the Axiom Cloud Service. If you do not have this information, contact Kaufman Hall Software Support.

**NOTE:** Only administrators can create and edit remote data connections. Users with the **Scheduled Jobs User** permission cannot see the **Remote Data Connections** tab in the Scheduler dialog.

#### To create a remote data connection:

- 1. On the Service tab of the Scheduler dialog, click Remote Data Connections.
- 2. At the bottom of the dialog, click New Connection.

A set of empty fields displays in the Configuration details section.

3. Complete the following settings:

Item	Description
Name	The name of the remote data connection. This is the name that you will select in import and export utilities to indicate that the utility should use the remote data connection.
Description	Optional. The description of the remote data connection.
Connection String (SAS)	The connection string to the namespace that was created by Kaufman Hall Software Support in the Microsoft Azure environment for your cloud system. If you do not have this connection string, contact Kaufman Hall Software Support.
Use for authentication service	Specifies whether the remote data connection is used for Windows Authentication. This only applies if Windows Authentication is enabled for your cloud system.
	If enabled, this remote data connection will also be used for any Active Directory Import tasks set up for your system.
Cloud Integration Server	The name of the Cloud Integration Service installed in the local network environment. The name of this server is <i>ServerName-CIS</i> .

#### 4. Click Save Connection.

The connection now displays in the list of remote data connections. You can now use this connection when setting up import and export utilities.

- Editing or deleting remote data connections
  - To edit a connection, select the connection in the list and then make the necessary edits in the **Configuration details**. Click **Save Connection** to save the changes.
  - To delete a connection that you no longer need, select the connection in the list and then click **Delete Connection**.



# Advanced Scheduler Features

Scheduler supports certain advanced features that extend the ability to automate and execute jobs.

## Using file watcher to trigger imports

You can use the Scheduler "file watcher" functionality to automatically trigger imports. To do this, you specify a folder to watch and an associated event handler. When a new or modified file is detected in that folder, the associated event is raised and any Scheduler jobs using that event will be run. A reserved variable can be used in the import task to import the file that raised the event.

**NOTE:** The presumed use case for file watcher is to trigger imports for execution, however, the event handler can be used to trigger any Scheduler job.

Use of file watcher requires the following setup steps:

- Defining the file watcher folder and associated event name. You can define any number of folders to watch. File watcher folders are defined per system.
- Configuring the Scheduler import job to use the file watcher event name and to process the file in that folder.

### Defining file watcher folders for Scheduler

On the **File Watchers** tab of Scheduler, you can define one or more folders to "watch" for new or modified files. If a new or modified file is detected, a specified event handler is raised, which can trigger one or more jobs for execution. The primary intent of the file watcher functionality is to trigger import jobs for execution, and to import the file that triggered the event.

**NOTE:** Only administrators can create and edit file watcher folders. Users with the **Scheduled Jobs User** permission cannot see the **File Watchers** tab in the Scheduler dialog.

To define a file watcher folder:

1. In the Scheduler dialog, on the Service tab of the ribbon, click File Watchers.

The File Watchers tab opens, displaying all defined file watcher folders.

2. At the bottom of the File Watchers tab, click **Add** to create a new file watcher.

The **Configuration details** area displays at the bottom of the tab.

- 3. Complete the following settings for the file watcher:
  - Folder Path: Click the Browse button to navigate to the folder to watch, or type the full path. This is the folder that Scheduler will watch for new or modified files. Sub-folders are not monitored.

The path must be a UNC path (i.e. \\servername\foldername). If a mapped drive is specified, it will automatically be converted to a UNC path.

Your Scheduler server(s) must be able to access the specified folder path in order to monitor the folder. Additionally, your Axiom Application Server(s) must be able to access the folder path in order to import the file when the import job is run. You should coordinate with your IT department to enable this access.

- File Filter: If desired, enter a file filter to restrict the file watcher functionality to a specified file type or file name. You can use wildcards (\* or ?). For example, you could specify \*.csv so that the associated event would only be triggered by new or modified CSV files.
- Event Name: The name of the event handler to raise when a new or modified file is found in the specified folder. This event will be raised in all systems. Any Scheduler job that contains an event handler with a matching name will be triggered for execution.

The **Enabled** check box is selected by default and should remain selected unless you want the folder to not be monitored at this time.

4. Click **Update** to save.

Scheduler will begin monitoring the folder the next time it checks for available jobs to run. It will continue monitoring the folder until the file watcher is deleted, disabled, or edited.

- To disable a file watcher, clear the **Enabled** check box in the **Configuration details** section, and then click **Update**. Note that this will disable the file watcher for all systems and all matching jobs—if you only want to disable a specific job, then you can disable the corresponding event handler in that job.
- To delete a file watcher, click Remove.
- To edit an existing file watcher, make changes as desired in the **Configuration details** section, and then click **Update**. Any changes will automatically be applied the next time Scheduler checks for available jobs to run.

### File monitoring details

Once Scheduler starts monitoring a file watcher folder, it will react to any new or modified file that is placed in that folder. It will not process any files that already exist in that folder when the monitoring begins.

File watcher functionality tracks files as follows:

- A Windows monitoring tool informs Scheduler if a new or modified file is found in the designated folder.
- If the file meets the file watcher filter, Scheduler checks the file name and modified date against a system table. If a matching combination already exists in the table, no further action is taken. If no match is found, the item is added to the system table, with an empty processing date.
  - If the file does not meet the file watcher filter, then it is ignored by that particular file watcher instance (however, the file may be picked up by a different file watcher that looks to the same folder but uses a different filter).
- The next time a Scheduler service checks the queue for available jobs to run, it will also check the
  file watcher system table for any items with an empty processing date. If found, the Scheduler
  service will raise the associated event handler name in all systems, and will stamp the processing
  date.
- Any Scheduler jobs with a matching event name will be added to the Scheduler queue as eligible
  for immediate processing (pending Scheduler availability and any higher-priority jobs in the
  queue).

The files in the file watcher folder are not deleted after being processed. They will remain in the folder until they are manually deleted or overwritten.

**NOTE:** If all non-system Scheduler services are stopped when a new or modified file is detected in the file watcher folder, then Scheduler will miss the notification from Windows and will not process the file. If your Scheduler services have been stopped, you may want to confirm that no new or modified files have been added to the file watcher folder(s). If a file was added while the services were stopped, you can delete the file and then re-copy it to the file watcher folder.

### Setting up an import to use file watcher

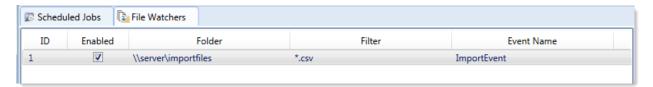
You can set up an import to be automatically executed when a new or modified file is detected in a watched folder ("file watcher" functionality). This topic discusses how to set up the import and the Scheduler job to use file watcher.

### Setup prerequisites

- In order to set up an import for use with file watcher, a file watcher definition must have already
  been created in Scheduler. This definition identifies the folder to watch for new or modified files,
  and the event to raise when these files are detected. For more information, see Defining file
  watcher folders for Scheduler.
- The Source tab for the import must be configured to Prompt for file during execution. This setup
  assumes that you want to import the new or modified file that caused the file watcher event to be
  triggered, and that you will use a variable in the Scheduler job to specify that file.

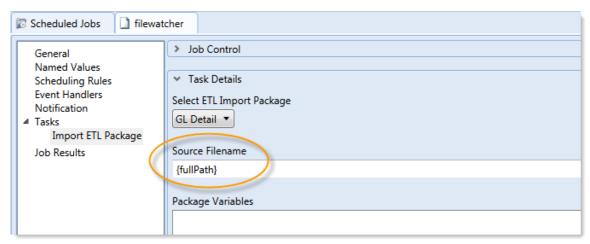
Setting up the import job in Scheduler

This example assumes that the following file watcher folder has already been defined on the File Watchers tab of Scheduler:



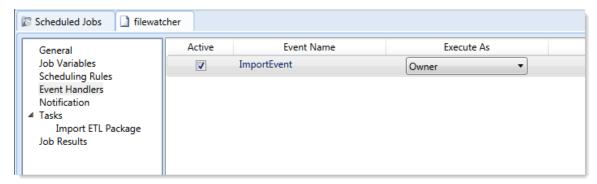
In Scheduler, create an import job as follows:

• Add an Import ETL Package task to the job, and select the import that you want to run. For the Source Filename, enter the variable {FullPath}.



• In the **Event Handlers** section of the job, create an event handler with the same name as the event defined in the file watcher definition.

For example, if the file watcher event is named ImportEvent, then the event in the Scheduler job should be named ImportEvent.



- Do not define any scheduling rules for the job—it will be automatically triggered for execution by the file watcher event.
- Configure the notification settings as desired.

For more information on setting up Scheduler jobs, see Scheduler Overview and Job properties.

### Example file watcher processing

The example file watcher folder and import job described above will work as follows:

- When a new or modified CSV file is placed in the specified file watcher folder, the event named ImportEvent will be raised in all systems.
  - The filter in this example limits the file watcher to CSV files. The filter could be left blank to respond to any new or modified file, or it could be more specific, such as looking for a particular file name.
- Since this job contains an event handler named ImportEvent, this job is added to the Scheduler processing queue, eligible to be processed immediately (pending Scheduler thread availability and any higher-priority jobs in the queue).
  - If any other jobs also contain an event handler named ImportEvent, those jobs would also be added to the queue.
- When this job is processed, the variable {fullpath} is resolved as the CSV file in the file watcher folder that caused the event to be raised.

This job will not be run again until:

- The existing file in the file watcher folder is overwritten with a newer file of the same name. The
  modified date of the file must change in order for the file to be recognized as a new file to be
  processed.
- A new file with a different name is placed in the file watcher folder. In this case the older file that was already processed is ignored (assuming it is still in the folder), and the new file is used.

For more information on how file watcher tracks files to be processed, see Defining file watcher folders for Scheduler.

### Using RunEvent to execute a job

Using RunEvent, you can trigger the execution of a Scheduler job from various contexts, such as within Axiom files, task panes, or Axiom forms. There are two different versions of RunEvent:

- **RunEvent function**: The RunEvent function can be used in Axiom files to trigger the execution of a Scheduler job from a spreadsheet.
- **RunEvent command**: The RunEvent command can be used in task panes or Axiom forms to trigger the execution of a Scheduler job.

Both the function and the command work in the same way and take the same parameters. Some limitations apply depending on the context where RunEvent is being used.

All of the information necessary to run the job is contained within the RunEvent function or command. It is assumed that an administrator (or a power user with the necessary rights) sets up the desired jobs within Scheduler, and then sets up RunEvent in the appropriate context so that end users can trigger it.

The end user who triggers the job using RunEvent does not need to have any Scheduler security permissions. The job itself can be configured to run using the permissions of the job owner or using the

permissions of the end user who triggered the job (the requester). If the job is run as the requester, then the end user must have the appropriate permissions to the files impacted by the job (for example to the target file for File Processing, or to the target file group and plan files for Process Plan Files).

### Setting up a Scheduler job for RunEvent

All uses of RunEvent require the same job setup in Scheduler:

- The job that you want to execute via RunEvent must already be created within Scheduler. When setting up the job, consider items such as the notification settings. Do you want the notifications to go to the user that executed the job, or to the job owner, or both?
- The job must contain an event handler that will be used to trigger the job execution. When creating the event handler, consider whether you want the job to run as the job owner, or as the requester (the user who clicks on the RunEvent function). This may impact email notifications and determines the user rights under which the job will run.



For more information, see Creating event handlers for a job.

• Optionally, the job can use variables. Variable values can be defined in the RunEvent function or command and passed to the job. You would do this if aspects of the job need to be dynamic; for example, if you want to use a different filter depending on which user is running the job or based on a user selection in the file. For more information see the *Variable example* section below.

### Setting up RunEvent

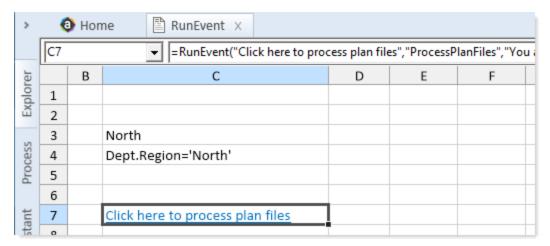
Regardless of the context, RunEvent uses the following properties to trigger Scheduler jobs:

- The event handler name that identifies the Scheduler job(s) to trigger for execution.
- An optional confirmation message to present to the user before proceeding with the event. Not available when using the command within an Axiom form.
- An optional success message to present to the user after the event has been raised.
- If variables are being used, one or more variable names and values to pass to the Scheduler job. This is available in all contexts, however, task panes do not currently support the ability to determine the variable values dynamically.

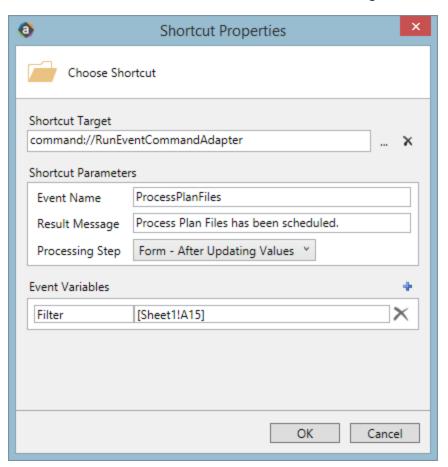
The following shows an example RunEvent function for use in an Axiom file:

```
=RunEvent("Click here to process plan files", "ProcessPlanFiles", "You are about to process plan files for the "&C3&" region. Do you want to continue?",, "filter = "&C4)
```

The first parameter defines the display text for the function, while the second parameter specifies the event handler name. In this example we have also defined a custom confirmation message for the user and a variable value to pass a filter to the job. The following screenshot shows the function in the spreadsheet:



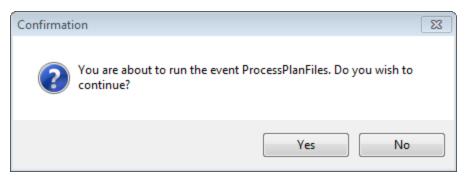
The next screenshot shows a RunEvent command set up on a Button component for an Axiom form. You can see the same event name and the filter variable also being read from a sheet location.



### User experience

The user experience for RunEvent depends on the context and whether you are using optional custom messages.

- The user starts the process by doing one of the following:
  - Double-clicking the RunEvent function in the spreadsheet.
  - Clicking the Button component that is configured for RunEvent in the Axiom form.
  - Double-clicking the RunEvent item in the task pane.
- A confirmation prompt displays to the user, asking them to confirm that they want to proceed. The user can click **Yes** to proceed, or **No** to cancel. Default text is used if no custom text is defined in the RunEvent properties.



**NOTE:** This step does not apply when executing RunEvent from an Axiom form. The Axiom form context does not support a confirmation message. However, you can configure the Button component to display a confirmation message before executing the RunEvent command.

• Axiom Software checks all Scheduler jobs to see if any contain an active event handler with the same name as specified within the RunEvent properties. If any are found, they are added to the schedule to be processed as soon as possible, depending on Scheduler thread availability and any higher-priority jobs in the queue.

It is important to note that RunEvent triggers job execution based on the event handler, not based on specific jobs. If multiple jobs contain an event handler with the specified name, then all of those jobs will be scheduled.

If variable values are defined in the RunEvent properties, those values are passed to the job. If a variable specified in the RunEvent properties is not used in the job, it is ignored.

- A confirmation message displays to the user as follows:
  - If no jobs were found that contained the specified event handler, the user is notified that no jobs were found.
  - If jobs were placed on the schedule, the user is notified that the specified event was scheduled. Default text is used if no custom text is defined in the RunEvent properties.

**NOTE:** If executing RunEvent from an Axiom form, this message displays in the bottom left corner of the form, not in a separate message dialog.

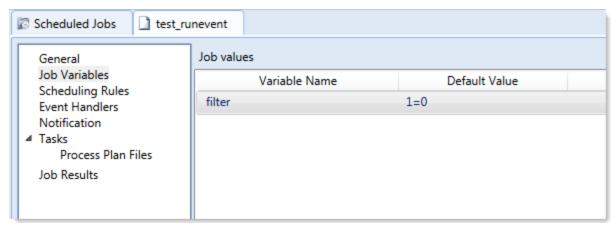
When the job is finished processing, email notifications are sent according to the settings in the job.

### Variable example

When using RunEvent to execute a Scheduler job, you can pass a variable value to the job. For example, imagine that you want to execute a Process Plan Files job, and you want to send a filter value to the job.

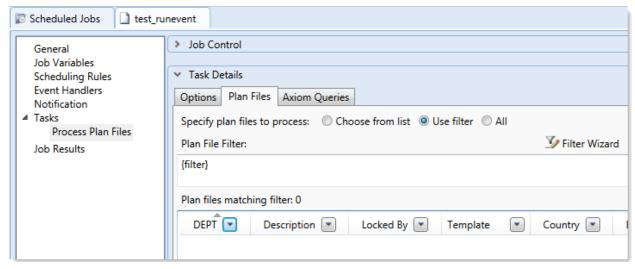
### Step 1: Set up the variable in the job

The first step would be to create a job variable in the job, and then use the job variable in the filter setting.



Example job variable

**NOTE:** In this case, we have defined a default value for the filter variable (1=0) that does not result in any plan files. This is because we do not want to process any plan files unless a filter is provided by RunEvent. If we left the default value blank, that would mean all plan files would be processed if no filter was provided by RunEvent.



Example use of variable in job settings

For more information on Scheduler job variables, see Using job variables.

#### Step 2: Configure RunEvent to use the Variable

Now that the job is set up to use the filter variable, you must configure RunEvent to pass in a value for that variable. If you are using the RunEvent function in an Axiom file, you use the following syntax within the function parameters:

```
variablename=variablevalue
```

These name / value pairs can be placed in the RunEvent function starting in the fifth parameter of the function. If you have two name / value pairs to pass to the job, you can use the fifth and sixth parameters, and so on.

For example, to pass the filter DEPT.Region='North' to the job, the RunEvent function would be constructed as follows:

```
=RunEvent("Double-click to process plan
files", "ProcessPlanFiles", , , "filter=dept.region='North'")
```

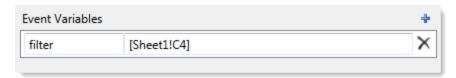
More likely, you would be reading the variable value from another place in the sheet, so the function would look something like:

```
=RunEvent("Double-click to process plan files", "ProcessPlanFiles", , , "filter="&C4)
```

Where the filter value is read from cell C5.

When the job is executed by use of this RunEvent function, the value DEPT.Region='North' will be placed in the Plan File Filter box of the Process Plan Files task, and the job will be run using that filter.

When using RunEvent in an Axiom form, the variables and their values are defined in the Event Variables section. In this context you place the name of the variable in the left-hand box, and then in the right-hand box you enter the cell reference (in brackets) where the variable value will be read.



The Event Variables section is also present when configuring RunEvent for use in a task pane, however in this context the only option is to "hard-code" the values in the RunEvent properties.



# Scheduler Task Reference

Each Scheduler task in a job has two sections of task properties:

- Task Control: Common task properties that apply to all task types. For more information, see Task Control properties.
- Task Details: Properties specific to the current task type. For more information, see the topics for each individual task type.

The following task types are available:

Task	Description
Active Directory Import	Import users from Active Directory into Axiom Software. This task adds new users, and can also disable users that no longer exist in the Active Directory domain.
Collect Worksheets	Collect worksheets from multiple files into a single file.
Create Plan Files	Create new plan files (same as the Create Plan Files utility for file groups).
Echo Task	Test the Scheduler server. This task sends a message to the Scheduler server and asks it to send the message back.
Execute Command Adapter	Execute a command from the Command Library.
Execute SQL Command	Run a SQL statement on an Axiom database.
Export ETL Package	Export data to an external database, using an export utility defined in the Exports Library.
File Processing	Perform file processing actions on a report. You can use the report's native file processing settings, or override the settings.
Import ETL Package	Import data into Axiom Software, using an import utility defined in the Imports Library.
Process Document List	Process any set of Axiom files—for example, driver files or report utilities. The task calculates and saves the files, and can also refresh Axiom queries and save data to the database.
Process Plan Files	Process plan files (same as the Process Plan Files utility for file groups).

Task	Description
Process Template List	Process a template file. The task runs designated Axiom queries, timestamps the queries, and saves the template.
Purge System Data	Purge old Scheduler results and system temp tables.
Raise Event	Trigger another Scheduler job for execution, using a named event handler.
SMTP Message Delivery	Deliver email notifications resulting from Scheduler jobs.
Start Process	Start a process definition for Process Management.
Update Indexes and Constraints	Update the indexes and constraints in your Axiom Software database.

# **Task Control properties**

The following task properties are available for all Scheduler task types. To edit these properties, select the task in the Scheduler job, then expand the **Task Control** section.

Item	Description
Skip this Task	If selected, the task will not be run when the job is processed.
	By default, this option is not selected, which means this task will be run.
Process task only if the value of this expression is true	Optional. Enter a logical expression to conditionally process this task depending on whether the expression resolves to true or false at the time the job is executed. If true, the task is processed as normal. If false, the task is skipped.
	The logical expression is evaluated by the Scheduler server using an IF function. The expression can be any statement that would be valid within an IF function. You can use Excel functions, Axiom functions, and Scheduler job variables in the expression. If you use a job variable in the expression, you must place the variable in double quotation marks unless you expect it to be resolved and evaluated as a number.  For more information, see Conditionally processing tasks in a job.
(Colors Table Colls	
If this Task fails, continue executing subsequent Tasks	If selected, the job will continue processing even if this task fails.
	By default, this option is not selected. If a task in a job fails, the job is canceled and no further tasks are processed.

Item	Description
Create a Subordinate Job for this Task	If selected, this task will be processed as a subordinate job to the current job.
	Selecting this check box enables concurrent processing of different tasks, if the option to Wait for all Subordinate Jobs to complete before proceeding to the next Task is not selected.
	NOTE: This option is not available for Process Plan File tasks.
Wait for all Subordinate Jobs to complete before proceeding to the next Task	If selected, the job will wait for any subordinate jobs to complete before moving to the next task.
	If this check box is not selected, and the option Create a Subordinate Job for this Task is selected, then tasks can be processed concurrently instead of sequentially.
	This check box is selected by default for Plan File Refresh and File Processing tasks. For other task types, this option is not selected by default.
Workbook processing engine to use	This option should always be set to <b>Axiom Web Engine</b> . Use of Excel for processing on the Scheduler server is no longer supported.
	If any tasks in the job involve spreadsheet processing, the spreadsheets are processed using the same spreadsheet emulation engine used by the Windows Client.
Override Log Level for this Task	By default, Scheduler jobs perform logging at the same level that is specified for the application. If necessary, you can override the logging level for a particular task, so that it always runs at a specified logging level. You may want to do this if you encounter performance issues for tasks that generate a lot of logged messages.
	To do this, select the check box for <b>Override Log Level for this Task</b> , then select the desired logging level from the drop-down list.
	<b>NOTE:</b> This option is only available for File Processing tasks.

### Iteration

This section can be used to optionally enable iterative processing for the task. For more information, see Using iterative task processing.

Item	Description
Iterate this Task	Specifies whether iterative processing is enabled for the task. If enabled, then the task will be performed N times, where N is the number of unique items in the specified iteration column. Job variables can be used to apply the current iteration value and iteration number to the task.
Create a Subordinate Job for each iteration	Specifies whether each iteration is processed as a separate subordinate job. By default, this is disabled, which means that all iterations are processed sequentially within the overall subordinate job created to process the iterations.
	If enabled, then each iteration is processed as a separate subordinate job, enabling concurrent execution of multiple iterations. This option should only be enabled if the order of iteration processing is not important.
Column	The column that contains the values to iterate over. Use Table.Column syntax to specify the column. Multiple-level lookups can be used.
	For example, if you specify <code>Dept.Region</code> , then the task will be processed once for each unique region value in the column (after applying any filter to limit the list of values).
Group By	Optional. By default, the group by column is the same as the iteration column, so that the task is processed once for each unique value in the iteration column. However, if needed, you can specify a different grouping level.
	You can use any column or columns that would be valid as the "sum by" level for an Axiom query, where the primary table is the table specified for the iteration column.
Order By	Optional. By default, the values are sorted based on the iteration column, in ascending order. You can specify a different sort column, or use the same sort column but change the order to descending.
	The sort order is ascending unless the keyword <code>desc</code> is used to specify descending order. For example:
	Dept.Dept desc
Filter	Optional. A filter criteria statement to limit the list of values for the iterative processing. You can use any filter that is valid against the source table (the table of the iteration column).

# Active Directory Import task

This task imports users from Active Directory groups into Axiom Software security. For more information on using Active Directory integration with Axiom Software, see the *Security Guide*.

This task has three tabs of settings: Source Directory, Notification, and Preview Import.

**NOTE:** The user running this task must be an administrator or have the **Administer Security** permission.

For Cloud Service systems, this task can import users from your local Active Directory by use of the Axiom Software Cloud Integration Service. If you have a remote data connection that is enabled for user authentication, this task will use that connection when the job is executed by a Scheduler server. For more information, see Managing remote data connections.

### Source Directory tab

On this tab, you specify the domain to import from and the groups to import.

Item	Description
Domain or Server	<ul> <li>Select either Domain or Server to specify the domain to import from.</li> <li>If you select domain, type the name of the domain.</li> <li>If you select server, type the name of the domain controller server.</li> <li>The server option is available in case you are not currently logged into the target domain, and your current domain does not have access to the specified target. You must enter domain credentials in order to access the domain.</li> <li>Only one domain can be selected per import task. If you want to import users from multiple domains into an Axiom Software system, then you must create multiple import tasks.</li> </ul>
Credentials	<ul> <li>Specifies the credentials to use when accessing Active Directory for the import.</li> <li>Select one of the following:</li> <li>Use process credentials: (Default) Use the credentials of the network service account of the Scheduler server.</li> <li>Specify domain credentials: Use the credentials of a specified domain User and Password.</li> </ul>
Never Enable Users	<ul> <li>Specifies whether the import enables imported users as part of the process:</li> <li>If unchecked (default), then newly imported users are enabled as part of the import. Additionally, any existing imported users who have been changed to disabled are re-enabled.</li> <li>If checked, then newly imported users are not enabled as part of the import. An administrator must modify the security settings after the import is complete to enable the new users. Existing imported users retain their current enabled status.</li> </ul>

Item	Description
Groups to import	The Active Directory groups for which members will be imported into Axiom Software Security.
	<ul> <li>Click Add to select from a list of groups for the specified domain. If the specified domain name is not valid or if Axiom Software cannot connect to it, then an error will result when attempting to add groups.</li> <li>If you need to remove a group, select the group and click Remove.</li> </ul>
Role Mapping	Click this button to map Active Directory groups to Axiom Software security roles and to user types. If a mapping exists for a group, then when users are imported for that group they are automatically assigned to the mapped role. You can map each group to multiple roles.
	In the Role Mapping dialog, click Add mapping (the plus icon) to add a role mapping. Then complete the following:
	<ul> <li>In the Directory Group column, select the Active Directory group to be mapped.</li> </ul>
	<ul> <li>In the Axiom Role column, select the role to be assigned to users in that group. If you want to map the group to more than one role, add another mapping row. You can select None if the users should not be assigned to any role.</li> </ul>
	<ul> <li>In the User Type column, select the license type for the imported users. The default license type is Standard.</li> </ul>
	If no mappings are defined for a group, then users in that group are not assigned to any roles when they are imported, and the user type is set to Standard.
	To remove a role mapping, select the mapping in the grid and then click <b>Remove mapping</b> (the X icon). If users have already been imported using this role mapping, removing the mapping will not remove the users from the role in subsequent imports.
	<b>NOTE:</b> If a user belongs to multiple imported Active Directory groups, and the groups do not have the same assigned user type, then the user will be assigned one of the user types.

### Notification tab

On this tab, you specify users to be notified when changes are made in Axiom Software Security due to the import.

Type in one or more email addresses to be notified. Separate multiple addresses with a semi-colon. For example:

jdoe@axiomepm.com;jsmith@axiomepm.com

When the import task is run, if any users are created or modified in the Axiom Software system, an email notification will be sent to the addresses specified here. The email summarizes the changes made. This email notification is independent of any job notification settings (which notify based on overall job completion or failure).

We recommend setting up notification to send emails to the administrator(s) responsible for maintaining the security settings in Axiom Software, so that he or she can define security settings for newly added users, validate changes made to existing users, and perform any other follow-up tasks.

Email notifications for Scheduler tasks depend on an SMTP Email Delivery task being run at regular intervals. For more information, see Configuring Scheduler system jobs.

Job variables can be used in this setting.

### Preview Import tab

On this tab, you can preview the import results to test that the import is set up as desired.

To preview the results, click **Preview**. Axiom Software processes the import task but does not actually make the changes to the system. Instead, the tab displays a summary of the changes that would result.

The preview shows a list of users that would be added, changed, or disabled.

**NOTE:** The preview is always executed locally, even for Cloud Service systems. The remote data connection to the Cloud Integration Service is only used when the task is executed by Scheduler.

### Collect Worksheets task

This task collects sheets from multiple source workbooks and combines them into a single target workbook. You can then save the target workbook to a specified file location, and/or email the workbook.

**NOTE:** Alternatively, you can use file processing to perform a file collect operation. The file processing report can then be scheduled using a File Processing task. For more information on setting up a file collect report, see the *Axiom File Setup Guide*.

Typically, this task would be used at the end of a job with multiple File Processing tasks, to take the results of those tasks and collect them into a single workbook.

Item	Description
Save or Email Workbook	Specifies the delivery option for the target workbook. Select one of the following:
	<ul> <li>Save Workbook: The target workbook is saved to the specified output folder.</li> </ul>
	<ul> <li>Email Workbook: The target workbook is emailed to the specified recipients.</li> <li>The file is not saved anywhere on the file system.</li> </ul>
	Save and Email Workbook: The target workbook is both saved and emailed.

### ► Target Workbook

Complete the following settings to define the target workbook:

Item	Description
Output Folder	The folder location where the target workbook will be saved (if you are saving the workbook). Click the folder icon to select a folder location, or type a folder location.
	If the specified folder does not already exist, Axiom Software attempts to create it.
	Job variables can be used in this setting.
Output File Name	The name of the target workbook. Job variables can be used in this setting.
File Type	The file type of the target workbook. Select XLS, XLSX, or XLSM.
	<b>NOTE:</b> PDF displays as an option, but it is not supported in this context.

### Email Settings

This section only applies if you are emailing the target workbook. The "From" address is always the Scheduler default From address (as defined in the system configuration settings).

Item	Description
То	Enter the email addresses to receive the target workbook via email. Separate multiple addresses with a semicolon.
Subject Line	The subject line for the email.
Body Text	The body text for the email.

#### Source Workbooks

In this section, you specify one or more source workbooks from which to collect worksheets. Workbooks are identified by folder location. Within a folder location, you can specify one or more workbooks by name, or by using wildcards, or by using \*.\* to collect all workbooks at the location.

All sheets in each source workbook will be collected. Ideally, you will be collecting from workbooks that only contain relevant sheets (for example, no blank "Sheet2," etc.), and where the sheets have unique names. If multiple workbooks have sheets with the same name, the sheets will be incremented by number in the target workbook.

- To add a workbook, click the Add button. In the Edit Workbooks Source dialog, complete the settings as described below, then click OK to add the workbook to the list.
- To remove a workbook, select the workbook in the list and then click the Remove button. Only one workbook can be selected at a time.
- To change the order of workbooks, select the workbook in the list and then click the arrow buttons to move the workbook up or down. Source workbooks are processed in the order they are listed in the grid.

Item	Description
Folder Path	The folder location of the source workbook(s). Click the folder icon to select a folder location.
	<b>NOTE:</b> The <b>Folder Path</b> location must be accessible by the Scheduler service user account. If you specify a network folder location using the Browse button, the location is automatically entered as a UNC path. If you specify a C: drive location, that will be evaluated as the C: drive of the Scheduler server.
	Job variables can be used in this setting.
Workbooks	The workbooks from which you want to collect worksheets, within the specified folder path.
	<ul> <li>Specify *.* if you want to collect all files in the folder path.</li> </ul>
	<ul> <li>Specify individual file names to collect from specific files. Separate multiple file names with semicolons.</li> </ul>
	You can use wildcards (* or ?) to specify groups of files that share naming conventions. For example: $North*.xls$ to collect all XLS files where the file name starts with "North".
	Only files with the following file types are valid to be collected: XLS, XLSX, XLSM. If you are using wildcards, the matches must be valid file types, or else the task will fail with an error.
	Job variables can be used in this setting.

Once you have saved a source workbook location, you can edit it by double-clicking the row.

## Create Plan Files task

This task creates plan files for a file group (same as the **Create Plan Files** utility that is available from the file group menu).

This task has two tabs of settings in the Task Details area: General and Plan Files.

**NOTE:** If you are using Create Plan Files to create new on-demand plan files, those plan files will be automatically started in the designated **Plan File Process** for the file group. This only applies when creating a brand new plan file. If an existing plan file is overwritten, its process status will be left as is.

#### General tab

The following settings are available on the General tab:

Item	Description
Select File Group	The file group for which plan files will be created. You can select any file group or file group alias.
	<b>NOTE:</b> If you select an alias, then you cannot select individual plan files on the <b>Plan Files</b> tab. Only the <b>Use Filter</b> and <b>All</b> options are supported for use with aliases. This is because the alias could change to point to any file group, which could result in a different list of plan files.
Overwrite existing plan files?	By default, this option is not selected, which means that existing plan files will not be overwritten, even if the plan file is selected to be created.
	If selected, existing plan files will be overwritten.

#### Plan Files tab

On the Plan Files tab, specify the plan files that you want to create. This tab lists all plan codes that you have the right to access. (If a plan code has been set to False in the designated Show On List Column for the plan code table, then it is not available in this list.)

You can create plan files in any of the following ways:

• Create all plan files: To create all plan files, select All. This will cause all plan files to be created, for all existing and future plan codes.

Alternatively, you can select **Choose from list** and then select the check box in the column header, causing all plan codes to be selected, but then the list of plan codes is fixed and will not adjust for any future changes. For example, if you add a new department in the future, that new department will only be created by this task if you use the **All** option.

• Create selected plan files: To create certain plan files, select Choose from list and then select the check boxes for the desired plan codes.

To find the plan files you are looking for, you can sort, filter, and group the list using standard Axiom grid features. You can show additional columns and hide columns by right-clicking in the column header. If you have filtered the list, you can select the check box in the header to select only the plan codes that currently display in the dialog.

**NOTE:** This option is not available when using a file group alias as the selected file group for the task. This is because the alias could change to point to any file group, which could result in a different list of plan files.

• Create a subset of plan files using a filter: To use a filter to create a subset of plan files, select Use filter, and then type a filter into the filter box. You can also use the Filter Wizard to build the filter. The filter must use the plan code table or a reference table that the plan code table links to. For example: DEPT.Region='West'.

Once you have entered a filter, you can click **Refresh plan file list** to show only those plan codes that currently match the filter. This feature is to help you determine whether you have defined the filter as intended. The filter will be applied to the list of plan codes when the Scheduler job is processed, so if changes have been made to the plan code table since then, the actual list of plan files processed will reflect those changes.

You can also use a job variable for the filter. For example, you can define a job variable named "filter" and then place the text {filter} in the filter box. This is intended for use when running Scheduler jobs by using the RunEvent function. If a value is specified in the RunEvent function, such as "Filter=dept.region='west'", then that filter will be used in place of the {filter} variable to determine the list of plan files to be created.

**NOTE:** If you use a variable, and you leave the default value for that variable blank within the **Job Variables** tab, then all plan codes will be created if no value is passed by the RunEvent function (or if the value is invalid). You may want to define a default filter that results in no values (such as 1=0), so that plan files are only created if a valid filter value is passed.

**IMPORTANT:** For all of these options, the **Overwrite existing plan files** option on the General tab determines whether all selected plan files are created, or only the plan files that do not already exist.

### Echo task

This task is used for testing purposes only, to check whether a Scheduler server is running and operational. The task sends a message to the Scheduler server, and asks it to send the message back (an "echo"). If successful, the message displays in the job results. No other action is performed.

ltem	Description
Message to Echo	The message to send to the Scheduler server for testing.
	Job variables can be used in this setting.
Sleep Time	The time to pause in between message echoes, in seconds. Scheduler will echo the message once, then wait the specified sleep time, then echo the message again.

## **Execute Command Adapter task**

This task executes a selected command from the Command Library.

#### Task properties

This task has one property named Command Name that specifies the command to execute.

To select a command to execute:

- 1. Click Edit Command.
- 2. In the Shortcut Properties dialog, click the browse button [...] to the right of the Shortcut Target box.
- 3. In the Axiom Explorer dialog, select the desired command from the Command Library, then click Open.
  - This returns you to the Shortcut Properties dialog. The selected command is now listed in the Shortcut Target box, and the Shortcut Parameters section displays the parameters for the command.
- 4. Complete the **Shortcut Parameters** for the command as needed. The available parameters depend on the selected command.

You can later edit the shortcut parameters or select a different command by clicking Edit Command.

#### Supported commands

Only certain commands are available for execution in this context. The following commands are available:

• File Group Rollover

## **Execute SQL Command task**

This task runs a SQL statement on an Axiom database. By default, each system contains a system job that runs this task to maintain the indexes on your Axiom databases. If needed, you can also use this task in a user-defined job to run any valid SQL statement on an Axiom database.

**NOTE:** The SQL statement in this task will be run as the Axiom database user, regardless of which user executes the job.

This task has the following settings:

Item	Description
Source Axiom Database	<ul> <li>Select the database on which to run the SQL statement:</li> <li>Current system database: The database for the current system.</li> <li>Current audit database: The corresponding audit database for the current system.</li> </ul>
SQL Command Text	Enter any valid SQL statement to be run against the specified database.  To validate the syntax of the SQL statement, click the Check SQL syntax button . Axiom Software sends the statement to your database server to see if the statement can be parsed, resulting in either a success message or an error message.
	Job variables can be used in the SQL statement. The Check SQL syntax button is not available if the statement uses variables.

## **Export ETL Package task**

This task exports data from Axiom Software to an external database (same as executing an export from the Exports Library).

This task has one setting, **Select ETL Export Package**. This is the name of the export package to process. You can select any export that is defined in the current system.

## File Processing task

This task performs file processing on a specified report file (or file group utility). The file must already be enabled for file processing. You can use the file processing settings that are already in the file, or you can override any setting.

**NOTE:** Although other file types can use file processing, they cannot be processed in this task. If you have a non-report file set up to use file processing, you must process it manually.

You can have multiple File Processing tasks within a single job to perform "batch" processing of files. Each task can process a different file, and/or process the same file with different settings (using overrides).

The following settings must be completed for the task:

Item	Description
File to Process	The report to process for the task. Click the <b>Browse</b> button to open the <b>Axiom Explorer</b> dialog, and then select a report to process.
	Only one report can be selected for each File Processing task. If you want to process multiple reports, you can add multiple File Processing tasks to the Scheduler job.
	<b>TIP:</b> Once the file is selected, only the file name displays in the task. If you want to know the folder location of the selected file, hover your cursor over the field. The folder location is listed in the tooltip.
Process	Specifies whether the report will be run using multipass processing.
Multipass	<ul> <li>If this option is selected, multipass processing is performed. This is equivalent to selecting File Output &gt; File Processing &gt; Process File Multipass.</li> </ul>
	<ul> <li>Otherwise, multipass processing is not performed and multipass settings do not display in the task. This is equivalent to selecting File Output &gt; File Processing &gt; Process File.</li> </ul>
	<b>NOTE:</b> If you select <b>Process Multipass</b> , but the file does not have any defined multipass settings, then you must override the blank multipass settings for the file and define them in the equivalent of "advanced mode." If you want to use "basic mode" settings (specify only a source column and Axiom Software automatically completes the rest of the settings for you), then you should edit the file to define the basic mode multipass settings so that they can be inherited by the task.
Enable iterative calculation while processing	Specifies whether iterative calculations are enabled for the file during processing. In most cases you will leave this option disabled.
	If this option is selected, then iterative calculations are enabled for the file during the Axiom query refresh process. The iterative calculation settings are fixed at a maximum of 100 iterations and a maximum change value of .001.
	For more information on iterative calculations, see the Microsoft Excel Help.

### Advanced options

This section only displays if multipass processing is enabled for the task, and the task uses settings that are eligible for parallel processing. Click on the down arrow next to the title to expand the section and view the options.

Parallel processing for file processing tasks is performed based on multipass passes. With certain task settings, multiple passes can be separated into sub-jobs, which can then be processed at the same time (in parallel). This can improve the performance of the task.

For example, imagine that you are multipass processing a file by department. If the task is processed sequentially, then the task would process Dept 100 and finish it, then move to Dept 110 and finish it, etc. When parallel processing is used instead, Depts 100-199 can be separated into one sub-job, Depts 200-299 into another sub-job, etc. Because the sub-jobs are processed in parallel, multiple departments are processed at the same time, so the overall task can complete more quickly.

Item	Description
Maximum Parallel Jobs	The maximum number of subordinate jobs to run in parallel. The default number is 4.
	This is the total number of sub-jobs that can be run at the same time for this task. Ultimately the number of sub-jobs that are run in parallel depends on the number of Scheduler threads that have been configured for use at your organization, and the number of Scheduler threads that are currently available (threads that are not processing other higher-priority jobs).
Processing Batch Size	The number of multipass passes to include in each sub-job at a time. The default number is 10 for eligible snapshot and export processes, and 7000 for eligible save-to-database processes (save once at end).
	Passes are determined based on the multipass list of items. For example, if you are processing by department (DEPT.DEPT), then each department is a separate pass. If the batch size is set to 10, then each sub-job would process 10 departments at a time.
	In most cases, the default settings are sufficient. If you are experiencing lengthy processing times and want to optimize performance, you can adjust this setting as follows: divide the number of passes by the number of available Scheduler threads. For example, if there will be 100 passes and there are 4 Scheduler threads, set the batch size to 25.

#### **NOTES:**

- For save processes, only "save once at end" processes are eligible for parallel processing. In this case, the records to be saved to the database are extracted after each pass to a central temporary table. Once all passes are complete, then all records are saved to the database from the temporary table. Save processes where data is saved directly after each pass are not eligible, because these processes may depend on sequential processing.
- There is no way to disable parallel processing if the task is eligible; however, you can adjust the parallel processing settings if desired.

#### ► File processing settings

Once you have selected a file to process, the file processing settings from that file display within the task as read-only. You can leave the settings as they are, or you can override any setting.

- To override a setting, select the **Override** check box to the right of the setting. The setting becomes editable, and you can change it. The change only applies to the file processing task—the setting remains unchanged within the file.
- If you override a setting, make sure that any related settings make sense in the context of the
  change. For example, if File Generation is set to Multiple Output Files, and you override it to be
  Single Output File, then you should also check the Sheet Names setting to make sure that you
  will end up with unique sheet names within the file.

For more details on file processing settings, see the Axiom File Setup Guide.

Note the following about specific settings for the task:

- The Output Folder location must be accessible by the Scheduler service user account. If you specify a network folder location using the Browse button, the location is automatically entered as a UNC path. If you specify a C: drive location, that location will be evaluated as the C: drive of the Scheduler server.
- If the file processing type is **Print**, the Scheduler server(s) must be configured to access the specified printer. This may require the assistance of your IT department.

Job variables can be used in any file processing setting that accepts a typed user input.

#### Batch variables

If the file has defined batch variables, you can specify variable values to be used for the file processing task. When the task is executed, any specified variable values are temporarily placed within the file, within the designated cell for that variable value. The file can be set up to use the variable value in some way during processing.

Item	Description
Variable Names	The names of the variables, as defined in the File Processing Control Sheet for the source file. If no names are listed, then no variables are defined in the file.
Variable Values	The variable values to be placed within the file when the file processing task is executed. You must type a "free-text" value—batch variables do not have defined lists to select from.

Job variables can be used in the batch variable settings.

## Import ETL Package task

This task imports data into Axiom Software (same as executing an import from the Imports menu).

#### **NOTES:**

- If the import package is configured to Ignore lookup and key errors, then if errors are found the execution status of the job will be Partial Success. This will trigger an email notification if the job is configured to notify only on error.
- If you want to use the "file watcher" functionality to automatically trigger an import to run when a source file is placed in a designated folder, see Using file watcher to trigger imports.

Item	Description
Select ETL Import Package	The import package to process. You can select any import that is defined in the current system.
Source Filename	The path and name of the source file. This option only applies in the following situations:
	<ul> <li>The import is configured to pull data from a source file (instead of a database table).</li> </ul>
	<ul> <li>The import is configured to prompt the user for the source file during execution.</li> </ul>
	If the import is configured to always use the same source file, then that file displays for reference in the <b>Source Filename</b> box, but it is grayed out and cannot be changed.
	<b>NOTE:</b> If you are configuring the import to use file watcher, enter the predefined variable {fullpath}.
	Job variables can be used in this setting.
Package Variables	Specifies values for any variables used in the import package.
	Variables are listed in the right-hand side of the grid. Use the drop-down list next to the variable name to select from the defined set of choices, or type in a value.
	Job variables can be used in this setting.

## Process Plan Files task

This task processes plan files in a file group. It performs the same actions as the **Process Plan Files** utility available from the file group menu.

The Process Plan Files task uses several tabs to define different options. The available tabs and the options on those tabs depend on the selected **Processing Mode** on the **Options** tab.

- Options: Defines the overall processing mode and processing options
- Plan Files: Specifies the plan files to process
- Axiom Queries: Specifies which Axiom queries to run in plan files (only applies to Normal Processing)
- **Utilities**: Specifies which data source to use for utility processing (only applies to Process with Utilities)
- **Processing Variables**: Defines variables to pass into plan files from Scheduler, and to Scheduler from plan files

#### Options tab

The following options are available on the Options tab:

Item	Description
Processing Mode	Select the type of processing to perform:
	<ul> <li>Normal Processing: Plan files are opened, refreshed, and saved. You can configure which actions occur.</li> </ul>
	<ul> <li>Process with Utilities: A list of utilities is iteratively processed per plan file. Utilities are opened, refreshed with data for each plan code, and saved. This is primarily intended for processing form-enabled composite plan files.</li> </ul>
	<ul> <li>Update Persistent Plan Files: Update existing plan files for text, formatting, or formula fixes. This is an advanced feature.</li> </ul>
	<ul> <li>Process with Custom Utility: Plan files are processed using a custom utility provided by Kaufman Hall Software Support. This is an advanced feature.</li> </ul>
	The default processing mode is Normal Processing. However, if the file group has been configured so that utility processing is the default processing mode for that file group, then Process with Utilities is selected by default.

Item	Description	
Select File Group	The file group for which plan files will be processed. You can select any file group or file group alias.	
	NOTES:	
	<ul> <li>If you select a file group alias, then you cannot select individual plan files on the Plan Files tab. Only the Use Filter and All options are supported for use with aliases. This is because the alias could change to point to any file group, which could result in a different list of plan files.</li> <li>File group scenarios are not available on the list cannot be processed</li> </ul>	
	via Scheduler.	
Advanced Options: Worker Batch Size	Optional. Specifies the number of plan files to be processed in each batch. The batch size must be a number between 10 and 100.	
	By default this is left blank, which means that the batch size is automatically calculated based on the number of plan files to be processed divided by the total number of threads on all enabled Scheduler servers. Generally speaking, you should not customize this setting unless you are advised to by Axiom Software Support.	
	<b>NOTE:</b> Each batch of plan files is processed by a subordinate job. These subordinate jobs are automatically created for the Process Plan Files task and are processed in parallel, dependent on the number of Scheduler threads that are available at any one time.	

### **Options for Normal Processing mode**

If Normal Processing is the selected processing mode, the following additional options are available on the Options tab:

Option	Description	
Save document after processing	Specifies whether plan files are saved during processing. This option is selected by default.	
	This option does <i>not</i> cause a save-to-database to be performed—that option must be selected separately.	
	NOTES:	
	<ul> <li>If this option is not selected, then the utility will open the file as read-only and will not attempt to acquire the document lock before processing.</li> </ul>	
	<ul> <li>If the file group uses virtual plan files, this option does not apply because the plan files cannot be saved. However, if the option is enabled, Axiom Software will attempt to acquire the document lock before processing, which is not necessary. This option should not be enabled when processing virtual plan files.</li> </ul>	
Run Save To Database on	Specifies whether a save-to-database is performed in plan files during processing. This option is selected by default.	
plan files after processing	This option does <i>not</i> cause the file itself to be saved—that option must be selected separately. It is not required to save the file in order to perform a saveto-database.	
Create a plan file restore point before processing	If selected, then a plan file restore point will be created before processing begins. This option is not selected by default.	
	Restore points can be used to restore plan files to the state they were in before changes were made.	
	<b>NOTE:</b> If the file group uses virtual plan files, this option does not apply. Plan files are not saved and therefore restore points are irrelevant.	

#### **Options for Process with Utilities**

If **Process with Utilities** is the selected processing mode, there are no additional options on the Options tab.

Plan files are not saved when using Process with Utilities, and plan file restore points are not created. When using this mode, the processing is being performed in the utility files, not in the plan files, so it is not necessary to save the plan files. Additionally, in most cases the plan files used with this mode are virtual form-enabled plan files, so the save and restore options are irrelevant.

#### **Options for Update Persistent Plan Files**

If **Update Persistent Plan Files** is the selected processing mode, the following additional option is available on the Options tab:

Option	Description	
Report File	Click the Browse button to select the report file that is configured with the PlanFileReconfig_ControlSheet. This file must be saved in the Reports Library.	
	This control sheet contains the settings that will be applied to plan files during processing.	

Plan files are always saved when using this processing option, and plan file restore points are always created before processing. A save-to-database is not performed in this mode, so if you need to save data, you should process plan files using Normal Processing after you have verified the results of the plan file update.

#### **Options for Process with Custom Utility**

If **Process with Custom Utility** is the selected processing mode, the following additional options are available on the Options tab:

Item	Description	
Report File	Click the <b>Browse</b> button to select the Microsoft Excel spreadsheet file that contains the VBA custom utility. The file must be saved in the Reports Library.	
VBA Module	Select the VBA module to run as part of this utility. The drop-down list shows the VBA modules available in the selected file.	
VBA Function	Select the VBA function to run as part of this utility. The drop-down list shows the VBA functions available in the selected module.	

Plan files are always saved when using this processing option, and plan file restore points are always created before processing. A save-to-database is not performed in this mode, so if you need to save data, you should process plan files using Normal Processing after you have verified the results of the custom utility processing.

#### Plan Files tab

On the Plan Files tab, specify the plan files that you want to process. There are three different options that you can use to specify the plan files: Choose from list, Use filter, and All. You should use the option that corresponds to how many plan files you want to process—all plan files, or a subset of plan files. If you want to process a subset of plan files, you can select individual files to process or you can use a filter to define the subset.

#### **NOTES:**

- If a plan file is locked by another user when the task is executed, then processing for that file will fail. Failures are noted in the result history for the job.
- If a plan file has not yet been created for a particular plan code, then that plan code will not display in this list and will be ignored when processing. Scheduler does not support creating plan files as part of the Process Plan Files task (you must use the separate Create Plan Files task for this purpose).
- If the file group uses a **Show on List** column, then any plan code that is set to **False** will not display in the plan file list and will be ignored when processing.

#### Process all plan files

To process all plan files, select AII. The list of all plan files is generated each time the Scheduler task is executed, so that if new plan files have been added then those new plan files will be included in the processing (the reverse is also true if any plan files have been removed).

Alternatively, you can select **Choose from list** and then select the check box in the column header, causing all current plan codes to be selected. However, in this case the list of selected plan codes is fixed and therefore will not automatically adjust for any future changes.

#### Process selected plan files

To process certain plan files, select **Choose from list**, and then select the check boxes for the plan files that you want to process. When the Scheduler task is executed, Axiom Software will process only the selected plan files.

To find the plan files you are looking for, you can sort, filter, and group the list using standard Axiom grid features. You can show additional columns and hide columns by right-clicking in the column header. If you have filtered the list, you can select the check box in the header to select only the plan files that currently display in the dialog.

**NOTE:** This option is not available when using a file group alias as the selected file group for the task. This is because the alias could change to point to any file group, which could result in a different list of plan files.

#### Process a filtered set of plan files

To use a filter to process a subset of plan files, select **Use Filter**. When the Scheduler task is executed, Axiom Software will process only the plan files that meet the filter.

You can use the Filter Wizard to create the filter, or you can manually type a filter criteria statement into the filter box. The filter must use the plan code table or a lookup table. For example: DEPT.Region='US West' where Dept is the plan code table.

Once you have entered a filter, you can click **Refresh plan file list** to show the plan files that currently match the filter. The refresh feature is intended to help you determine whether you have defined the filter correctly.

You can also use a job variable for the filter. For example, you can define a job variable named "filter" and then place the text {filter} in the filter box. This is intended for use when running Scheduler jobs via RunEvent. If a variable/value pair is specified in the RunEvent function, such as

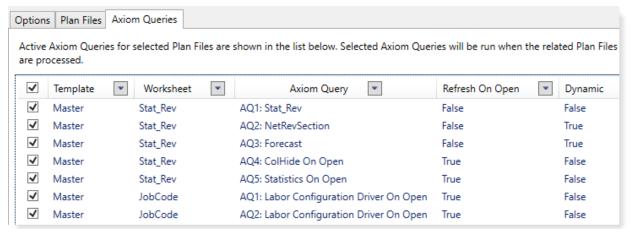
"Filter=dept.region='west'", then that filter statement will replace the {filter} variable and will be used to determine the list of plan files to be processed.

**NOTE:** If you use a variable, and you leave the default value for that variable blank within the **Job Variables** tab, then all plan codes will be processed if no value is passed by the RunEvent function (or if the value is invalid). You may want to define a default filter that results in no values (such as 1=0), so that plan files are only processed if a valid filter value is passed.

#### Axiom Queries

On the **Axiom Queries** tab, select the queries that you want to run in the plan files. By default, all listed queries are selected. This tab only applies when using **Normal Processing** mode.

If you do not want to run a particular query, you can clear the check box. You can select or clear individual check boxes, or you can use the check box in the header to select or clear all queries currently displayed in the list. You can sort, filter, and group the list using standard Axiom grid functionality.



Example Axiom Queries tab

The list of Axiom queries is based on the source templates that were used to create the plan files. Only Axiom queries that meet the following criteria are eligible for selection:

- Active is set to On, or the setting uses a formula.
- Refresh during document processing is set to On.

If a query uses a formula for the Active setting, this means the query is dynamic and may or may not be run, depending on how the formula resolves in each plan file to be processed. When a particular plan file

is processed, each selected query will be evaluated based on the current settings in that plan file. If both **Active** and **Refresh during document processing** are **On** for that plan file, then the query will be run. If either or both settings are **Off** for that plan file, the query will not be run. You can tell whether a query is dynamic or not by looking at the **Dynamic** column in the query list.

If a query is *not* selected on this tab, then that query will not be run in any plan files during processing, regardless of whether Active or Refresh during document processing are enabled in the plan file.

The plan file selection on the Plan Files tab affects the Axiom query list as follows:

- If you have selected individual plan files, then only the eligible queries for the source templates of the selected plan files are shown.
- If you have selected **All** or **Use Filter**, then all eligible queries for all used templates are shown. If the file group has templates that have not been used to create any plan files, then those templates are not included in the list.

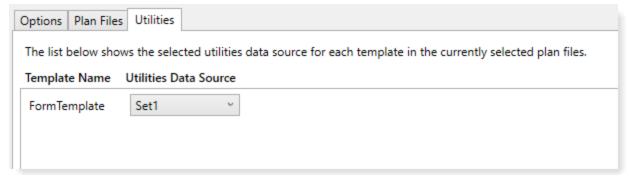
The listed queries are identified by template, worksheet, and query name. The following additional properties are also listed for each query:

- Refresh On Open: Indicates whether the Axiom query is configured to refresh automatically when the file is opened. This is for information purposes only, to help you determine whether the query needs to be included in the processing. The Refresh on Open status is ignored by Process Plan Files—if the query is selected it will be run along with the other selected queries, and if it is not selected it will not be run.
- **Dynamic**: Indicates whether the query is dynamically enabled. True means that the query uses a formula for the **Active** setting.

**NOTE:** If a query is listed on this tab but it is grayed out and unavailable for selection, that means that although the query is active (either directly or dynamically), the query is not eligible to be run using Process Plan Files (because the setting **Refresh during document processing** is set to **Off**). This query is listed for your information only, so that you understand the query cannot be run as part of the process.

#### Utilities tab

On the **Utilities** tab, select the ProcessPlanFileUtilities data source to use during processing. This data source determines which utility files are processed and the processing order. This tab only applies when using **Process with Utilities** mode.



Example Utilities tab

For each template listed, use the **Utilities Data Source** field to select the data source to use for plan files created from that template.

- If the template only has one data source, that data source is selected.
- If the template has multiple data sources, then the data source marked as the default data source is selected by default. If desired, you can use the drop-down list to select a different data source.

When plan files are processed, Axiom Software reads the specified data source in each plan file to determine the utilities to be processed for that plan file.

The plan file selection on the Plan Files tab affects the Utilities list as follows:

- If you have selected individual plan files, then only the templates used to create the selected plan files are shown.
- If you have selected **All** or **Use Filter**, then all used templates are shown. If the file group has templates that have not been used to create any plan files, then those templates are not included in the list.

#### Processing Variables

This tab can be used to define variables to pass into plan files before processing begins, and to pass variables back to the Scheduler job after processing has been performed. This tab is optional and is only used in special situations.

#### **Pre-Processing Document Variables**

This section can be used to pass document variables into plan files before processing. This can impact the processing of plan files if the files are configured to use the variable values in some way.

For each pre-processing document variable, you can specify a variable name and a variable value. The plan files must be set up with GetDocumentInfo functions that return the values for the specified variables.

- \* To add a variable, click the Add button to add a row to the list. Complete the settings for the variable as described below.
- X To remove a variable, select the variable in the list and then click the Remove button. Only one variable can be selected at a time.

To edit the variable settings, double-click the applicable cell to make the cell contents editable. When you are finished editing, you can press the Enter key or Tab key to exit the cell, or click outside of the cell.

Item	Description	
Variable Name	The name of the variable. Do not enclose the variable name in curly brackets (you are not <i>using</i> the variable here, you are defining its value).	
Variable Value	The value of the variable. The value can be a "hard-coded" value, or it can be a job variable that will be resolved at time of processing.	
	If you use a job variable to define the value, the job variable must be enclosed in curly brackets.	

#### **Pre-Processing Workbook Variables**

This section can be used to pass values into plan files before processing. This can impact the processing of plan files if the files are configured to use the values in some way.

For each pre-processing variable, you can specify a workbook location to place the value, and the value to be placed.

Item	Description	
Workbook Location	The location in the workbook for the value to be placed. Any existing value in this location will be overwritten for the duration of the processing. If the file is saved as part of the processing, then the value will be saved in the file.	
	The location can be specified using <code>SheetName!CellRef</code> syntax (for example: Report!A13), or by using a named location in the file.	
Formula	The value to be placed in the specified workbook location. The value can be a "hard-coded" value, or a formula, or a job variable that will be resolved at time of processing.	
	If the value is a formula, the formula is placed into the target cell and calculated in the plan file. The formula can be any formula that would be valid within a spreadsheet in the Axiom client. This includes using Excel functions and Axiom functions. The formula can also use job variables, which will be resolved before placing the formula in the target cell.	

The specified location and value will apply to all plan files being processed by the task. If you are going to use pre-processing variables, the location should be predefined in the template and therefore available to all plan files built using that template. If the plan files will be built using multiple templates, then all templates should be set up with the same designated location, or you should set up separate processing tasks based on template type.

#### **Post-Processing Workbook Variables**

This section can be used to pass a value from plan files back to the Scheduler job after processing has been performed. This can impact the processing of subsequent tasks in the job if those tasks are configured to use the value in some way.

For each post-processing variable, you can specify the location in the workbook to find the value, and the job variable to use that value.

**NOTE:** If this task processes multiple plan files, the resulting variable value will be from the last file that was processed.

Item	Description	
Workbook Location	The location in the workbook to find the value to be passed to Scheduler. This value will become the value for the assigned job variable for the duration of executing the current job (unless a later process within the same job overwrites the value for the same job variable).	
	The location can be specified using SheetName! CellRef syntax (for example: Report!Al3), or by using a named location in the file.	
Job Variable	The job variable that you want to use the value in the specified workbook location. Do not enclose the variable name in curly brackets (you are not <i>using</i> the variable here, you are simply referencing the variable name).	
	If the job variable does not already exist in the job (on the <b>Job Variables</b> tab), then it will be created. However, in most cases you will want the variable to be already set up with a default value, so that the job does not have validation errors that prevent saving.	

The specified location and job variable will apply to all plan files being processed by the task. If you are going to use post-processing variables, the location should be predefined in the template and therefore available to all plan files built using that template. If the plan files will be built using multiple templates, then all templates should be set up with the same designated location, or you should set up separate processing tasks based on template type.

Even though the task may process many plan files, only the job variable value from the last-processed plan file will be used. The plan files must be set up so that all plan files result in the same value after processing, or else your results will vary depending on which plan file was the last file to be processed.

### Process Document List task

This task processes a user-defined set of documents. The process operation always calculates the files. In addition, you can opt to run Axiom queries in the files, process alerts in the files, and then perform a save-to-database and/or save the files.

You can process any Axiom-managed Excel files by using this task. The primary intent of the task is to process files such as driver files or report utilities. For example, you may be using Axiom queries and GetData functions in your driver files that need to be updated regularly. Rather than opening, refreshing, and saving each driver file, you can use this task to define the set of files and schedule processing.

#### **NOTES:**

- Although it is possible to process plan files by using this task, generally plan files should be
  processed by using the Process Plan Files task, which provides more options for processing
  plan file.
- This task does not perform file processing actions on the file. File processing can be set up for report files and driver files, and can be used to perform actions such as file delivery, using standard or multipass processing. If you want to perform file processing using Scheduler, use the File Processing task.

#### Documents to process

Specify the documents to be processed when the task is run. Documents are processed sequentially in the order listed.

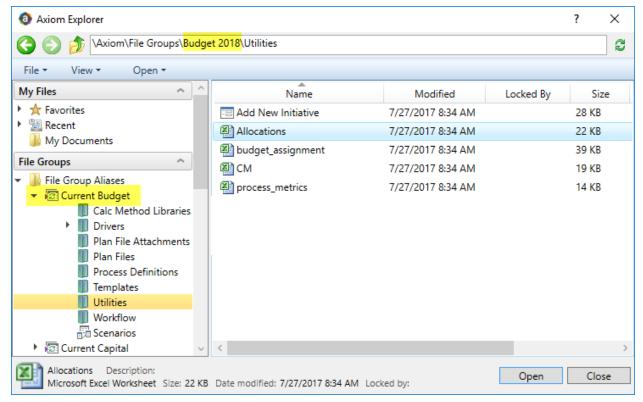
- To add a document, click the Add button. In the **Axiom Explorer** dialog, select the file or files that you want to add, and then click **Open**.
- To remove a document, select the document in the list and then click the Remove button. Only one document can be selected at a time.
- To change the order of documents, select the file in the list and then click the arrow buttons to move the file up or down.

Only Axiom-managed Excel files are valid to be processed in the task.

#### Selecting a document using a file group alias

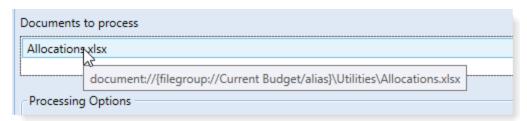
You may want to specify the document to process using a file group alias, so that the task does not have to be manually updated after rolling over to a new file group for a new year of planning. When you are selecting a document in the Axiom Explorer dialog, you can expand the file group alias to see all of the files in the current target of the alias.

For example, in the following screenshot, the file group alias Current Budget is expanded. Because the alias currently points to the file group Budget 2018, the folders and files under the alias are from Budget 2018. When you select a file or folder, you can see the real file path in the top of Axiom Explorer.



Selecting a document to process using a file group alias

When you select a document underneath an alias, the file path is written using alias syntax, so that the Scheduler task will look for the document within the current target of the alias. This path is visible in the tooltip that displays when you hover your cursor over a selected document.



File path using file group alias syntax

When the file group alias is updated to point to a new file group, the Scheduler task will use the file in the new file group automatically. If the file does not exist in the new file group, the task will fail with an error.

### Processing Options

By default, both options are selected. Axiom queries are refreshed before the save-to-database occurs.

If neither option is selected, then the files are calculated and then saved.

Item	Description	
Perform all enabled Axiom Queries in	If this option is selected, then all eligible Axiom queries in all selected files will be refreshed when the task is run. This option is selected by default.	
selected workbooks	<b>NOTE:</b> In order to be eligible for processing, the query must be active, and <b>Refresh during document processing</b> must be enabled.	
Enable iterative calculation while	Specifies whether iterative calculations are enabled for the file during processing. In most cases you will leave this option disabled.	
processing	If this option is selected, then iterative calculations are enabled for the file during the Axiom query refresh process. The iterative calculation settings are fixed at a maximum of 100 iterations and a maximum change value of .001.	
	For more information on iterative calculations, see the Microsoft Excel Help.	
Save document after processing	If selected, then files will be saved after processing. This option is selected by default.	
	This option does <i>not</i> cause a save-to-database to be performed—that option must be selected separately.	
	<b>NOTE:</b> If this option is not selected, then the utility will open the file as read-only and will not attempt to acquire the document lock before processing.	
Run Save To Database on plan files after	If selected, then a save-to-database will be performed after processing. This option is selected by default.	
processing	This option does <i>not</i> cause the file itself to be saved—that option must be selected separately. It is not required to save the file in order to perform a save-to-database.	
Process alerts in selected workbooks	If selected, then alerts in the file will be processed. The file must contain an Alert Control Sheet and one or more alerts must be defined in the file.	
	If Axiom queries are enabled for processing as well, the queries will be run before alerts are processed.	

### Pre-Processing Document Variables

This section can be used to pass document variables into the target files before processing. This can impact processing if the files are configured to use the variable values in some way, such as to filter an Axiom query.

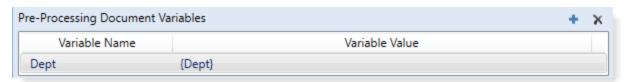
For each pre-processing document variable, you can specify a variable name and a variable value. The target file must be set up with GetDocumentInfo functions that return the values for the specified variables.

- To add a variable, click the Add button to add a row to the list. Complete the settings for the variable as described below.
- To remove a variable, select the variable in the list and then click the Remove button.
  Only one variable can be selected at a time.

To edit the variable settings, double-click the applicable cell to make the cell contents editable. When you are finished editing, you can press the Enter key or Tab key to exit the cell, or click outside of the cell.

Item	Description	
Variable Name	The name of the variable. Do not enclose the variable name in curly brackets (you are not <i>using</i> the variable here, you are defining its value).	
Variable Value	The value of the variable. The value can be a "hard-coded" value, or it can be a job variable that will be resolved at time of processing.	
	If you use a job variable to define the value, the job variable must be enclosed in curly brackets.	

The following example screenshot defines the value for the document variable Dept. The value is defined using the value of a Scheduler job variable {Dept}. A value for Dept can be passed into the Scheduler job when the job is started, and then passed into the target file using the document variable.



#### Pre-Processing Workbook Variables

This section can be used to pass values into the file before processing. This can impact processing if the files are configured to use the values in some way. For each pre-processing variable, you can specify a workbook location to place the value, and the value to be placed.

**NOTE:** The specified value will be placed in all files listed to process. Therefore, unless all files are set up to use the same location and the same way of deriving the value, you will need to create a separate Process Document List task for each file. If you are not using pre-processing variables, then this does not matter.

- To add a variable, click the Add button to add a row to the list. Complete the settings for the variable as described below.
- X To remove a variable, select the variable in the list and then click the Remove button. Only one variable can be selected at a time.
- To change the order of variables, select the variable in the list and then click the arrow buttons to move the variable up or down.

To edit the variable settings, double-click the applicable cell to make the cell contents editable. When you are finished editing, you can press the Enter key or Tab key to exit the cell, or click outside of the cell.

Item	Description	
Workbook The location in the workbook for the value to be placed. Any existing this location will be overwritten for the duration of the processing. saved as part of the processing, the value will be saved in the file.		
	The location can be specified using <code>SheetName!CellRef</code> syntax (for example: Report!A13), or by using a named location in the file.	
Formula	The value to be placed in the specified workbook location. The value can be a "hard-coded" value, or a formula, or a job variable that will be resolved at time of processing.	
	If the value is a formula, the formula is placed into the target cell and calculated in the target file. The formula can be any formula that would be valid within a spreadsheet in the Axiom client. This includes using Excel functions and Axiom functions. The formula can also use job variables, which will be resolved before placing the formula in the target cell.	

#### Post-Processing Workbook Variables

This section can be used to pass a value from the file back to Scheduler after processing has been performed. This can impact the processing of subsequent tasks in the job if those tasks are configured to use the value in some way. For each post-processing variable, you can specify the location in the workbook to find the value, and the job variable to use that value.

**NOTE:** If this task processes multiple documents, the resulting variable value will be from the last document processed.

- To add a variable, click the Add button to add a row to the list. Complete the settings for the variable as described below.
- To remove a variable, select the variable in the list and then click the Remove button.
  Only one variable can be selected at a time.
- To change the order of variables, select the variable in the list and then click the arrow buttons to move the variable up or down.

To edit the variable settings, double-click the applicable cell to make the cell contents editable. When you are finished editing, you can press the Enter key or Tab key to exit the cell, or click outside of the cell.

Item	Description	
Workbook Location	The location in the workbook to find the value to be passed to Scheduler. This value will become the value for the assigned job variable for the duration of executing the current job (unless a later process within the same job overwrites the value for the same job variable).	
	The location can be specified using <code>SheetName!CellRef</code> syntax (for example: Report!A13), or by using a named location in the file.	
Job Variable	The job variable that you want to use the value in the specified workbook location. Do not enclose the variable name in curly brackets (you are not <i>using</i> the variable here, you are simply referencing the variable name).	
	If the job variable does not already exist in the job (on the <b>Job Variables</b> tab), then it will be created when the job is executed. However, in most cases you will want the variable to be already set up with a default value, so that the job does not have validation errors that prevent saving.	

## **Process Template List task**

This task processes a user-defined list of file group templates. During processing, any Axiom queries with **Refresh during template processing** enabled are executed and time-stamped, and then the template files are saved.

The primary purpose of this task is to enable use of *time-stamped Axiom queries* with virtual plan files. Because virtual plan files are re-created from template each time they are accessed, Axiom queries cannot be time-stamped within the plan files. Virtual plan files can use the time stamp from the template, but under normal circumstances, Axiom queries are not time-stamped when they are run in templates. However, when Axiom queries are run during template processing, the Last refresh time for the query is updated, which means that the queries can be configured to only run if the primary table has changed.

To use this task to enable time-stamped Axiom queries for virtual plan files, do the following:

- In the template, enable Refresh only if primary table changed since last refresh and Refresh during template processing for the Axiom queries that you want to be time-stamped.
- In Scheduler, create a job with a **Process Template List** task and add the template to the task. Define a scheduling rule for the job as appropriate. For example, you might want the template to be processed nightly.

When the template is processed, the designated Axiom queries will be run if the primary table has changed, and the time stamps are updated. When a virtual plan file that uses this template is opened, the queries will not be run again if the primary table has not changed.

This task should only be used to process Axiom queries that meet the requirements of time-stamped queries.

#### ► Templates to process

Specify the templates to be processed when the task is run. Templates are processed sequentially in the order listed. If you have multiple templates to process (in the same or different file groups), you can run them all in the same task.

- To add a template, click the Add button. In the **Axiom Explorer** dialog, select the file or files that you want to add, and then click **Open**.
- To remove a template, select the file in the list and then click the Remove button. Only one file can be selected at a time.
- To change the order of templates, select the file in the list and then click the arrow buttons to move the file up or down.

Normal template behavior rules apply during processing. For example, save-to-database and action codes are not run in templates. The only exception to normal template behavior during this task is that any executed Axiom queries will be time stamped.

#### Selecting a template using a file group alias

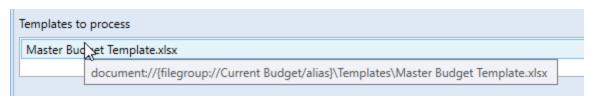
You may want to specify the template to process using a file group alias, so that the task does not have to be manually updated after rolling over to a new file group for a new year of planning. When you are selecting a template in the Axiom Explorer dialog, you can expand the file group alias node to see all of the files in the current target of the alias.

For example, in the following screenshot, the file group alias Current Budget is expanded. Because the alias currently points to the file group Budget 2018, the folders and files under the alias are from Budget 2018. When you select a file or folder, you can see the real file path in the top of Axiom Explorer.



Selecting a template to process using a file group alias

When you select a template underneath an alias, the file path is written using alias syntax, so that the Scheduler task will look for the template within the current target of the alias. This path is visible in the tooltip that displays when you hover your cursor over a selected template.



File path using file group alias syntax

When the file group alias is updated to point to a new file group, the Scheduler task will use the file in the new file group automatically. If the file does not exist in the new file group, the task will fail with an error.

## Purge System Data task

The Purge System Data task is intended to clean up old data in your system, to help keep your system running efficiently.

**NOTE:** Scheduler automatically creates a system job for this task, which you can edit as needed.

This task purges the following data when it is run:

- Scheduler job result history
- Scheduler and system email notifications
- System temp table data
- Audit history
- Alerts

For each category of data, you can specify a number of days of data to keep when the task is run. All results older than the specified number of days will be deleted. Note that 0 days means that no data is purged for that category.

Section	Item	Description
Scheduler Results	Number of days to keep result history	The number of days of job result history to keep when the task is run. By default, this is set to 15 days.
SMTP Messages	Number of days to keep delivered messages and attachment data	The number of days of delivered message data to keep when the task is run. By default, this is set to 15 days.
Temporary Tables	Number of days to keep temp table data	The number of days of temp table data to keep when the task is run. By default, this is set to 15 days.
Audit History	Number of days to keep system history	The number of days of system audit history to keep when the task is run. By default, this is set to 15 days.
		"System history" encompasses all audit data—including prior document versions and deleted documents—except table audit data.
Table History	Number of days to keep table history	The number of days of table audit history to keep when the task is run. By default, this is set to 15 days.
		Table audit data is tracked for tables where Audited is set to True.
Alerts	Number of days to keep alerts	The number of days of alerts to keep when the task is run. By default, this is set to 60 days.

Job variables can be used in all of these settings.

Each purge routine in the task is limited to purging a specific number of rows at a time (50000). If the number of rows to be purged exceeds this limit, then the excess data is retained until the next time the

task is run. If you notice data in the database that you expected to be purged, most likely the amount of data to be purged exceeded the limit, and the data will be purged next time the task is run.

#### Other purged data

This task also cleans up the following items in your system:

- Deleted columns. When a column is deleted from a table in Axiom Software, the column is immediately deleted from the associated view (which prevents it from being accessed in the system), but it remains in the base table. This task finishes the process of removing obsolete columns from the base tables.
- Orphaned user folders. If the system contains any user folders that do not match up with existing users, these folders are deleted. Although user folders are deleted when a user is deleted from security, orphaned user folders can result from other processes, such as migrating a system between different management databases.

These items are not associated with any specific task settings; the delete process is performed whenever the task is executed.

### Raise Event task

The Raise Event task can be used to trigger other Scheduler jobs for execution, using a named event handler. This task has one required setting:

Item	Description
Event Name	Enter the name of the event that you want to raise for execution. This name must match a defined event handler name in one or more other Scheduler jobs.  When this task is run, it looks for any jobs that contain the specified event
	handler name. These jobs are added to the schedule and are eligible to be processed immediately, depending on Scheduler thread availability and any other higher-priority jobs already in the queue.

#### **Event Variables**

This section can be used to pass variables into the jobs triggered by the event handler. If the jobs are configured to use the variables, these values can impact how the jobs are processed.

To add a variable, click the Add button to add a row to the list. Complete the settings for the variable as described below.

To remove a variable, select the variable in the list and then click the Remove button. Only one variable can be selected at a time.

For each variable, you can specify a variable name and a variable value. To edit the variable settings, double-click the applicable cell to make the cell contents editable. When you are finished editing, you can press the Enter key or Tab key to exit the cell, or click outside of the cell.

Item	Description
Variable Name	The name of the variable. Do not enclose the variable name in curly brackets (you are not <i>using</i> the variable here, you are defining its value).
Variable Value	The value of the variable. The value can be a "hard-coded" value, or it can be a job variable that will be resolved at time of processing.
	If you use a job variable to define the value, the job variable must be enclosed in curly brackets.

## **SMTP Message Delivery task**

This task delivers email notifications for Scheduler jobs.

**NOTE:** Scheduler automatically creates a system job for this task, which you can edit as needed. For more information, see Configuring Scheduler system jobs.

Item	Description
Server name	The server name of the SMTP email server.
Port number	The port number for the SMTP email server. By default, the port is 25, but you can specify a different port number if needed.
Server requires authentication	Select this check box if the SMTP email server requires authentication.
	If selected, type a Username and Password.
Test Mode	Specifies whether the task is run in test mode. If this check box is selected, the task verifies that it can successfully connect to the SMTP server to send email notifications, but no emails are actually sent.
	For the System.SMTPMessageDelivery job, new systems are automatically set to test mode. If you restore a database, the restore process also sets the system job to test mode. You must disable test mode before any emails will be sent.

### **Start Process task**

This task starts a process for Process Management. You can use this task to automatically start a process at a specific point in time, including recurring schedules (such as to automatically start a monthly process).

This task can be used to start a generic process definition or a plan file process definition.

Item	Description
Process to start	The process definition to start. Click the <b>Browse</b> button to select the process definition file.
	You can select any process definition in the Process Definition Library or in a file group Process Definitions folder.
Restart process if it is already running	Specifies whether the Scheduler task will restart the process if it is already running, or if the process will be left as is.
	<ul> <li>Select this option if you want to start the target process regardless of whether it is already running. The current process instance will be aborted and a new process instance will start over at step 1. This option is selected by default.</li> </ul>
	<ul> <li>Clear this option if you want to leave the existing process instance running.</li> <li>In this case, the Scheduler task will take no action if the target process is already running.</li> </ul>

## **Update Indexes and Constraints task**

This task is a predefined version of the Execute SQL Command task. It is configured to execute on the **Current System Database**, and it contains a predefined SQL statement, intended to update database indexes and constraints.

This task is included in the System.IndexMaintenance system job to help maintain your database. Under normal circumstances, there should be no need to manually add this task to any other job.

# Index

A	J
Active Directory synchronization	job result history (Scheduler)
import task 65	purging 45
Axiom Software cloud service 49	viewing 44
С	jobs (Scheduler) about 2
Collect Worksheets task 68	creating 11
Create Plan Files	deleting 13
Scheduler task 71	editing 13
E	executing using RunEvent 55 history 44
Echo task 72	managing scheduled jobs 42
event handlers	rescheduling an execution 42
system event handlers 47	results
user-defined 19	purging 45
Execute Command Adapter task 73	viewing 44
Execute SQL Command task 73	running 35
Export ETL Package task 74	stopping an execution 42
exports	system jobs 2
scheduling an export 74	variables 20
F	N
File Processing task 74	notifications
file watcher 51	Scheduler 17
defining folders to watch 51	
setting up an import for file watcher 53	Р
I	parallel processing 23
	Process Document List task 88, 94
Import ETL Package task 77	process management
imports	scheduling processes 99
file watcher, using to execute 51	Process Plan Files task 78
scheduling an import 77	Purge System Data task 96
	purging job results 45

R	properties 63	
Raise Event task 98	renaming 41	
remote data connections	scheduling rules	
creating 49	adding 14	
RunEvent	defining 16	
setting up in Scheduler 55	deleting 17	
setting up in seneduler 55	SMTP Message Delivery task 99	
S	Start Process task 99	
Scheduler 2	U	
about 2		
conditional processing 32	Update Indexes and Constraints task 100	
dialog overview 4	V	
email notifications 8, 17	variables	
enabling services 7	job variables 20	
event handlers 47	job variables 20	
hosted connections 49		
iterative processing 25, 28		
job result history		
purging 8, 45		
viewing 44		
job variables 20		
jobs		
about 2		
managing 11		
running 35		
scheduling rules 14		
settings 36		
system jobs 2		
parallel processing 23		
processing priority 34		
scheduled jobs, managing 42		
servers, managing 45		
setup 8		
tasks		
about 2,63		
conditional processing 32		
managing 11		