

SAS[®] Viya[®] 3.5 Administration: Jobs and Flows

<i>Jobs and Flows: Overview</i>	2
Reasons to Use the Jobs and Flows Feature	2
Sources That Generate Job Requests	3
License Requirement for SAS Job Flow Scheduler	4
<i>Jobs and Flows: Concepts</i>	4
How Jobs and Flows Work	4
Object Definitions	10
Objects in a Job Flow	11
<i>About the Jobs and Flows Page (SAS Environment Manager)</i>	12
Introduction	12
About Tasks and the Group-Managed Service Account Feature	13
Access the Jobs and Flows Page	13
<i>Jobs and Flows: How To (Jobs in SAS Environment Manager)</i>	13
Monitor Jobs	13
Schedule Jobs	18
Manage Jobs	21
<i>Jobs and Flows: How To (Job Flows in SAS Environment Manager)</i>	25
Monitor Job Flows	25
Create a Job Flow and Its Objects	26
Schedule Job Flows	32
Manage Job Flows	34
<i>Jobs and Flows: How To (CLI)</i>	37
Create a Job Request	38
Run a Job Request	39
Schedule a Job Request	39
Create a Job Action for Use in a Job Flow	40
Create a Job Flow and Its Definition	41
Update a Job Flow	43
Trigger a Job Flow with a Manual Trigger	44

Add a Trigger	45
Define a Time-Based Trigger	46
Determine If a Job or Job Flow Completed Successfully	51
Obtain the History of an Executed Job Request or Job Flow	52
Capture Results from a Program or Command	54
Pause the Scheduler Before Migration	56
Adjust for Daylight Saving Time	57
Obtain IDs That Are Used as Input Values	58
Retrieve the Launcher Context Name	59
Construct a URI for Use in a Definition	60
Specify a Filter for Command Output	60
Specify Standard Input	62
Jobs and Flows: Reference (CLI)	63
Considerations	63
Deprecations to the Job CLI Plug-In	63
Definitions Commands	65
Requests Commands	67
Actions Commands	77
Flows Commands	79
Schedulers Commands	89
Webapp Command	95
Job-Definition Definition	95
Job-Action Definition	99
Job-Request Definition	100
Job-Flow Definition	103
Job-Scheduler Definition	110
Time-Based Trigger Definition	111
Unsupported Characters in Job Action and Job Flow Names	116
Using a Group-Managed Service Account	117
Group-Managed Service Account: About	117
Group-Managed Service Account: Prerequisites	120
Group-Managed Service Account: How To	121
Jobs and Flows: Troubleshooting	125
After Adjusting for DST, Schedules Are Still Off by One Hour	125
Command-Line Command Fails in a Job Flow	126
Determine If the Job CLI Plug-In Is Installed	127
The Filter Option on CLI Commands Fails	127
No Service Account Available	127
Error: The job action or job flow name is not valid.	128

Jobs and Flows: Overview

Reasons to Use the Jobs and Flows Feature

The jobs and flows feature enables you to do these tasks:

- Monitor and schedule job requests from a variety of sources on the SAS Viya platform. See [“Sources That Generate Job Requests” on page 3.](#)
- Create and schedule job flows, which can contain multiple job requests (via job actions) and conditions.
- Schedule job requests or job flows to run at a specific time or in response to a specific trigger.
- Edit an existing job request or job flow.
- Create a job request from a SAS DATA step program.

Note: This task can be done on the **Jobs and Flows** page of SAS Environment Manager only.

- View the execution history of a job instance or a job flow.
- Unschedule, delete, and view the properties of a job instance or job flow.

The jobs and flows feature is included in the SAS Viya platform deployment. You can access the jobs and flows functions in the following ways:

- the job command-line interface (CLI) plug-in. See [How To \(CLI\) on page 37.](#)
- the **Jobs and Flows** page of SAS Environment Manager. See [“About the Jobs and Flows Page \(SAS Environment Manager\)” on page 12.](#)

TIP Before using the jobs and flows feature, be sure to review [“Jobs and Flows: Concepts” on page 4.](#)

Sources That Generate Job Requests

The following examples show sources that generate job requests:

- the **Jobs and Flows** page in SAS Environment Manager. See [“Create a Job from a SAS Program” on page 22.](#)
- Job plug-in to the command-line interface (CLI). See [“Create a Job Request” on page 38.](#)
- CAS table state management

Note: Job requests are provided to enable you to manage CAS tables. See [“About Managing CAS Tables with Jobs” on page 21.](#)

- SAS Data Explorer. See [“Create Import Job Requests” in SAS Data Explorer: User’s Guide.](#)
- SAS Data Studio. See [“Creating Jobs for Scheduling” in SAS Data Studio: User’s Guide.](#)
- SAS Studio. See [“Managing Jobs” in SAS Studio Developer’s Guide: Working with Jobs.](#)
- SAS Visual Analytics. See [“Schedule a New Report Distribution” in SAS Visual Analytics: Designing Reports.](#)

Note: Creates job requests that are scheduled in SAS Visual Analytics. You can view and modify the schedules in SAS Environment Manager or via the job CLI plug-in.

- SAS Job Execution Web Application. See [“Overview” in SAS Job Execution Web Application: User’s Guide](#).

After a job request is created, you can schedule, execute, update, and manage the job request by using either of these options:

- the job plug-in to the sas-admin CLI
- the **Jobs and Flows** page of SAS Environment Manager

License Requirement for SAS Job Flow Scheduler

SAS Viya platform provides the ability to work with jobs when using either the job CLI plug-in or SAS Environment Manager. However, if you want to work with job flows, you must have a license for SAS Job Flow Scheduler.

When you have a license, the job flow tasks are available when using either the job CLI plug-in or SAS Environment Manager.

To determine whether you have a license, open SAS Environment Manager and view the **Jobs and Flows** page. See [“Access the Jobs and Flows Page” on page 13](#) for detailed steps.

- If the page is titled **Jobs and Flows**, you have a license for SAS Job Flow Scheduler. You can perform both job and job flow tasks.
- If the page is titled **Jobs**, you do not have a SAS Job Flow Scheduler license. You can perform job tasks only. Contact SAS to purchase a license if you require one.

For information about adding a SAS Job Flow Scheduler license, see [“Licensing: How To Apply Licenses” in SAS Viya Administration: Licensing](#).

Jobs and Flows: Concepts

How Jobs and Flows Work

Terms to Know

The following terms are important to know in this topic:

Job-Flow-Scheduling Service (JFS)

is a microservice. It manages the creation, maintenance, scheduling, and execution of job flows. It does not manage the executions of the jobs themselves.

SAS Job Flow Scheduler (SASJFS)

is a type of scheduler that the Job-Flow-Scheduling Service supports. SAS Job Flow Scheduler manages the microservices that are necessary for executing job flows with or without triggers.

Job Execution Service (JES)

is a microservice. It executes a single instance of a job request each time the Job Execution Service is called. This microservice also creates and manages the job-definition definitions.

SAS Launcher Service

is a microservice. It executes a command-line command (via a job action) each time SAS Launcher Service is called by the Job-Flow-Scheduling Service (JFS). For details, see [“SAS Launcher Server and Launcher Service”](#) in *SAS Viya Administration: Programming Run-Time Servers* and [“Launcher Contexts”](#) in *SAS Viya Administration: Server Contexts*.

Note: A command-line command can fail if the command is not permitted by the SAS Launcher Service allow list. For more information about the allow list feature, see [“Security Considerations”](#) in *SAS Viya Administration: Programming Run-Time Servers*.

Scheduler Service

is a microservice. It manages time-based events that can trigger any object. SAS Job Flow Scheduler uses the Scheduler Service to schedule and trigger a job flow. When a time-based-event condition is met, the Scheduler Service notifies JFS, which in turn notifies SAS Job Flow Scheduler to start the affected job flow.

Note: Other SAS Viya platform services and SAS solutions use the Scheduler Service to schedule their work instances also.

Process Overview

The Microservices Start

- 1 The following microservices start with the SAS Viya platform deployment:
 - Job-Flow-Scheduling Service (JFS)
 - Job Execution Service (JES)
 - Scheduler Service
- 2 When JFS starts, it defines a SAS Job Flow Scheduler scheduler definition and creates the scheduler.

Preparation

- 1 You create a job-definition definition. It contains a program, such as a SAS program, that you want to run.
- 2 You use the job-definition definition to create a job request. A job request has only one definition (either by reference or inclusion). That definition contains only one program. If you want to execute more than one program, you must create a job flow.

6

- 3 You can choose to process the job request in the following ways:
 - Execute the job request immediately. See [“Execute a Job Request” on page 6](#).
 - Schedule the job request to execute at a later time. See [“Schedule a Job Request” on page 7](#).
 - Add the job request to a job flow. See [“Submit a Job Flow” on page 7](#).

TIP Job flows are useful when you want to orchestrate the order in which jobs execute. Jobs can be job requests, command-line commands, and job flows (subflows).

Execute a Job Request

- 1 When you submit a job request for execution, the job request is sent to the Job Execution Service (JES).
- 2 JES passes the program to the provider. Recall that the program is specified in the job-definition definition that is contained in the job-request definition.

The provider is specified by the `type` key-value pair and the `parameters` structure that are also in the job-definition definition. Typically, you use the `Compute` provider. Compute is used to run SAS programs.

Note:

- `Compute` is the provider intended for use by users. Other providers are used by software only.
- The design of a provider can vary depending on the specific type of resource associated with the provider. How a provider executes a job request also varies depending on the provider. Many providers are microservices, but some are not.

-
- 3 The `Compute` provider starts processing the job request.

The provider calls the SAS Launcher Service, which in turn starts a `Compute` server and a `Compute` session. Then, the provider sends the program to the `Compute` server that executes the SAS program in the `Compute` session. For details, see steps 3 through 5 in [“How It Works” in SAS Viya Administration: Programming Run-Time Servers](#).

Note: Step 1 in the diagram “How a Client Submits Code to the SAS Compute Server” refers to an OAUTH token. When you use the job CLI plug-in, the `sas-admin` CLI provides security by requiring you to authenticate.

-
- 4 When the program completes, the provider updates the history status of the job-request instance and passes it to JES.

Note: Because you can reuse a job request, each time you execute a job request it is called a job-request instance and has its own job-request-instance ID.

The history status includes the following information:

- the results from the program in the job-request instance
- the completion state of the job-request instance
- a link to the log from the provider

All of this information and the associated job request are stored by the SAS Viya platform.

Note: When the job request is part of a job flow, the job-request-instance ID is returned to the job flow. You can use this ID to locate the stored information about the job-request instance.

Note: SAS Environment Manager does not send the job request to the Job Execution Service (JES). Instead, SAS Environment Manager sends the job request to the Scheduler Service with instructions to schedule the job request for immediate execution. Then, the Scheduler Service calls the JES to execute the job request.

Schedule a Job Request

- 1 You can schedule a job request by using the `job requests schedule` command. This command requires that you provide trigger information for the schedule.
- 2 When you schedule a job request, the job request is sent to the Scheduler Service. The Scheduler Service stores the schedule trigger information and the URI of the associated job request.
- 3 When the trigger conditions are met, the Scheduler Service notifies the Job Execution Service to process the associated job request. See step 2 in [“Execute a Job Request” on page 6](#).

Submit a Job Flow

- 1 To use a job request in a job flow, you first must create a job-action definition that contains the job request. You can also create a job action that contains a fully qualified command-line command. A job action can contain only one object (job request or command), but you can specify multiple job actions in a job flow.

For details, see [“How Job Actions Work in a Job Flow” on page 9](#).

Note:

- A command-line command can fail if the command is not permitted by the SAS Launcher Service allow list. For more information about the allow list feature, see [“Security Considerations” in SAS Viya Administration: Programming Run-Time Servers](#).
 - SAS Launcher Service does not validate whether the command is a valid command to run in the pod. It is the responsibility of the creator of the job action that contains the command to ensure that the command is valid to run.
-

- 2 You create a job-flow definition. This definition can contain some or all of the following information:
 - The job action that you created in the previous step. You can specify multiple job actions and job flows (subflows). Remember that job actions can contain either a job request or a command.

- Schedule trigger information. When you want to trigger the job flow with a time event, specify event and include the time-event-trigger definition. You can also specify how to evaluate the conditions associated with the trigger.

Note: When you want to execute the job flow immediately, you do not specify any trigger information. However, if you define a job flow with trigger information, you can still run it immediately by using the `job flows execute` command. It ignores any trigger information contained in the job-flow definition.

- Several optional properties are available. Some of these properties can control how the job flow or its objects can end processing. See [“Controlling How a Job Flow or Its Objects End” on page 9](#).
- 3 Submit the job flow. The job flow is sent to the Job-Flow-Scheduling Service (JFS).
You can submit a job flow to be executed in the following ways:
 - Execute the flow immediately. See [“Execute a Job Flow Immediately” on page 8](#).
 - Schedule the flow to execute at a later time. A time-event trigger starts the flow. See [“Schedule a Job Flow with a Time-Event Trigger” on page 8](#).

Execute a Job Flow Immediately

- 1 The Job-Flow-Scheduling Service (JFS) sends the job flow to the associated scheduler, SAS Job Flow Scheduler.
- 2 SAS Job Flow Scheduler begins by processing the first object in the job flow. If the object has dependencies, the object is not processed until its dependencies are met.
 - When a flow object is a job request, it is sent to the Job Execution Service (JES). See steps 2 through 4 of [“Execute a Job Request” on page 6](#).
 - When a flow object is a command-line command, it is sent to SAS Launcher Service.

Note: SAS Job Flow Scheduler also sends the launcher context that is provided in the scheduler definition.

 - When a flow object is a subflow, SAS Job Flow Scheduler processes it in the same way as the primary job flow.
- 3 SAS Job Flow Scheduler processes each subsequent flow object depending on these criteria:
 - its order in the job flow
 - whether any specified dependencies are met
- 4 When the job flow is complete, all of its information is stored and SAS Job Flow Scheduler returns the job flow output to Job-Flow-Scheduling Service.

Schedule a Job Flow with a Time-Event Trigger

- 1 The Job-Flow-Scheduling Service (JFS) sends the job flow to the associated scheduler, SAS Job Flow Scheduler.

- 2 SAS Job Flow Scheduler sends the job flow to the Scheduler Service.
- 3 The Scheduler Service stores the time-event trigger and the URI of the associated job flow.
- 4 When the time-event-trigger conditions are met, the Scheduler Service notifies SAS Job Flow Scheduler. See step 2 in [“Execute a Job Flow Immediately” on page 8](#).

How Job Actions Work in a Job Flow

Before you can use a job request or a command-line command in a job flow, it must be specified in a *job-action definition*. A job-action definition can contain only a job request or a command-line command. If you want to include both a job request and a command in a job flow, then you must create a job-action definition for each.

The job-action definition is a container that groups the job request or the command with properties that are specific to the scheduler that is associated with the job flow. Neither the job request nor the command need to know about these properties, but they are affected by them because the scheduler uses the properties only for that job request or command in the applicable job action.

The job flow, Job-Flow-Scheduling Service (JFS), and the Scheduler Service do not know (and do not need to know) any of the information contained in the job action. Neither the job request nor the command-line command are processed by JFS or the Scheduler Service:

- Job requests are processed by Job Execution Service (JES).

For details, see [“Execute a Job Request” on page 6](#).

- Commands are processed by SAS Launcher Service.

When a job action contains a command-line command, that command is passed to SAS Launcher Service, which in turn executes the command. The associated scheduler definition contains a context. The Launcher uses that context to run the command. For details, see [“Launcher Contexts” in SAS Viya Administration: Server Contexts](#).

Note: A command-line command can fail if the command is not permitted by the SAS Launcher Service allow list. For more information about the allow list feature, see [“Security Considerations” in SAS Viya Administration: Programming Run-Time Servers](#).

Controlling How a Job Flow or Its Objects End

The job-flow definition contains the `flowProperties` properties. By using some of these properties, you can control how the following objects end processing:

- job actions in the job flow
- subflows in the job flow
- the job flow itself

You can use one or more of the following properties, alone or in combination, to control the end of the job flow and its objects:

- `completionLevel`

- completionSelectedJobs
- endBehavior
- exitcodeStrategy

For detailed descriptions of these properties and their available settings, see “[flowProperties](#)” on [page 108](#).

CAUTION

Use care when specifying these properties and combinations of their settings. It is possible to cause unintended consequences, such as a job flow that remains in a running state forever.

Object Definitions

Job definition

Contains information about a job such as the code, the job type, and job parameters. A job definition is identified by its job-definition ID.

Job request

Contains a job definition plus runtime information (such as values for prompts and the compute context). Job requests enable job definitions to be scheduled and executed.

- An executed job request is identified by its job-request ID.
- A scheduled job request is identified by its scheduled-job-request ID.

Job flow

Contains job actions, job flows, conditions, and dependencies that can be scheduled as a group. A job flow is identified by its job-flow ID.

Job-flow instance

A job flow can run multiple times. Each time a job flow is scheduled or executed, a job-flow instance is created. A job-flow instance is identified by its job-flow-instance ID.

Job action

Created when you add a job request or a command-line action to a flow. A job action is identified by its action ID.

Job-request instance

Job requests can run multiple times. Each time a job request is scheduled or executed, a job-request instance is created. A job-request instance is identified by its job-request-instance ID.

Trigger

Contains a time specification that is used to schedule a job request or a flow.

Objects in a Job Flow

A job flow is a group of jobs, dependencies, and conditions that are organized in a sequence and can be scheduled as a group. Job flows contain objects (such as jobs, gates, and events) and connections, which define the order in which objects are evaluated in the flow.

A job flow can contain any of these objects:

Job action

Any job that is available for scheduling. A flow can contain multiple jobs, but a job can be included only once in a flow.

Job flow

A previously created job flow.

AND gate

A conditional step that is met only if all of the input conditions are met. For example, if Job A and Job B are inputs to an AND gate and Job C is the output, Job C runs only when Job A and Job B successfully complete.

TIP The **Jobs and Flows** page in SAS Environment Manager uses **All** instead of AND.

OR gate

A conditional step that is met if any of the input conditions are met. For example, if Job A and Job B are inputs to an OR gate and Job C is the output, Job C runs when either Job A or Job B successfully completes.

TIP The **Jobs and Flows** page in SAS Environment Manager uses **Any** instead of OR.

Time event

A trigger that occurs at a specified time or on a specified schedule. The event is true each time that its conditions are met. The event triggers the job flow each time it is true as long as an AND gate does not evaluate this event with other triggers.

If the time event is a dependency to a job action, then the job action is executed only the first time it is true.

Note: When an AND gate evaluates a time event with other triggers, the `duration` parameter can further affect the time event. The `duration` option specifies the number of minutes that the time event stays true. The default is 1 minute.

Command-line action

A command that is sent to the operating system command line and executed as a step in the flow.

Object connections

After you add objects to the flow, you make connections to specify the order and conditions under which the objects are evaluated in the flow. Connecting two jobs creates a job dependency. The execution of the second job depends on the completion of the first job.

You can choose how to define the completion of a flow:

- All of the objects and actions in the flow complete successfully or any object in the flow fails.
- A selected item in the flow ends.

About the Jobs and Flows Page (SAS Environment Manager)

Introduction

Note: For concepts and background, see [“Jobs and Flows: Concepts” on page 4](#).

The **Jobs and Flows** page  provides two sets of tasks: jobs and flows.

If the SAS Environment Manager page is titled **Jobs**, you can perform the following tasks for jobs:

- schedule jobs to run at a specific time or in response to a specific trigger
- create a job from a SAS DATA step program
- view the execution history of a job
- unschedule, delete, and view the properties of a job

If the SAS Environment Manager page is titled **Jobs and Flows**, you can perform the jobs tasks and the following tasks for job flows:

- create a new job flow or edit an existing job flow
- schedule job flows to run at a specific time or in response to a specific trigger
- view the execution history of a job flow
- unschedule, delete, and view the properties of a job flow

Note:

The flow tasks require a license. To determine whether you have a license, see [“License Requirement for SAS Job Flow Scheduler” on page 4](#).

About Tasks and the Group-Managed Service Account Feature


When the applicable hot fixes are applied, the group-managed service account feature is available. The Jobs and Flows page enables members associated with a group-managed service account to perform tasks by using the service account. For more information, see [“Using a Group-Managed Service Account” on page 117](#).

- Members of a group-managed service account can view information about job requests and job flows that are associated with the group-managed service account. This information includes the **Monitor** table and the **Jobs and Flows** table.

TIP The **Monitor** table provides the **View as** list from which a group member can select the group-managed service account.

- A group of users can share the ability to schedule, reschedule, and unschedule a job request or job flow by being members of a group with access to a group-managed service account.

Access the Jobs and Flows Page

- 1 Open SAS Environment Manager. In the applications menu (☰), under **ADMINISTRATION**, select **Manage Environment**.
- 2 In the vertical navigation bar, select .

Jobs and Flows: How To (Jobs in SAS Environment Manager)

Monitor Jobs

Note: When the applicable hot fixes are applied, the group-managed service account feature is available. The Jobs and Flows page enables members associated with a group-managed service

account to perform tasks by using the service account. For more information, see [“Using a Group-Managed Service Account” on page 117](#).

View a Table of Job Executions

TIP If you are a group member associated with a group-managed service account, you can view the execution history of the jobs that have run via that group-managed service account. Select the **View as** list. Then select the group-managed service account.

By default, when you open the **Monitoring** tab, the **Monitor** table displays a list of all jobs that have executed in the previous 24 hours.

The table displays the following information by default:

- The job name.
- The start date and time.

Note: If you change to a different time zone, the new time zone is not automatically reflected in the **Monitor** table. Close and reopen your browser to use the new time zone in the **Monitor** table.

- The status.

Note: If a job did not complete successfully, the message **Failed** appears in the **Status** column. Click the **Failed** message to view information (if available) about the reason for the failure.

- The user that submitted the job.

Note: If the user submitted the job via a group-managed service account, that service account is shown in this column.

The group-managed service account option is available after the applicable hot fixes have been applied.

- If the job has completed, the table also includes a link to download the log, if one was created. For more information, see [“Viewing the Job Log” on page 17](#).

Note: A job can run via a group-managed service account.

The group-managed service account option is available after the applicable hot fixes have been applied.


The following information is not displayed by default. For information about changing the columns that are displayed, see [“Work with Information Displayed in Tables” in SAS Viya Administration: Using SAS Environment Manager](#).

- The end date and time.

Note: If you change to a different time zone, the new time zone is not automatically reflected in the **Monitor** table. Close and reopen your browser to use the new time zone in the **Monitor** table.

- The run time.
- The environment in which the job ran.


View a Chart of Job Executions

From the **Monitoring** tab, click  to display a chart of the jobs that have executed in the selected time period (the default is the previous 24 hours).

- The sliders below the graph enable you to zoom into a specific time window within the selected time period.
- Jobs that ran successfully are displayed in green. Jobs that failed are displayed in red.
- Place your cursor over a bar in the chart to display the name, start time, and status of the job.

View the Execution History for a Job

You can view information about previous runs of a job that is available for scheduling.

- 1 From the **Scheduling** tab, select a job in the **Jobs and Flows** table.
- 2 Click  in the toolbar or select **Execution history** in the pop-up menu.
- 3 Information about previous runs of the selected job is displayed in the **Monitoring** tab.

View Scheduled Jobs

By default, when you open the **Scheduling** tab, the **Jobs and Flows** table displays the following information:

- The job name.
- The scheduled status.
- The description.
- The date on which the job was created.

The following information is not displayed by default. For information about changing the columns that are displayed, see [“Work with Information Displayed in Tables” in SAS Viya Administration: Using SAS Environment Manager](#).


Note: Some of these columns apply to only job requests or only scheduled jobs.

- The ID of the user that created the job.

Note: Jobs for report distributions from SAS Visual Analytics are scheduled using a different user ID than the user ID under which they were created. The jobs are created under the user ID `sas.reportDistribution`, but they are scheduled under the user ID `sas.scheduler`.

- The date on which the job was last modified.
- The ID of the user who last modified the job.
- The job ID.
- The scheduled job ID.
- The job type.

View a Chart of Scheduled Jobs

From the **Scheduling** tab, click  to display a chart of the jobs that are scheduled over a selected time period.

- The default time period is one year, and all scheduled jobs are shown.
- Each scheduled job is listed on a separate line in the graph.
- Bars in the chart represent each scheduled execution of a job. Hover over a bar to display the job name, the status, and the date and time of the scheduled execution.
- The sliders below the graph enable you to zoom into a specific time window within the selected time period.

Filter Details about Jobs

You can specify filters to narrow the jobs that are displayed in the **Monitoring** tab. For example, you can specify that only jobs that failed or only jobs that were created by a specific user are displayed.

Complete these steps in the **Jobs Filter** pane to the left of the table:

- 1 To filter by name, enter text into the **Filter by name** field. The results display only those jobs in the displayed list that contain the specified text.

Note:

The **Filter by name** field filters only the currently displayed jobs. For example, you might have 100 jobs of which 10 contain the word “test” in their names. If the current page contains 3 jobs that have “test” in their names, specifying **test** in the **Filter by name** field causes only the 3 jobs from the current page to be listed rather than all 10 jobs in the complete list. Using the specified filter, you must navigate to different pages in order to see all the filtered jobs.

- 2 To filter by job status, select one or more check boxes in the **Status** list that you want to display.
 - 3 To filter by creator, select one or more check boxes in the **Created By** list. You can enter text in the **Filter** text box to find an existing creator or to specify a creator.
-

Note:

- You can filter by creator if you meet the following criteria:
 - You are an administrator.
 - You opted in to the SAS Administrators group when you signed in to SAS Environment Manager.
- When you enter text for the filter, do not enter both a single quotation mark and a double quotation mark. Doing so causes an error when the filter is applied.

-
- 4 After you have selected all the filters that you want to use, click **Apply**. The filters affect the jobs that are displayed in both the table of jobs and the jobs bar chart.
 - 5 To remove a filter, deselect its check box and click **Apply**.
 - 6 To remove all filters in either the **Status** or **Created By** list, click |↶ next to the list.
To remove all filters, click |↶ next to **Jobs Filter**.

Viewing the Job Log

About the Job Log

If the job execution component for a job generated a log, you can download the log file for further analysis.

Not all jobs create a log, and a log is not always produced when an error occurs. For example, the following jobs or situations do not produce a log:

- backup jobs
- when a time-out occurs
- when the source data cannot be found for a data plan

Note: A job can run via a group-managed service account.

The group-managed service account option is available after the applicable hot fixes have been applied.

View the Job Log

In the **Log** column of the **Monitor** table, click **Download** to save or open a local copy of the log file.

Note: The specific behavior depends on your browser.

Schedule Jobs


Note: When the applicable hot fixes are applied, the group-managed service account feature is available. The Jobs and Flows page enables members associated with a group-managed service account to perform tasks by using the service account. For more information, see [“Using a Group-Managed Service Account” on page 117](#).



Schedule a Job

Note: Scheduled jobs from SAS applications are not dynamically updated. If the SAS code of a scheduled job changes, the job is not updated automatically to use the changed code. In order to use the new code, you must unschedule the existing scheduled job and then schedule the job with the revised code.

Complete these steps to create a schedule for a job request. The schedule includes a trigger that controls when the job runs.

- 1 From the **Scheduling** tab, select a job in the **Jobs and Flows** table.

TIP If the table is not displayed, click .

- 2 Click  in the toolbar or select **Schedule** from the pop-up menu.
- 3 (Optional) By default, the schedule runs using your account. In the **Run as** field, you can choose to run it using another account. Click  to select from available accounts.

(Default) Current user

Runs using your own user account.

A group-managed service account

Runs using a group-managed service account. You must be a member of a group that has been granted access to the service account credentials.

Note: The group-managed service account option is available after the applicable hot fixes have been applied. When the SAS Viya environment is updated, the group-managed service account function becomes available and the prior ability to use an individual's account other than your own is no longer available. For more information, see [“Group-Managed Service Account: About” on page 117](#).

An individual's user ID other than your own

Runs using an individual's account other than your own. The user accounts available for selection must have signed in to SAS since it was installed. This option is for SAS administrators only.

Note: If the applicable hot fixes have been applied, this option is no longer available. For more information, see [“Group-Managed Service Account: About” on page 117](#).


- 4 Select one or more triggers. A trigger controls when the job runs. In the **Available triggers** table, for each of the triggers that you want to use, select the **Enabled** slider control.

To create a new trigger, see [“Create a Time-Based Trigger for a Job Schedule” on page 19](#).

Note:

- You can use a trigger with only the job for which it was created.
 - **Time Event** is the only trigger type supported.
-

- 5 Click **Save**.

The  appears in the **Scheduled** column for the job listing in the **Jobs and Flows** table.

Create a Time-Based Trigger for a Job Schedule

- 1 In the Schedule Job window, click **+** above the **Available triggers** table.
- 2 In the New Trigger window, assign a name to the new trigger. By default, the name is set to **New trigger**.
- 3 In the **Frequency** field, specify how often to repeat the trigger (such as a specified number of minutes, hours, or days).
- 4 Depending on your selection of the frequency interval, additional fields appear in the window to enable you to completely specify a frequency for the trigger. Use these fields to specify the criteria for the trigger interval.

Here are examples of frequency-interval criteria:

- Select **Yearly** in the **Frequency** field. You can specify a day of a month (such as the first of January), the last day of a month, or a specific weekday in a month (such as the third Thursday in February).
 - Select **Minutes** in the **Frequency** field. You can specify that the job runs every 5, 10, 15, 20, or 30 minutes.
-

Note: If you select **Date List** in the **Frequency** field, you cannot select a date more than once.

- 5 In the **Start time** or **Offset** field, specify when to start the schedule. Click the field to select a time. Times are specified in 24-hour format.
-

Note: When you select **Minutes** or **Hours** in the **Frequency** field, the **Offset** field is displayed. The field specifies an offset from midnight.

When you select any other frequency, the **Start time** field is displayed.

Note:

If a hot fix is installed, when you select **Minutes** or **Hourly** in the **Frequency** field, the **Start time** field is now called **Offset**.

TIP For examples, see “[Example: How to Specify the Start Time or Offset](#)” on page 20.

- 6 In the **Time zone** field, specify the time zone to use when evaluating the time for the trigger and the date on which the trigger starts.

Note: If you choose **Date List** in the **Frequency** field, you must select the same value in the **Time zone** field for every scheduled date.

- 7 In the **Start date** field, select the date on which the trigger starts.

Note: When you select the **Date List** frequency, the **End** field is not displayed.

- 8 In the **End** field, specify when the trigger ends. You can specify that the trigger never ends, that it ends after a certain number of times, or that it ends on a specific date.

Note: When you select the **Date List** frequency, the **End** field is not displayed.

- 9 Click **Save**.

Example: How to Specify the Start Time or Offset

Note:

If a hot fix is installed, when you select **Minutes** or **Hourly** in the **Frequency** field, the **Start time** field is now called **Offset**.

Here are some examples of specifying the **Start time** or **Offset**:

Frequency	Start Time or Offset	Result
Every hour	10:15	The job or job flow runs at 10:15, 11:15, 12:15, and so on.
Every hour	00:10	The job or job flow runs at 12:10, 1:10, 2:10, and so on.
Every 20 minutes	09	The job or job flow runs at 9:00, 9:20, 9:40, and so on.

Frequency	Start Time or Offset	Result
Every 5 minutes	00	<p>Scenario 1: The schedule is saved before 10:00. The job or job flow runs at 10:00, 10:05, 10:10, 10:15, 10:20, 10:25, 10:30, and so on.</p> <p>Scenario 2: The schedule is saved at 10:27. All trigger time points before 10:27 do not run. (That is, 10:00, 10:05, 10:10, 10:15, 10:20, 10:25 do not run.) Instead, the job or job flow starts at the current time of 10:27. Therefore, it runs at 10:30, 10:35, 10:40, 10:45, and so on.</p>

Manage Jobs

Note: When the applicable hot fixes are applied, the group-managed service account feature is available. The Jobs and Flows page enables members associated with a group-managed service account to perform tasks by using the service account. For more information, see [“Using a Group-Managed Service Account” on page 117](#).

About Managing CAS Tables with Jobs

Three jobs are provided to manage CAS tables:

- Import cas-shared-default Public data
- Load cas-shared-default Public data
- Unload cas-shared-default Public data

When working with these jobs, remember these key points:

- You can schedule these jobs, but you cannot delete them.
- You can modify the job options only on copies of the jobs.
- If you schedule one of these jobs and then make a copy of the job, only the job is copied, not any triggers that are associated with the job.

For more information about these jobs, see [“CAS Table State Management” in SAS Viya Administration: Data](#).

Create a Job from a SAS Program

You can create a job from a SAS program.

- 1 In the **Scheduling** tab, select **New** ⇒ **Job**.
- 2 In the New Job window, specify a name for the job.
- 3 In the **SAS DATA step program** field, navigate to the program that you want to schedule, select it, and click **OK**.
- 4 Click **OK** in the New Job window.

The new job appears in the list of jobs that can be scheduled and in the list of jobs that you can include in a job flow.

Run a Job from the Scheduling Tab

You can run a job regardless of whether it has been scheduled.

- 1 From the **Scheduling** tab, select a job in the **Jobs and Flows** table.
- 2 To run the job under your own credentials, click ▷ in the toolbar or select **Run** from the pop-up menu.
- 3 (When applicable hot fixes are installed) To run the job using a group-managed service account, complete these steps:
 - a Select **Run as** from the pop-up menu.
 - b The Choose Identity window appears. Select the group-managed service account of which you are a member.
- 4 (When applicable hot fixes are not installed) To run the job using an account other than your own, complete these steps:
 - a Select **Run as** from the pop-up menu.
 - b The Choose Identity window appears. Select the account under which to run the job.

.....
Note: The user ID that you select must have previously signed in to SAS.


- 5 Click **OK**.

Rerun a Job from the Monitoring Tab

- 1 Select the job in the **Monitor** table.
- 2 Click ▷ in the toolbar or select **Execute** from the pop-up menu.

A copy of the job is created and is displayed in the list.

View Job Properties

- 1 From the **Scheduling** tab, select a job in the **Jobs and Flows** table.
- 2 Click  in the toolbar or select **Properties** from the pop-up menu.



Note: The information in the Job properties window is read-only.


Edit a Scheduled Job

After a job is scheduled, you can edit the schedule for the job.

- 1 From the **Scheduling** tab, select a scheduled job () in the **Jobs and Flows** table.

TIP The following indicators appear in the **Scheduled** column:

-  indicates a scheduled job that contains at least one enabled trigger.
-  indicates a scheduled job that contains disabled triggers.

- 2 To modify the schedule for the job, click  from the toolbar or select **Edit Schedule** from the pop-up menu.
- 3 In the Edit Schedule window, you can add, edit, and remove triggers for the job schedule.
- 4 Click **Save** after you have finished modifying the schedule.



Disabling or Uncheduling the Schedule for a Job

Choosing between Disable and Unschedule

To prevent a job from running its specified schedule, you can either disable the triggers or unschedule the job.



- Disabling the triggers prevents the job from running the schedule and preserves the defined triggers.
- Uncheduling the job prevents the job from running on the defined schedule.

Disable the Triggers

- 1 From the **Scheduling** tab, select a scheduled job () in the **Jobs and Flows** table.
- 2 Click  from the toolbar or select **Edit Schedule** from the pop-up menu.

- 3 In the **Enabled** column of the **Available triggers** table, disable all of the slider controls.

TIP



- If you disable all triggers for a job,  appears in the **Scheduled** column.
- If any of the triggers for the job are enabled,  appears in the column.

Unschedule a Job

CAUTION


When you unschedule a job, any enabled triggers that are associated with the job are deleted. To unschedule a job and retain the triggers, see [“Disable the Triggers” on page 23](#).

Note: A job that was scheduled via a group-managed service account can be unscheduled only by users with access to the same group-managed service account.

- 1 From the **Scheduling** tab, select a scheduled job () in the **Jobs and Flows** table.
- 2 Click  from the toolbar or select **Unschedule** from the pop-up menu.

Delete a Job from the Scheduling Tab


Jobs remain in the **Jobs and Flows** table until you delete them.

- 1 From the **Scheduling** tab, select a job in the **Jobs and Flows** table.
- 2 Click  in the toolbar or select **Delete** from the pop-up menu.

Note: You cannot delete any of the provided CAS table state management jobs (Import cas-shared-default Public data, Load cas-shared-default Public data, and Unload cas-shared-default Public data).

Delete a Job from the Monitoring Tab

Details about jobs remain in the **Monitor** table unless you delete the entry.

- 1 From the **Monitoring** tab, select a job execution entry in the **Monitor** table.
- 2 Click  on the toolbar and select **Delete** from the menu. You can also select **Delete** from the pop-up menu.

Jobs and Flows: How To (Job Flows in SAS Environment Manager)

Use these tasks in SAS Environment Manager to work with job flows.

Monitor Job Flows

Note: When the applicable hot fixes are applied, the group-managed service account feature is available. The Jobs and Flows page enables members associated with a group-managed service account to perform tasks by using the service account. For more information, see [“Using a Group-Managed Service Account” on page 117](#).

View Job Flows by User Account


TIP This task is useful for non-administrators who want to view the execution histories of job requests within job flows associated with a group-managed service account.

- 1 Select the **Monitoring** tab on the Jobs and Flows page.
- 2 Above the **Monitor** table, select the **View as** list.

Note: This list is displayed for non-administrators.

- 3 Select a user account from the list.

View the Execution History for a Job Flow and Status of Its Objects

- 1 From the **Scheduling** tab, select a job flow in the **Jobs and Flows** table.
- 2 Click  from the toolbar or select **Execution history** from the pop-up menu.

The **Flow_name Execution History** window displays the results for the specified job flow. By default, the window displays the 20 most recent executions of the job flow. However, the entries that are displayed depend on several factors.

- If the job flow was executed as a result of a schedule, the window displays all instances of the job flow's scheduled executions.
 - If the job flow was executed from the **Execute** command, the window displays only the most recent instance of the job flow's execution.
 - If the job flow was executed both by a schedule and the **Execute** command, the window displays only scheduled executions.
- 3 You can filter on the execution status and change the time period that is displayed (the default is the previous 24 hours).
 - 4 Expand the entry for an execution instance to view the execution status of the objects in the job flow. The status is displayed for these objects:
 - jobs that have executed
 - jobs that are running
 - jobs that are marked for execution
 - subflows (job flows that are included in the flow), regardless of whether they execute
 - command-line actions

The status is not displayed for these objects:

- jobs that cannot execute because their dependencies are not met and therefore are not included
- gates
- time events

Note: The execution history is displayed for only the first ten objects that are marked for execution in a job flow.

Create a Job Flow and Its Objects

See [“Objects in a Job Flow” on page 11](#) for more information.

Create a Job Flow

Note: Non-administrative users can view job flows that are created by other users.

However, non-administrative users cannot open, edit, or schedule those job flows. They cannot change or un-schedule scheduled job flows.

- 1 From the **Scheduling** tab, select **New** ⇒ **Job flow** to display the New Job Flow window.
- 2 In the left pane, select objects that you want to add to the job flow and drag them to the job-flow editor in the center pane. The time-event and command-line action objects require further definition:

- “Create a Time Event” on page 27
 - “Add a Command-Line Action” on page 29
- 3 As you add objects to the job flow, make connections between the objects in order to specify the sequence of the job flow.
 - a Click on the right side of the first object in the order.
 - b Drag a line to the left side of the next object in the order.


Remember these key points when you make connections:

 - Each object (other than gates) can have only one input connection and one output connection.
 - Objects change positions in the window automatically as you make connections.
 - By default, a connection from a job to another object (a job dependency) specifies that the connected object runs when the job it is connected to completes successfully. You can change the job-dependency criteria. See [“Modifying a Job Dependency” on page 31](#).
 - A connection from a time event to a job specifies that the job runs only the first time that the time-event conditions are met. If you specify a repeating time condition, the connected job does not run every time that the time-event conditions are met. To create a time event, see [“Create a Time Event” on page 27](#).
 - 4 In the right pane, you can modify properties and attributes for the job flow and its objects.

The job-flow attributes include the criteria that determine whether the job flow is complete. You can change that criteria. See [“Modify Job Flow Properties and Attributes” on page 31](#).

See also [“Modify Object Properties” on page 32](#).
 - 5 When the job flow is defined, click **Save**.

Create a Time Event

- 1 In the New Job Flow window, expand **Events** in the object tree and drag a **Time** entry to the job-flow editor in the center pane.
- 2 Select the time event in the editor and click  in the vertical toolbar on the right side of the editor.
- 3 In the **Time Event Properties** pane, assign a name to the event.

TIP By default, the name is set to **Time Event n**.

- 4 Use the **Frequency** field to specify how often to repeat the event (such as a specified number of minutes, hours, or days).

Note: The time event is true every time the time-event conditions are met. However, a job that is dependent on the time event executes only the first time that the conditions are met.

- Depending on your selection for the frequency interval, additional fields appear in the window to enable you to completely specify a frequency for the event. Use these fields to specify the criteria for the event interval.

TIP Use the scroll bar to view all of the fields in the **Time Event Properties** pane.

Here are examples of frequency-interval criteria:

- Select **Yearly** in the **Frequency** field. You can specify a day of a month (such as the first of January), the last day of a month, or a specific weekday in a month (such as the third Thursday in February).
- Select **Minutes** in the **Frequency** field. You can specify that the job runs every 5, 10, 15, 20, or 30 minutes.

Note: If you select **Date List** in the **Frequency** field, you cannot select a date more than once.

- In the **Start time** field, specify when to start the event schedule. Click the field to select a time. Times are specified in 24-hour format.

TIP For examples, see [“Example: How to Specify the Start Time or Offset”](#) on page 20.

- In the **Time zone** field, specify the time zone to use when evaluating the time for the trigger and the date on which the trigger starts.

Note: If you choose **Date List** in the **Frequency** field, you must select the same value in the **Time zone** field for every scheduled date.

- In the **Start date** field, select the date on which the trigger starts.

Note: When you select the **Date List** frequency, the **End** field is not displayed.

- In the **End** field, specify when the trigger ends. You can specify that the trigger never ends, that it ends after a certain number of times, or that it ends on a specific date.



Note: When you select the **Date List** frequency, the **End** field is not displayed.

- Select **>>** to close the **Time Event Properties** pane.

Create a Command-Line Action

Note: You can save command-line action objects to use with other job flows. See [“Saving and Reusing Command-Line Actions”](#) on page 30.

Add a Command-Line Action

- 1 In the New Job Flow window, select , and then select **Add** ⇒ **Command Line Action** from the pop-up menu.
A **Command Line *n*** object is placed in the job-flow editor.
- 2 Select the object. If the right pane is not open, click  from the toolbar.
- 3 In the **Command Line Action** pane, specify the name and description for the action.
- 4 In the **Command line** field, specify the command that runs when the object is activated. See [“Examples of Commands” on page 29](#).

Note:

- The command syntax must match the operating system of the server on which the command runs.
- Command-line actions are not monitored or logged. Add the appropriate logging options to the command if you want to produce a log.
- Because commands run using the credentials of the user who created the job flow, make sure that you have permissions for any folders that the command uses.
- Commands run using the credentials of the user account that scheduled the job flow.

The group-managed service account must be assigned all of the permissions and authorizations required to use any of the resources that the scheduled jobs and job flows use. Resources can include folders, files, databases, and so on.

Individual accounts must also have the applicable permissions and authorizations.

- 5 In the **Priority** field, specify the priority that the action takes in the operating system when it runs.

Note: The action completes successfully if the return code from the command is zero (0). If the return code is any value other than zero, the action fails.

- 6 Select **>>** to close the **Command Line Action** pane.

Examples of Commands

Here are examples of commands that you can use.

Specify a full path to a batch SAS program and any job options:

```
/opt/sas/spre/home/SASFoundation/bin/sas_u8 -sysin /tmp/jobtest.sas -log /tmp/jobtest.log
```

Specify a full path to a Linux shell script:

```
/opt/sas/viya/tmp/job.sh
```

Note: The job.sh file can contain commands such as the following:

```
#!/bin/bash
```

```
echo "hello world" >> /opt/sas/viya/tmp/job.log
```

Specify shell script commands:

```
/bin/bash -c 'ls -la /opt/sas/viya/home/bin/ >> /opt/sas/viya/command/test4'
```

```
/bin/bash -c 'rm /opt/sas/viya/command/test4'
```

TIP The `/bin/bash -c` at the beginning of the command line enables you to issue multiple commands on a single line.

Saving and Reusing Command-Line Actions

Limitations of Reusable Command-Line Actions

You can save command-line action objects to the **Saved Actions** area in the left pane, and then use those objects in other job flows.

Remember the following limitations when working with command-line action objects:

- You can save only command-line actions.
- In SAS Environment Manager you can use a saved action only in the job flow for which you created the action.
However, in the jobs plug-in to the sas-admin CLI, you can create job actions and share them among multiple job flows.
- Only the account that created the saved action can see it. These actions cannot be shared with other users.
- The group-managed service account must be assigned all of the permissions and authorizations required to use any of the resources that the scheduled jobs and job flows use. Resources can include folders, files, databases, and so on.

Save and Reuse Command-Line Actions

- To save a command-line action to the **Saved Actions** area, right-click on the object in the job-flow editor and select **Add to saved actions**.
- To use a saved command-line action, expand the **Saved Actions** node in the left pane and drag a saved action to the job-flow editor.
- To delete a saved command-line action, right-click on the saved action and select **Delete**.
- To view the properties of a saved command-line action, right-click on the saved action and select **View properties**.

Modifying a Job Dependency

About the Job Dependency



By default, a connection from a job to another object (a job dependency) specifies that the connected object runs after the job completes successfully.

Here are the possible conditions for the job dependency:

- completes successfully
- ends with any exit code
- starts
- ends with an exit code (you specify the exit code criteria)
- maximum run time (you specify the run time)
- minimum run time (you specify the run time)

Modify a Job Dependency



To change the job-dependency criteria, complete these steps:

- 1 From the **Scheduling** tab, select a job flow in the **Jobs and Flows** table.
- 2 Click  on the toolbar and select **Edit flow** from the menu. You can also select **Edit Flow** from the pop-up menu.
- 3 In the Edit Job Flow window, select the connection between a job and another object.
- 4 Click  from the vertical toolbar on the right side of the job-flow editor.
- 5 In the **Job Dependency Properties** pane, modify the event type or any associated properties.
- 6 Click **Save**.

Modify Job Flow Properties and Attributes

The properties and attributes for a job flow are available from the vertical toolbar on the right side of the job-flow editor.

TIP Make sure none of the objects in the job-flow editor are selected.


- Click  to open the **Job Flow Properties** pane, where you can specify the name and description.
- Click  to open the **Attributes** pane. In this pane you can specify the following items:
 - the job-flow completion criteria
 - the source for the job-flow exit code

- actions to use after the job flow has completed




Note: By default, the job flow is defined as completed when one of these conditions is true:

- All the objects and actions complete successfully.
 - Any object in the job flow fails.
-
- Click » to close the properties or attributes pane.

Modify Object Properties

- 1 In the job-flow editor, select an object in the job flow.
- 2 Click  from the vertical toolbar on the right side of the job-flow editor to view the object properties.

When you select a job, additional types of information are available:



- **Arguments:** Click  to view name-value pairs for values such as contentType and backupType.
- **Job definition:** Click  to view details about the job such as ID, type, and parameters.
- **Actions:** Click  to view the name, description, job request URI, and priority of the job request.

Schedule Job Flows

Note: When the applicable hot fixes are applied, the group-managed service account feature is available. The Jobs and Flows page enables members associated with a group-managed service account to perform tasks by using the service account. For more information, see [“Using a Group-Managed Service Account” on page 117](#).

Schedule a Job Flow

You can create a schedule for a job flow, including the execution time and the time interval.

- 1 From the **Scheduling** tab, select a job flow in the **Jobs and Flows** table.
- 2 Click  in the toolbar or select **Schedule** from the pop-up menu.
- 3 (Optional) By default, the schedule runs using your account. In the **Run as** field, you can choose to run it using another account. Click  to select from available accounts.

(Default) Current user

Runs using your own user account.

A group-managed service account

Runs using a group-managed service account. You must be a member of a group that has been granted access to the service account credentials.

Note: The group-managed service account option is available after the applicable hot fixes have been applied. When the SAS Viya environment is updated, the group-managed service account function becomes available and the prior ability to use an individual's account other than your own is no longer available. For more information, see [“Group-Managed Service Account: About” on page 117](#).

- 4 Select or create a trigger to schedule the job flow. A trigger controls when the job flow runs. In the **Available triggers** table, for each of the triggers that you want to use, select the **Enabled** slider control. To create a new trigger, see [“Create a Time-Based Trigger for a Job Flow” on page 33](#).

Note:

- You can use a trigger with only the job flow for which it was created.
- **Time Event** is the only trigger type that is supported.

- 5 Click **Save** to save the schedule for the job flow.

Note: If you edit the job flow later, you must unschedule the job flow and then schedule it again.

Create a Time-Based Trigger for a Job Flow

- 1 In the Schedule Flow window, select **+** above the **Available triggers** table.
- 2 In the **Frequency** field, specify how often to repeat the trigger (such as a specified number of minutes, hours, or days).
- 3 Depending on your selection of the frequency interval, additional fields appear in the window to enable you to completely specify a frequency for the trigger. Use these fields to specify the criteria for the trigger interval.

Here are examples of frequency-interval criteria:

- Select **Yearly** in the **Frequency** field. You can specify a day of a month (such as the first of January), the last day of a month, or a specific weekday in a month (such as the third Thursday in February).
- Select **Minutes** in the **Frequency** field. You can specify that the job runs every 5, 10, 15, 20, or 30 minutes.

Note: If you select **Date List** in the **Frequency** field, you cannot select a date more than once.

- 4 In the **Start time** or **Offset** field, specify when to start the schedule. Click the field to select a time. Times are specified in 24-hour format.

Note: When you select **Minutes** or **Hours** in the **Frequency** field, the **Offset** field is displayed. The field specifies an offset from midnight.

When you select any other frequency, the **Start time** field is displayed.

Note:

If a hot fix is installed, when you select **Minutes** or **Hourly** in the **Frequency** field, the **Start time** field is now called **Offset**.

TIP For examples, see [“Example: How to Specify the Start Time or Offset”](#) on page 20.

Note: The following restriction is for job-flow triggers only. It does not affect job triggers.

If you specify **Hourly** in the **Frequency** field, the value of the **Offset** field (the offset from midnight) must be less than the value for the **Every** field (the skip count of occurrences). If you specify an invalid value, a warning message is displayed and the **Every** field is set to the nearest valid value.

- 5 In the **Time zone** field, specify the time zone to use when evaluating the time for the trigger and the date on which the trigger starts.
-

Note: If you choose **Date List** in the **Frequency** field, you must select the same value in the **Time zone** field for every scheduled date.

- 6 In the **Start date** field, select the date on which the trigger starts.
-

Note: When you select the **Date List** frequency, the **End** field is not displayed.

- 7 In the **End** field, specify when the trigger ends. You can specify that the trigger never ends, that it ends after a certain number of times, or that it ends on a specific date.
-

Note: When you select the **Date List** frequency, the **End** field is not displayed.

- 8 Click **Save**.
-

Manage Job Flows

Note: When the applicable hot fixes are applied, the group-managed service account feature is available. The Jobs and Flows page enables members associated with a group-managed service account to perform tasks by using the service account. For more information, see [“Using a Group-Managed Service Account”](#) on page 117.

Considerations for Working with Job Flows

After a job flow is created, the following considerations apply when working with job flows in SAS Environment Manager:

- Only the user who created the job flow or an administrator can edit or delete the job flow.
- Any authenticated user can schedule the job flow.
- Whoever schedules the job flow (or an administrator) can reschedule or unschedule the job flow.
- A group of users can share the ability to schedule, reschedule, and unschedule a job flow by being members of a group with access to a group-managed service account.

Execute a Job Flow Immediately

- 1 From the **Scheduling** tab, select a job flow in the **Jobs and Flows** table.
- 2 Choose the applicable option from the pop-up menu:
 - To use your own ID, select **Execute**.
 - To use a group-managed service account, select **Run as**.

TIP

- The **Flow_name Execution History** window displays only the most recent instance of a job flow that is executed using this command.
- If you use the **Execute** command on a job flow that has also executed from a schedule, the history for only the scheduled executions is displayed.

Edit a Job Flow

- 1 From the **Scheduling** tab, select a job flow in the **Jobs and Flows** table.
- 2 Click **:** on the toolbar and select **Edit flow** from the menu. You can also select **Edit Flow** from the pop-up menu.
- 3 In the Edit Job Flow window, modify the job flow.
- 4 Click **Save**.

Note: If you previously scheduled the job flow, you must unschedule it and then schedule the job flow again.

Edit a Job-Flow Schedule

After a job flow is scheduled, you can edit the schedule for the job flow.

- 1 From the **Scheduling** tab, select a scheduled job flow (🕒) in the **Jobs and Flows** table.

TIP The following indicators appear in the **Scheduled** column:

- 🕒 indicates a scheduled job flow that contains at least one enabled trigger.
- 🚫 indicates a scheduled job flow that contains disabled triggers.

- 2 Click 🗑️ or select **Edit Schedule** from the pop-up menu.
- 3 In the Edit Schedule window, you can add, edit, and remove triggers for the job-flow schedule.
- 4 Click **Save** when you have finished modifying the schedule.

Disabling or Uncheduling the Schedule for a Job Flow

Choosing between Disable and Unschedule

To prevent a job flow from running its specified schedule, you can either disable the triggers or unschedule the job flow.

- Disabling the triggers prevents the job flow from running the schedule and preserves the defined triggers.
- Uncheduling the job flow prevents the job flow from running on the defined schedule.



CAUTION

This option deletes the triggers that are specified for the job flow. It also deletes the job-flow history.

Disable Triggers

- 1 From the **Scheduling** tab, select a scheduled job flow (🕒) in the **Jobs and Flows** table.
- 2 Click 🗑️ from the toolbar or select **Edit Schedule** from the pop-up menu.
- 3 In the Edit Schedule window, disable all slider controls in the **Enabled** column of the **Available triggers** table.

TIP



- If you disable all triggers for a job flow,  appears in the **Scheduled** column.
- If any of the triggers for the job flow are enabled  appears in the column.

Unschedule a Job Flow

CAUTION


When you unschedule a job flow, any enabled triggers that are associated with the job flow are deleted. The job-flow history is deleted also. To unschedule a job flow and retain the triggers, edit the schedule and manually disable the triggers. See [“Disable Triggers” on page 36](#). Alternatively, you can use the `job flows unschedule` CLI command. It does not delete the triggers associated with the job flow. The command does delete the job-flow history.

Note: If you are using a group-managed service account, you can unschedule a job flow if it was scheduled via the same group-managed service account.

- 1 From the **Scheduling** tab, select a scheduled job flow () in the **Jobs and Flows** table.
- 2 Click  from the toolbar or select **Unschedule** from the pop-up menu.

Delete a Job Flow from the Scheduling Tab

Job flows remain in the list on the **Scheduling** tab unless you delete them.

- 1 From the **Scheduling** tab, select a job flow in the **Jobs and Flows** table.
- 2 Click  on the toolbar and select **Delete** from the menu. You can also select **Delete** from the pop-up menu.

Jobs and Flows: How To (CLI)

You can perform many of the tasks of managing SAS Viya platform jobs and flows with the job plugin to the `sas-admin` CLI. Before using the command-line interface, be sure to review [“Object Definitions” on page 10](#).

Create a Job Request

TIP To use the sas-admin CLI, you must sign in to the SAS Viya platform at the command line. See “[Command-Line Interface: Preliminary Instructions](#)” in *SAS Viya Administration: Using the Command-Line Interfaces* for information.

Step 1: Obtain the Job-Definition Definition Template

Obtain a template for the job definition. Enter this command:

```
sas-admin job definitions generate-template --file-out definition-filename
```

For details, see “[job definitions generate-template](#)” on page 65.

Step 2: Modify the Template and Create the Job Definition

- 1 Modify the job-definitions template to supply information about the job that you want to run.

For an annotated example of the job-definitions template, see “[Job-Definition Definition](#)” on page 95.

- 2 Use the job-definitions template to create the job definition.

```
sas-admin job definitions create --file-in definition-filename
```

For details, see “[job definitions create](#)” on page 65.

This command returns the ID for the job definition.

Step 3: Create the Job-Requests Definition Template

Obtain a template for the job request. Enter this command:

```
sas-admin job requests generate-template --file-out request-filename
```

For details, see “[job requests generate-template](#)” on page 68.

Step 4: Modify the Template and Create the Job Request

- 1 Modify the generated job-request template file to supply information about the job that you want to run.

Note: You can specify the job definition in the job-request template in either of the following ways:

- Use the job definition that you created in [“Step 2: Modify the Template and Create the Job Definition” on page 38](#).
- Specify the job-definition URI constructed with the ID that you received in [“Step 2: Modify the Template and Create the Job Definition” on page 38](#). See [“Construct a URI for Use in a Definition” on page 60](#).

For an annotated example of the job-requests definition template, see [“Job-Request Definition” on page 100](#).

- 2 Use the job-request definition template to create the job request.

```
sas-admin job requests create --file-in request-filename
```

For details, see [“job requests create” on page 68](#).

The command returns an ID for the job request.

TIP After you create a job request, you can reuse its output for other commands. For examples, see [“Run a Job Request” on page 39](#) and [“Schedule a Job Request” on page 39](#).

Run a Job Request

TIP After you create a job request, you can run it multiple times or schedule it.

- 1 Create a job request or use an existing job request. See [“Create a Job Request” on page 38](#).
- 2 Run the request. Use the job request ID that you created in the prior step. The job runs immediately.

```
sas-admin job requests execute --id request-ID
```

For details, see [“job requests execute” on page 71](#).

Schedule a Job Request

- 1 Create a job request or use an existing job request. See [“Create a Job Request” on page 38](#).
- 2 Create a JSON file to specify the time triggers for the job. See [“Define a Time-Based Trigger” on page 46](#) for information about specifying the triggers and creating a trigger file.
- 3 To schedule the job request, specify the ID of the job request and the JSON file that contains the time triggers.

```
sas-admin job requests schedule --id job-request-id --file-in triggers-file
```

For details, see [“job requests schedule” on page 72](#).

Create a Job Action for Use in a Job Flow

You can use either a job request or a command-line command in a job flow. To do so, they must be specified in a job-action definition. Each job-action definition can contain only one job request or one command-line command. A job flow can contain multiple job actions.

Use a Job Request to Create the Job Action

Complete these steps:

- 1 Obtain a template of the job-action definition. Enter this command:

```
sas-admin job actions generate-template [--file-out filename.json]
```

where *filename.json* specifies the name of an optional JSON file to contain the generated template. For details about this command, see [“job actions generate-template” on page 77](#).

- 2 Construct the job-request URI to use in the template. See [“Construct a URI for Use in a Definition” on page 60](#).

- 3 Add the job-request URI to the job-action template:

```
{
  "name": "Job-action example with a job request",
  "description": "Create a job action by specifying a job request.",
  "jobRequestUri": "/jobExecution/jobRequests/job-request_URI"
}
```

Where *job-request_URI* is the job-request URI that you constructed in the previous step. For details about the template, see [“Job-Action Definition” on page 99](#).

- 4 Save the definition in its JSON file.

TIP Remember to validate the JSON.

- 5 Create the job action and its ID. Use this command:

```
sas-admin job actions create --file-in filename.json
```

where *filename.json* specifies an optional JSON file that contains the definition. For details about this command, see [“job actions create” on page 78](#).

Use a Command-Line Command to Create the Job Action

Complete these steps:

- 1 Obtain a template of the job-action definition. Enter this command:

```
sas-admin job actions generate-template [--file-out filename.json]
```


where *filename.json* specifies the name of an optional JSON file to contain the generated template. For details about this command, see [“job actions generate-template” on page 77](#).

- 2 Add the fully qualified command-line command to the job-action template:

```
{
  "name": "Job-action example with a command-line command",
  "description": "Create a job action by specifying a command-line command.",
  "commandLine": "fully_qualified_command_line_command"
}
```

Where *fully_qualified_command_line_command* is the location and file name of the command. For details about the template, see [“Job-Action Definition” on page 99](#).

Note:

- A command-line command can fail if the command is not permitted by the SAS Launcher Service allow list. For more information about the allow list feature, see [“Security Considerations” in SAS Viya Administration: Programming Run-Time Servers](#).
 - SAS Launcher Service does not validate whether the command is a valid command to run in the pod. It is the responsibility of the creator of the job action that contains the command to ensure that the command is valid to run.
-

- 3 Save the definition in its JSON file.

TIP Remember to validate the JSON.

- 4 Create the job action and its ID. Use this command:

```
sas-admin job actions create --file-in filename.json
```

where *filename.json* specifies an optional JSON file that contains the definition. For details about this command, see [“job actions create” on page 78](#).

Create a Job Flow and Its Definition

Step 1: Obtain the Job-Flow Definition Template

Obtain a template of the job-flow definition. Enter this command:

```
sas-admin job flows generate-template --file-out flow_template.json
```

where *flow_template.json* is the name specified for the output JSON file.

For an annotated example of the job-flow definition template, see [“Job-Flow Definition Template” on page 103](#).

Step 2: Construct the URI Values for the Jobs Array

The jobs array is an array of URI values for job actions, subflows, or both. You must construct each URI that you want to add to the jobs array. See [“Construct a URI for Use in a Definition” on page 60](#).

Step 3: Update the Jobs Array in the Definition

- 1 If the definition does not contain the jobs array, insert the jobs array into the definition template.

```
"description": "Example 1 - create a flow",
  "jobs": [
  ],
  "triggerType": "runNow",
```

- 2 If the definition does contain the jobs array, delete the following text inside the jobs array: "Replace this string with a comma-separated-values list of job action IDs, subflow IDs, or both."
- 3 Add the URIs that you created in [“Step 2: Construct the URI Values for the Jobs Array” on page 42](#) to the jobs array:

```
"jobs": [
  "/jobFlowScheduling/jobs/job_action_ID",
  "/jobFlowScheduling/flows/subflow_ID"
]
```

Note:

- Surround each URI with quotation marks (“”).
 - Separate each URI with a comma.
-

Step 4: Complete the New Definition and Create a Job Flow

- 1 Confirm that you have updated the required key-value pairs in the definition.
- 2 Confirm that you have set any optional key-value pairs for your job flow.
- 3 Save the definition in its JSON file.
- 4 Create the job flow. Use this command:

```
sas-admin job flows create --file-in flow_template.json
```

where *flow_template.json* is the name of the JSON file that contains the definition.

Update a Job Flow

You can update several key-value pairs in a job-flow definition as well as add a trigger.

- 1 List all existing job flows. Enter this command:

```
sas-admin job flows list
```

- 2 In the output, find the name of the job flow that you want to use. Its job-flow ID is listed with its name.

- 3 Use the job-flow ID to obtain the existing job-flow definition. Enter this command:

```
sas-admin job flows show --id job_flow_id
```

The existing job-flow definition is returned in JSON format via Stdout.

TIP Pipe the output into a file to save the definition JSON.

- 4 Update one or more values in the definition. You can choose to include only the key-value pairs that you are changing in the definition. For example, if you want to change the name of the job flow, you can submit a definition that contains only the name key-value pair:

```
{  
  "name": "Flow for starting backups"  
}
```

Note:

- For descriptions of the definition, its key-value pairs, and their settings, see [“An Existing Job-Flow Definition” on page 105](#).
- Updates can include a trigger. See [“Add a Trigger Definition to a Job Flow Definition” on page 46](#).

- 5 Update the job flow with the updated definition. Use the `--file-in` option to submit the updated definition in a JSON file. Enter this command:

```
sas-admin job flows update --id job_flow_id --file-in path_to_updated_flow_json
```

- 6 Schedule the updated job flow. Enter this command:

```
sas-admin job flows schedule --id job_flow_id
```

Note: A job flow is scheduled by the Job-Flow-Scheduling Service. For time-based triggers, the Job-Flow-Scheduling Service uses the SAS Scheduler Service.

Trigger a Job Flow with a Manual Trigger

Why Use a Manual Trigger?

A manual trigger is useful for debugging job flows. A manual trigger enables you to trigger a job flow immediately instead of waiting for the job-flow criteria to meet its conditions to trigger the job flow.

Note: The `job flows trigger` command enables you to cause a job flow to run manually. This is the equivalent of the **Run Now** option on the **Jobs and Flows** page in SAS Environment Manager.

Here are examples of debugging scenarios that benefit from the manual trigger:

- You need to test a complex job flow. For example, you might have a job flow with an AND gate that waits for a collection of job requests to complete before executing the next step.

By manually triggering the job flow, you can review the job-flow history to ensure that the job requests ran in the order that you expected.

- You receive an error after using the **Run Now** option on the **Jobs and Flows** page in SAS Environment Manager.

By manually triggering the job flow, you can determine whether the equivalent CLI commands generate the same errors that you received in SAS Environment Manager. This method enables you to identify whether the problem is in SAS Environment Manager or in SAS Job Flow Scheduler. See [“Execute a Job Flow Immediately” on page 8](#).

How the Manual Trigger Works

By setting the `triggerType` to `manual` in the job-flow definition, you specify that the job flow is scheduled but does not use a defined trigger such as a time-based event. Instead, the job flow expects a manual trigger. Typically, the manual trigger is delivered by using the `job flow trigger` command.

Note: Alternatively, you can have an interactive workflow that causes a REST API call to trigger the job flow after a step in the workflow completes.

Use the Manual Trigger

Note: You cannot schedule a job flow with a manual trigger via the **Jobs and Flows** page in SAS Environment Manager. You must schedule the job flow by using the job CLI.

- 1 Obtain the definition of the job flow on which you want to use the manual trigger. For a detailed procedure, see [“Update a Job Flow” on page 43](#).

- 2 Update the job-flow definition to use the manual trigger type:

```
"triggerType": "manual",
```

Note: For descriptions of the definition, its key-value pairs, and their settings, see [“An Existing Job-Flow Definition”](#) on page 105.

TIP Save the values from the following key-value pairs in the job-flow definition structure:

- definitionId
- schedulerId

These values are used in a later step.

- 3 Update the job flow with the updated definition. Use the `--file-in` option to submit the updated definition in a JSON file. Enter this command:

```
sas-admin job flows update --id job_flow_id --file-in path_to_updated_flow_json
```

- 4 Schedule or reschedule the updated job flow by using the job CLI. Enter this command:

```
sas-admin job flows schedule --id job_flow_id
```

- 5 Manually trigger the job flow. Enter this command:

```
sas-admin job flows trigger --id job_flow_definition_id --sch scheduler_id
```

where *job_flow_definition_id* and *scheduler_id* are the values that you saved in step 2.

For more information about the command, see [“job flows trigger”](#) on page 89.

- 6 The CLI triggers the specified job-flow instance. To determine whether the job-flow instance was triggered, check the job-flow instance status or the job-flow instance history for the execution status. See [“job flows show-status”](#) on page 88 and [“job flows show-history”](#) on page 87.

TIP You can also check the execution status on the **Jobs and Flows** page in SAS Environment Manager. See [“View the Execution History for a Job Flow and Status of Its Objects”](#).

Add a Trigger

Overview

TIP When creating or updating any JSON structure, use a JSON validation utility to ensure that the structure is valid.

For details about the trigger definition, see [“Define a Time-Based Trigger” on page 46](#).

Add a Trigger Definition to a Job Request

You can add a time-based trigger to a job request by using the `job requests schedule` command. To do so, save the trigger definition in a JSON file. Then, specify that file in the `filename` option. For details, see [“job requests schedule” on page 72](#).

Add a Trigger Definition to a Job Flow Definition

You must update the job-flow definition that is returned by using the `job flows show` command. To update, add the trigger definition structure between the `triggerType` and `triggerCondition` key-value pairs. When you have made all your changes, save the definition in its JSON file. Complete the process described in [“Update a Job Flow” on page 43](#).

Note: The following example is a fragment from a job-flow definition JSON structure. For an example of an entire job-flow definition structure, see [“An Existing Job-Flow Definition” on page 105](#).

This example shows the time-based trigger definition.

```
"triggerType": "event",
"triggers": [
  {
    "type": "timeevent",
    "name": "hourly trigger",
    "active": true,
    "event": {
      "recurrence": {
        "type": "hourly",
        "startDate": "2022-09-13",
        "skipCount": 4
      },
      "minutes": "15",
      "hours": "12"
    }
  }
],
"triggerCondition": "any",
```

Define a Time-Based Trigger

Considerations

- For details about adding a trigger, see [“Add a Trigger” on page 45](#).
- For descriptions of the options for each recurrence, see [“Time-Based Trigger Definition” on page 111](#).

- The time-based trigger uses a 24-hour format. Time is specified using 0-23.
- When creating or updating any JSON structure, use a JSON validation utility to ensure that the structure is valid.

Minutely Recurrence Interval

Here is an example that specifies a trigger that starts on November 26, 2022:

```
[{
  "type": "timeevent",
  "name": "minutely trigger",
  "active": true,
  "event": {
    "recurrence": {
      "type": "minutely",
      "startDate": "2022-11-26",
      "skipCount": 15
    },

    "minutes": "6"
  }
}]
```

- The recurrence sets the job to run every 15 minutes.
- Because the trigger uses the `minutely` recurrence type, `minutes` indicates to start the job at 6 minutes past the hour. Therefore, jobs run at 06, 21, 36, and 51 minutes of each hour.

The job runs at 06, 21, 36, and 51 minutes of each hour starting on November 26, 2022.

Note: If the trigger is saved on November 26, 2022, then the first job runs at the next `minutes` interval after the save. For example, if the trigger is saved at 09:15, the first job runs at 09:21.

Hourly Recurrence Interval

Here is an example that specifies a trigger that starts on September 13, 2022:

```
[{
  "type": "timeevent",
  "name": "hourly trigger",
  "active": true,
  "event": {
    "recurrence": {
      "type": "hourly",
      "startDate": "2022-09-13",
      "skipCount": 4
    },

    "minutes": "15",
    "hours": "12"
  }
}]
```

- The `recurrence` sets the job to run every 4 hours.
- This setting is ignored. Because the trigger uses the `hourly` recurrence type, `hours` indicates to start the job at the 12th hour from the start of the day. However, the `hours` offset is not valid because it is not less than the `skipCount` value of 4.
- The `minutes` setting indicates to run the jobs at 15 minutes past the hour.

The job runs at 04:15, 08:15, 12:15, 16:15, 18:15, 22:15 hours of each day starting on September 13, 2022.

Note: If the trigger is saved on September 13, 2022, then the first job runs at the next `hours` interval after the save. For example, if the trigger is saved at 09:15, the first job runs at 12:15.

Daily Recurrence Interval

Here is an example that specifies a trigger that starts on July 4, 2022:

```
[{
  "type": "timeevent",
  "name": "daily trigger",
  "active": true,
  "event": {
    "recurrence": {
      "type": "daily",
      "startDate": "2022-07-04",
      "daysOfWeek":
["monday", "tuesday", "wednesday", "thursday", "friday"]
    },
    "hours": "13",
    "minutes": "0"
  }
}]
```

- The `recurrence` sets the job to run every weekday.
- The `hours` setting indicates to start the job at the 13th hour from the start of the day.
- The `minutes` setting indicates to run the jobs at 0 minutes past the hour.

The job runs every weekday at 13:00 hours starting on July 4, 2022.

Weekly Recurrence Interval

Here is an example that specifies a trigger that starts on November 3, 2021:

```
[{
  "type": "timeevent",
  "name": "weekly trigger",
  "active": true,
  "event": {
    "recurrence": {
      "type": "weekly",
      "startDate": "2021-11-03",
      "skipCount": 4,

```



```

        "daysOfWeek": ["tuesday"]
      },
      "hours": "6",
      "minutes": "15"
    }
  ]
}

```

- The `recurrence` sets the job to run every four weeks on Tuesday.
- The `hours` setting indicates to start the job at the 6th hour from the start of the day.
- The `minutes` setting indicates to run the job at 15 minutes past the hour.

After the trigger starts on Wednesday, November 3, 2021, the job runs every four weeks on Tuesday at 06:15 hours.

Monthly Recurrence Interval

Example 1 specifies a trigger that starts on February 1, 2022:

```

[ {
  "type": "timeevent",
  "name": "monthly trigger",
  "active": true,
  "event": {
    "recurrence": {
      "type": "monthly",
      "startDate": "2022-02-01",
      "skipCount": 1,
      "daysOfWeek": ["sunday"],
      "occurrence": "second"
    },
    "hours": "17",
    "minutes": "0"
  }
} ]

```

- The `recurrence` sets the job to run on the second Sunday of every month.
- The `hours` setting indicates to start the job at the 17th hour from the start of the day.
- The `minutes` setting indicates to run the job at 0 minutes past the hour.

After the trigger starts on Tuesday, February 1, 2022, the job runs on the second Sunday of every month at 17:00 hours.

Example 2 specifies a trigger that starts on May 1, 2022:

```

[ {
  "type": "timeevent",
  "name": "monthly trigger",
  "active": true,
  "event": {
    "recurrence": {
      "type": "monthly",
      "startDate": "2022-05-01",
      "dayOfMonth": 15
    },
    "hours": "8",

```

```

        "minutes": "15"
      }
    ]}

```

- The `recurrence` sets the job to run on the 15th day of every month.
- The `hours` setting indicates to start the job at the 8th hour from the start of the day.
- The `minutes` setting indicates to run the job at 15 minutes past the hour.

After the trigger starts on Monday, May 1, 2022, the job runs on the 15th day of every month at 08:15 hours.

Yearly Recurrence Interval

Example 1 specifies a trigger that starts on May 1, 2022:

```

[ {
  "type": "timeevent",
  "name": "yearly trigger",
  "active": true,
  "event": {
    "recurrence": {
      "type": "yearly",
      "startDate": "2022-05-01",
      "skipCount": 1,
      "daysOfWeek": ["friday"],
      "occurrence": "last",
      "monthOfYear": "june"
    },
    "hours": "15",
    "minutes": "0"
  }
} ]

```

- The `recurrence` sets the job to run on the last Friday of June in every year.
- The `hours` setting indicates to start the job at the 15th hour from the start of the day.
- The `minutes` setting indicates to run the job at 0 minutes past the hour.

After the trigger starts on Monday, May 1, 2022, the job runs on the last Friday of June in every year at 15:00 hours starting on May 1, 2022.

Example 2 specifies a trigger that starts on May 1, 2022:

```

[ {
  "type": "timeevent",
  "name": "yearly trigger",
  "active": true,
  "event": {
    "recurrence": {
      "type": "yearly",
      "startDate": "2022-05-01",
      "skipCount": 1,
      "dayOfMonth": 31,
      "monthOfYear": "may"
    },
    "hours": "14",

```

```

        "minutes": "0"
      }
    ]}

```

- The `recurrence` sets the job to run on the 31st day of May of every year.
- The `hours` setting indicates to start the job at the 14th hour from the start of the day.
- The `minutes` setting indicates to run the job at 0 minutes past the hour.

After the trigger starts on May 1, 2022, the job runs on the 31st day of May of every year at 14:00 hours starting on May 1, 2022.

Date List Recurrence Interval

This example specifies a trigger that starts on January 1, 2022:

```

[ {
  "type": "timeevent",
  "name": "datelist trigger",
  "active": true,
  "event": {
    "recurrence": {
      "type": "datelist",
      "startDate": "2022-01-01",
      "dates": [
        "2022-02-28",
        "2022-02-03",
        "2022-03-25"
      ]
    },
    "hours": "14",
    "minutes": "0"
  }
} ]

```

- The `recurrence` sets the job to run on February 3, 2022, February 28, 2022, and March 25, 2022.
- The `hours` setting indicates to start the job at the 14th hour from the start of the day.
- The `minutes` setting indicates to run the job at 0 minutes past the hour.

After the trigger starts on January 1, 2022, the job runs on February 3, 2022, February 28, 2022, and March 25, 2022 at 14:00 hours.

Determine If a Job or Job Flow Completed Successfully

Determine If a Job Request Completed Successfully

Note: To use the following procedure, one of the following statements must be true:

- You are the creator of the job request.
 - The creator of the job request or an administrator has shared the job request ID with you.
-

1 You must know the job request ID.

If you do not know the ID and you created the job request, you can obtain the ID by using the `job requests list` command. For details, see [“job requests list” on page 69](#).

Otherwise, you must obtain the ID from the creator of the job request or an administrator.

2 Next, you must know the ID of the instance of the job request. If you do not, you can obtain the ID by using the `job requests list-history` command. For details, see [“job requests list-history” on page 74](#).

3 Use the `job requests show-history` command to determine the completion state of your specified job request instance. For details, see [“job requests show-history” on page 75](#).

Determine If a Job Flow Completed Successfully

1 You must know the job-flow ID. If you do not, you can obtain the ID by using the `job flows list` command. For details, see [“job flows list” on page 82](#).

2 Next, you must know the ID of the instance of the job flow. If you do not, you can obtain the ID by using the `job flows list-history` command. For details, see [“job flows list-history” on page 85](#).

3 Use the `job flows show-history` command to determine the completion state of your specified job-flow instance. For details, see [“job flows show-history” on page 87](#).

Obtain the History of an Executed Job Request or Job Flow

Obtain the History of an Executed Job Request

1 Obtain a list of job requests. Enter this command:

```
sas-admin job requests list
```

The output includes the name and job-request ID for each job request.

TIP You can reduce the number of job requests returned by using the `--filter` option. Be sure to review [“Specify a Filter for Command Output” on page 60](#).

Note: For details about this command and its options, see [“job requests list” on page 69](#).

- 2 In the output, find the name of the job request whose history you want. Its job-request ID is listed with its name.
- 3 Use the job-request ID to obtain the history of the job request. Enter this command:

```
sas-admin job requests list-history --id job_request_id
```

The output returns the history of the specified job request.

Note: For details about this command and its options, see [“job requests list-history” on page 74](#).

Obtain the History of an Executed Job Flow

- 1 List all existing job flows. Enter this command:

```
sas-admin job flows list
```

TIP You can reduce the number of job flows returned by using the `--filter` option. Be sure to review [“Specify a Filter for Command Output” on page 60](#).

Note: For details about this command and its options, see [“job flows list” on page 82](#).

- 2 In the output, find the name of the job flow whose history you want to obtain. Its job-flow ID is listed with its name.
- 3 Use the job-flow ID to obtain a list of the completed instances of the job flow. Enter this command:

```
sas-admin job flows list-history --id job_flow_id
```

Note: For details about this command and its options, see [“job flows list-history” on page 85](#).

- 4 In the output, find the completed job-flow instance whose history you want. Note its job-flow instance ID.
- 5 Use the job-flow ID and the job-flow instance ID to obtain the history of the job-flow instance. Enter this command:

```
sas-admin job flows show-history --id job_flow_id --instance-id job_flow_instance_id
```

The output includes the start and end time of the job-flow instance, its state, exit code, and more.

- 6 If you need more information, you can obtain the history of the job actions and subflows within the specified instance of the job flow. Using the job-flow ID and the job-flow-instance ID, enter this command:

```
sas-admin job flows show-history-tree --id job_flow_id --instance-id job_flow_instance_id
```

The output includes all of the history records for the job actions and subflows that are associated with the specified job-flow instance. The output is sorted by start time.

TIP Sorted output enables you to see whether dependencies have run in the correct order.

Capture Results from a Program or Command

Note: The `--wait-results` option is available as of release 2.6.44 of the job CLI plug-in.

Capture Results via a Job Request

This example captures the exit code from a simple SAS program.

- 1 Specify a SAS program in a job-definition definition:

```
{
  "name": "--Wait-results example for job-definition definition",
  "description": "Show the results of data one using PROC PRINT",
  "type": "Compute",
  "code": "data one; do i=1 to 10; output; end; run; proc print; run;"
}
```

- 2 Save the JSON structure to a file.
- 3 Create the job-definition definition:

```
sas-admin job definitions create --file-in job_defn_wait_results.json
```

The CLI returns the ID of the job-definition definition:

```
9876-zzz
```

- 4 Specify the job-definition definition in a job request:

```
{
  "name": "--Wait-results example for job-request definition",
  "jobDefinition": {
    "name": "--Wait-results example for job-definition definition",
    "description": "Show the results of data one using PROC PRINT",
    "type": "Compute",
    "code": "data one; do i=1 to 10; output; end; run; proc print; run;"
  },
  "arguments": {
    "_contextName": "SAS Job Execution compute context"
  }
}
```

TIP You can specify the job-definition definition by reference instead of inclusion. Replace the `jobDefinition` structure with the following line:

```
"jobDefinitionUri": "/jobDefinitions/definitions/9876-zzz",
```

5 Save the JSON structure to a file.

6 Create the job-request definition:

```
sas-admin job requests create --file-in job_req_wait_results.json
```

The CLI returns the ID of the job request:

```
1234-abc
```

7 Execute the job request and wait on the results from the SAS program:

```
sas-admin job requests execute --id 1234-abc --wait-results
```

The CLI waits for the SAS program to complete, and the program's results are sent to stdout:

```
"State: completed"
```

Capture Results via a Job Flow

You can capture results from a job request or a command-line command by using a job flow.

1 Use the job request-ID that was created in [“Capture Results via a Job Request” on page 54](#). Specify the job request in a job-action definition to use in a job flow:

```
{
  "name": "--Wait-results example for job-action definition",
  "description": "Prepare the job request for use in a job flow.",
  "jobRequestUri": "/jobExecution/jobRequests/1234-abc"
}
```

TIP A job-action can contain a command-line command instead of a job request.

2 Save the JSON structure to a file.

3 Create the job-action definition:

```
sas-admin job actions create --file-in job_actn_wait_results.json
```

The CLI returns the ID of the job-action definition:

```
7273-klm
```

4 Specify the job action in a job-flow definition:

```
{
  "name": "--Wait-results example for job-flow definition",
  "description": "Create a job that contains the job request.",
  "jobs": ["/jobFlowScheduling/jobs/7273-klm"],
  "schedulerId": "57534784-2ba",
  "triggerType": "runNow"
}
```

}

TIP Use the `job schedulers list` command to identify the value for `schedulerId`. For details, see [“job schedulers list” on page 91](#).

5 Save the JSON structure to a file.

6 Create the job-flow definition:

```
sas-admin job flows create --file-in job_flow_wait_results.json
```

The CLI returns the ID of the job flow:

```
8123-trmisu-6
```

7 Execute the job flow. It executes the job request that it contains and waits on the results from the SAS program:

```
sas-admin job flows execute --id 8123-trmisu-6 --wait-results
```

The CLI waits for the SAS program to complete, and the program's results are sent to stdout:

```
"Exit code: 0 - State: completed"
```

For more information about the `--wait-results` option, see these topics:

- [“job requests execute” on page 71](#)
- [“job flows execute” on page 83](#)

Pause the Scheduler Before Migration

Before you migrate from one tenant to another tenant, you must pause the scheduler.

.....

Note: You must be a SAS administrator to use these commands.

.....

Use the following process:

- 1 Pause the scheduler. For details, see [“job schedulers pause” on page 92](#).
- 2 Migrate from one tenant to another tenant.
- 3 Restore the scheduler. For details, see [“job schedulers restore” on page 94](#).
- 4 Resume the scheduler. For details, see [“job schedulers resume” on page 93](#).

Adjust for Daylight Saving Time

About This Procedure

IMPORTANT This procedure must be done only once. The workaround that this procedure applies does not require any further adjustment during the year.

The start or end of Daylight Saving Time (DST) affects start times for the following items:

- unexecuted instances of scheduled job requests
- unexecuted instances of scheduled job flows

These instances might start an hour earlier or later, depending on the season.

To perform this procedure you must understand the following configuration properties:

`dstEnabled`

When `dstEnabled` is set to `true`, Scheduler Service enables time-event triggers to honor Daylight Saving Time. Only triggers that use the `daily` recurrence type are affected. For more information about this recurrence type, see [“Daily Recurrence”](#).

`timeZone`

Systems that host the SAS Viya platform are set with a default time-zone value via an operating-system setting. However, some environments might have multiple host systems that are set with different values for the default time zone. By setting the `timeZone` configuration property, you can ensure that when the `dstEnabled` property is set to `true`, the Scheduler Service interprets the trigger start time consistently among the multiple host systems.

IMPORTANT

- Verify that the selected time zone has adopted Daylight Saving Time. Coordinated Universal Time (UTC) is not a valid selection.
- It is recommended that you specify the time zone by using the TZ identifier (for example, `America/New_York`) instead of the TZ abbreviation (EST/EDT). For more information, see [Time Zone Database](#).

This procedure makes the following adjustments to unexecuted instances of scheduled job requests and scheduled job flows:

New instances

By setting `dstEnabled` to `true`, the Scheduler Service uses the Daylight Saving Time value for the start time on new unexecuted instances.

Existing instances

The job `schedulers restore` command ensures that existing unexecuted instances use the Daylight Saving Time value for the start time. Only instances that use the `daily` recurrence type are affected.

Adjust for Daylight Saving Time

Complete these steps:

- 1 In SAS Environment Manager, edit the Scheduler Service configuration instance.

TIP

- The Scheduler Service might be displayed as `schedule service`.
- For detailed steps, see [“Edit Configuration Instances” in SAS Viya Administration: Configuration Properties](#).
- If the Scheduler Service configuration instance is missing, create an instance. For detailed steps, see [“Create Configuration Instances” in SAS Viya Administration: Configuration Properties](#).

- 2 Set the `dstEnabled` configuration property to `true`.
- 3 Set the `timeZone` configuration property to the preferred time zone for the affected host system to use.

IMPORTANT

- Verify that the selected time zone has adopted Daylight Saving Time. Coordinated Universal Time (UTC) is not a valid selection.
- It is recommended that you specify the time zone by using the TZ identifier (for example, `America/New_York`) instead of the TZ abbreviation (EST/EDT). For more information, see [Time Zone Database](#).

- 4 Restore the existing scheduled instances that the Scheduler Service has saved in the SAS Viya platform.

All existing scheduled instances are restored unchanged except for scheduled instances that include a time-event trigger that uses the `daily` recurrence type. For these scheduled instances only, the start time of the trigger is modified for Daylight Saving Time.

```
sas-admin job schedulers restore
```

See [“job schedulers restore”](#).

Obtain IDs That Are Used as Input Values

Most of the job CLI commands require IDs for input values. Use the following table to identify the command that returns each type of ID.

TIP You can specify an optional filter to narrow the generated output on most of these commands.

ID Type	Command that Obtains the ID
Action ID	“job actions list” on page 79
Definition ID	“job definitions list” on page 66
Domain ID	Use the Domains view on the Domains page in SAS Environment Manager. See “Navigation” in SAS Viya Administration: External Credentials .
Flow ID	“job flows list” on page 82
Flow-instance ID	“job flows list-history” on page 85
Job-request ID	“job requests list” on page 69
Job-request-instance ID	“job requests list-history” on page 74
Scheduled-job-request ID	job requests list --scheduled on page 69
Scheduled-job-request-instance ID	job requests list-history --scheduled on page 74
Scheduler ID	“job schedulers list” on page 91

Retrieve the Launcher Context Name

To view the SAS Launcher service allow list, you must know the name of the launcher context that is used by the scheduler.

- 1 List all existing schedulers. Enter this command:

```
sas-admin job schedulers list
```

TIP A typical SAS Viya environment has only one scheduler.

- 2 In the output, find the name of the scheduler. Its scheduler ID is listed with its name.
- 3 Use the scheduler ID to obtain the scheduler definition. Enter this command:

```
sas-admin job schedulers show --id scheduler_id
```

The scheduler definition is returned in JSON format via Stdout.

TIP Pipe the output into a file to save the definition JSON.

- 4 In the JSON, locate the value in the `context` key-value pair. This is the launcher context name.

Construct a URI for Use in a Definition

- 1 Obtain the ID for the object whose URI you require.
See “Obtain IDs That Are Used as Input Values” on page 58.
- 2 Add the ID to the applicable URI path.

Object	URI Path
Job Definition	<code>/jobDefinitions/definitions/job-definition_ID</code> where <i>job-definition_ID</i> is the ID for the job definition.
Job Request	<code>/jobExecution/jobRequests/job-request_ID</code> where <i>job-request_ID</i> is the ID of the job request.
Job Action	<code>/jobFlowScheduling/jobs/job_action_ID</code> where <i>job-action_ID</i> is the ID of the job action.
Job Flow and Subflow	<code>/jobFlowScheduling/flows/flow_ID</code> where <i>flow_ID</i> is the ID of the job flow or subflow.
Scheduler	<code>/jobFlowScheduling/schedulers/scheduler_ID</code> where <i>scheduler_ID</i> is the ID of the scheduler in use.

Specify a Filter for Command Output

Example

Some commands in the job CLI include an option to filter the command output. The `--filter` option accepts an expression that you can specify to filter the output.

For example, the `job flows list` command returns a list of job flows. If you want to locate a job flow with a name that starts with “orion”, you can specify a filter expression in this way:

```
sas-admin job flows list --filter "startsWith(name,\"orion\")"
```

Where:

- `startsWith` is a filter function. For more filter functions, see [Filtering at developer.sas.com](#).
- `name` is the field name to evaluate in the expression.

TIP Field names are case sensitive. See “Known Limitations” on page 61.

- `orion` is the string that this example uses to evaluate the field name.

TIP Strings must be surrounded with either single-quotation marks (') or double-quotation marks ("). Use whichever quotation mark that is not surrounding the expression itself.

Note:

- You must surround the filter expression with either single-quotation marks (') or double-quotation marks (").
 - You must surround the string being evaluated with either single-quotation marks (') or double-quotation marks (").
 - If you use single-quotation marks (') to surround the filter expression, you must use double-quotation marks (") to surround the evaluation string, and vice versa.
-

The output returned by the command is limited to job flows with names that start with "orion".

Known Limitations

The CLI has the following known limitations when evaluating a filter expression:

- Strings that include one or more spaces are not evaluated correctly and produce a syntax error (for example, "new flow" or "new%20flow"). For a successful search, use strings that do not include spaces.
-

Note: This limitation is fixed by updating the job CLI plug-in to release 2.6.38 or later.

- Incorrect placement of single and double quotation marks cause syntax errors. Depending on the location of the incorrect quotation mark, the error can be generated by one of these entities:
 - the local operating system on which the CLI is running
 - the SAS Viya platform server-side evaluation

Make sure that you balance the single- and double-quotation marks correctly.

- Field names are case sensitive. The filter fails when the field name specified in the expression is not represented in the case that the CLI recognizes.

Incorrect	Correct
Any field name that contains "Id" or "id" (for example: schedulerId or schedulerid)	The field name must use "ID" (for example: schedulerID)
Any field name that contains "timestamp" or "Timestamp" (for example: modifiedtimestamp or modifiedTimestamp)	The field name must use "TimeStamp" (for example: modifiedTimeStamp)

Specify Standard Input

Some commands give the alternative of using standard input (Stdin) to provide input values instead of a JSON file.

Method 1

You can use a hyphen ("-") followed by a "here document" to input your values via Stdin.

Here is an example that creates a job definition:

```
sas-admin job definitions create --file-in -<<EOF
{"name": "my_definition", "type": "rest", "code": "/" }
EOF
```

Method 2

You can close a Stdin and send the command by using a keyboard combination.

The following example creates the job definition:

```
sas-admin job definitions create --file-in
{"name": "my_definition", "type": "rest", "code": "/" }
```

After the JSON structure, press the following three keys at the same time on your keyboard: Ctrl+Z +Enter.

Jobs and Flows: Reference (CLI)

Considerations

- Be sure to review [“Deprecations to the Job CLI Plug-In” on page 63](#).
- A job request is visible to its creator or a SAS administrator only.
- When you issue a command such as `job requests list`, the output is limited to the items that are visible to you.
- Job flows are visible to all users.
- The output generated by a command can vary. It depends on the output format (text, JSON, or fullJSON) that you selected when you ran the command `sas-admin profile init`. See [“Output Type” in SAS Viya Administration: Using the Command-Line Interfaces](#).
- A group-managed service account can use commands that have the `--domain-id` input option. See [“Using a Group-Managed Service Account” on page 117](#).

Note: The job CLI plug-in must be at release 2.6.29 or later. Also, the applicable hot fixes must be applied.

Deprecations to the Job CLI Plug-In

The job CLI plug-in has the following deprecations. The deprecations are still supported, but the support will be removed in the future.

Release 2.6.44

Some key-value pairs have been removed from the job-action definition template. These key-value pairs are removed for one or more of these reasons:

- They are deprecated.
- They are reserved for future use.
- They are assigned a permanent value.

The change does not affect existing job-action definitions. For details, see [“Job-Action Definition” on page 99](#).

Release 2.6.42

The Scheduled Subcommand

The `scheduled` subcommand is deprecated. Instead, use the `--scheduled` input option.

This change affects the following commands:

- [“job requests list” on page 69](#)
- [“job requests list-history” on page 74](#)
- [“job requests show” on page 70](#)

Note: The `scheduled` subcommand was introduced in release 2.6.5.

Definition Templates

Some key-value pairs have been removed from the following definition templates:

- [job definition on page 95](#)
- [job scheduler on page 110](#)
- [job request on page 100](#)

These key-value pairs are removed for one or more of these reasons:

- They are deprecated.
- They are reserved for future use.
- They are assigned a permanent value.

The change does not affect existing definitions.

Release 2.6.35

Some key-value pairs have been removed from the job-flow definition template. These key-value pairs are removed for one or more of these reasons:

- They are deprecated.
- They are reserved for future use.
- They are assigned a permanent value.

The change does not affect existing job-flow definitions. For details, see [“Job-Flow Definition Template” on page 103](#).

Release 2.6.5

The input option `--sch-id` is renamed to `--instance-id`. This change affects the “[job flows show-history](#)” command.

Definitions Commands

A job definition enables a job to run in batch. Use these commands to create and manage job definitions.

job definitions generate-template

Generates a job-definition template. Use the template to specify the job definition that you want to create with the command `job definitions create`. For details, see “[Job-Definition Definition](#)” on [page 95](#).

```
sas-admin job definitions generate-template [--file-out filename.json]
```

Input options

filename.json specifies the name of an optional JSON file to contain the generated template. For details, see “[Create a Job Request](#)” on [page 38](#).

TIP The command saves the file on the system that is running the CLI. When you specify the file name, include a path for its location on that system. If you do not specify a path, the file is written to the system’s working directory. You can identify the working directory by issuing the `pwd` command.

Output

- by default, writes the generated template to Stdout
- if `--file-out` is specified, returns the generated template in the specified JSON file

TIP The file is saved to the system that is running the CLI, not the server to which the CLI commands are issued.

job definitions create

Creates a job definition.

```
sas-admin job definitions create [--file-in filename.json] [--folder-id parent_folder_id]
```

Input options

- *filename.json* specifies an optional JSON file that contains the definition. Alternatively, you can provide the details through Stdin. See “[Specify Standard Input](#)” on [page 62](#).

- *parent_folder_id* specifies an ID to an optional parent folder on the SAS Viya platform in which to save the job-definition definition.

Note: The `--folder-id` option is available for this command as of release 2.7.1 of the job CLI plug-in. To acquire the folder ID to use for *parent_folder_id*, use the **Content** page in SAS Environment Manager or the folders CLI plug-in. See [“Folders: How To” in SAS Viya Administration: Folders](#).

Output

- **Success:** Returns the ID for the job definition.
- **Failure:** Returns a message indicating incorrect or missing input values.

job definitions update

Updates a job definition.

```
sas-admin job definitions update --id definition_id [--file-in filename.json]
```

Input options

- *definition_id* specifies the ID of the definition.
- *filename.json* specifies an optional JSON file that contains the updated definition. Alternatively, you can provide the details through Stdin. See [“Specify Standard Input” on page 62](#).

Output

- **Success:** Updates the specified definition.
- **Failure:** Returns a message indicating incorrect or missing input values.

job definitions delete

Deletes a job definition.

```
sas-admin job definitions delete --id definition_id
```

Input options

- *definition_id* specifies the ID of the definition.

Output

- **Success:** Deletes the specified definition.
- **Failure:** Returns a message indicating incorrect or missing input values.

job definitions list

Lists job definitions.

```
sas-admin job definitions list [--start record] [--limit limit_number] [--sort-by resource_name [,resource_name]] [--filter expression]
```

Input options

- *record* specifies the record from which to start displaying items. The default is to start from the first record.
- *limit_number* specifies the maximum number of items to display. The default value is 10.
- *resource_name* [, *resource_name*] specifies to sort the displayed items by one or more resource names.
 - Separate multiple values with commas.
 - The default sort order is ascending.
 - To sort in descending order, put the ~ (tilde) character before the resource name.
- *expression* specifies to filter the output using the specified regular expression. For more information, see [“Specify a Filter for Command Output” on page 60](#).

Output

Returns a list of job definitions including each associated definition ID.

job definitions show

Displays details about a job definition.

```
sas-admin job definitions show --id definition_id
```

Input options

definition_id specifies the ID of the definition.

Output

- **Success:** Returns details about the specified job definition including its ID, name, type, description, and more.
- **Failure:** Returns a message indicating incorrect or missing input values.

Requests Commands

Use these commands to manage job requests, which make job actions available for either immediate execution or via a schedule.

Considerations

- A job request is visible to its creator or a SAS administrator only.
- When you issue a command such as `job requests list`, the output is limited to the items that are visible to you.
- You must define a job request by using `job requests create`. After the job request is defined, you can do either of these tasks:
 - Execute the job request immediately.
 - Schedule the job request to be executed.

The scheduled-job-request ID is another ID that is separate and distinct from the job-request ID.

job requests generate-template

Generates a template for a job-request definition. Use the generated template to specify the job request that you want to create with the command `job requests create`. For details, see [“Job-Request Definition” on page 100](#).

```
sas-admin job requests generate-template [--file-out filename.json]
```

Input options

filename.json specifies the name of an optional JSON file to contain the generated template. For details, see [“Create a Job Request” on page 38](#).

TIP The command saves the file on the system that is running the CLI. When you specify the file name, include a path for its location on that system. If you do not specify a path, the file is written to the system's working directory. You can identify the working directory by issuing the `pwd` command.

Output

- by default, writes the generated template to Stdout
- if `--file-out` is specified, returns the generated template in the specified JSON file

TIP The file is saved to the system that is running the CLI, not the server to which the CLI commands are issued.

job requests create

Creates a job request.

```
sas-admin job requests create [--file-in filename.json] [--folder-id parent_folder_id]
```

Input options

- *filename.json* specifies an optional JSON file that contains the definition. Alternatively, you can provide the details through Stdin. See [“Specify Standard Input” on page 62](#).
- *parent_folder_id* specifies an ID to an optional parent folder on the SAS Viya platform in which to save the job-request definition.

Note: The `--folder-id` option is available for this command as of release 2.7.1 of the job CLI plug-in. To acquire the folder ID to use for *parent_folder_id*, use the **Content** page in SAS Environment Manager or the folders CLI plug-in. See [“Folders: How To” in SAS Viya Administration: Folders](#).

Output

- **Success:** Creates the job request and returns an ID for the job request.
- **Failure:** Returns a message indicating incorrect or missing input values.

job requests update

Updates a job request.

```
sas-admin job requests update --id job_request_id [--file-in filename.json]
```

TIP See “Obtain IDs That Are Used as Input Values” on page 58.

Input options

- *job_request_id* specifies the ID of the job request.
- *filename.json* specifies an optional JSON file that contains the updated definition. Alternatively, you can provide the details through Stdin. See “Specify Standard Input” on page 62.

Output

- **Success:** Updates the specified job request.
- **Failure:** Returns a message indicating incorrect or missing input values.

job requests delete

Deletes a job request.

```
sas-admin job requests delete --id job_request_id
```

TIP See “Obtain IDs That Are Used as Input Values” on page 58.

Input options

job_request_id specifies the ID of the job request.

Output

- **Success:** Deletes the specified job request.
- **Failure:** Returns a message indicating incorrect or missing input values.

job requests list

Lists information about either of the following items:

- job requests that are defined
- job requests that are scheduled for execution

Note: The output is limited to the items that are visible to you.

```
sas-admin job requests list [--scheduled] [--start record] [--limit limit_number] [--sort-by resource_name [,resource_name]] [--filter expression]
```

TIP See “Obtain IDs That Are Used as Input Values” on page 58.

Input options

- `--scheduled` indicates to return information about job requests that are scheduled for execution instead of job requests that are defined.

Note: As of release 2.6.35 of the job CLI plug-in, use the `--scheduled` input option. It replaces the `scheduled` subcommand that is deprecated.

- `record` specifies the record from which to start displaying items. The default is to start from the first record.
- `limit_number` specifies the maximum number of items to display. The default value is 10.
- `resource_name [, resource_name]` specifies to sort the displayed items by one or more resource names.
 - Separate multiple values with commas.
 - The default sort order is ascending.
 - To sort in descending order, put the `~` (tilde) character before the resource name.
- `expression` specifies to filter the output using the specified regular expression. For more information, see “Specify a Filter for Command Output” on page 60.

Output

- **Success:** If `--scheduled` is not specified, returns a list of job requests that are defined. The output includes the job-request ID, name, description, and more.
- **Success:** If `--scheduled` is specified, returns a list of job requests that are scheduled for execution. The output includes the scheduled-job-request ID, name, description, and more.

Note: The scheduled-job-request ID is another ID that is separate and distinct from the job-request ID.

- **Failure:** Returns a message indicating incorrect or missing input values.

job requests show

Displays detailed information about either of the following items:

- a job request that is currently defined
- a job request that is scheduled for execution

```
sas-admin job requests show [--scheduled --id scheduled_job_request_id] [--id job_request_id]
```

Note: You must specify either `job_request_id` or `scheduled_job_request_id`, but not both.

The scheduled-job-request ID is another ID that is separate and distinct from the job-request ID.

TIP See “Obtain IDs That Are Used as Input Values” on page 58.

Input options

- `--scheduled` indicates to return information about job requests that are scheduled for execution instead of job requests that are defined.

Note: As of release 2.6.35 of the job CLI plug-in, use the `--scheduled` input option. It replaces the `scheduled` subcommand that is deprecated.

- `scheduled_job_request_id` specifies the ID of the scheduled job request. The scheduled-job-request ID is another ID that is separate and distinct from the job-request ID.

Note: When `--scheduled` is specified, the scheduled-job-request ID is required.

- `job_request_id` specifies the ID of the job request.

Note: When `--scheduled` is not specified, the job-request ID is required.

Output

- **Success:** If `--scheduled` is not specified, returns the details of the specified job request that is currently defined. The output includes the job-request ID, name, description, version, job definition, and more.
- **Success:** If `--scheduled` is specified, returns the details of the specified job request that is scheduled for execution. The output includes the scheduled-job-request ID, name, description, version, job definition, and more.

Note: The scheduled-job-request ID is another ID that is separate and distinct from the job-request ID.

- **Failure:** Returns a message indicating incorrect or missing input values.

job requests execute

About This Command

Executes a job request. The command sends the job request to the Job Execution Service (JES). For details about this process, see “Execute a Job Request” on page 6.

Note: SAS Environment Manager does not send the job request to the Job Execution Service (JES). Instead, SAS Environment Manager sends the job request to the Scheduler Service with instructions to schedule the job request for immediate execution. Then, the Scheduler Service calls the JES to execute the job request.

Note: The `--wait-results` option is available as of release 2.6.44 of the job CLI plug-in.

This option enables the job CLI plug-in to wait for the results from the program associated with the job request. The CLI polls the program until the CLI receives the program results. The results are written to stdout. For an example of using this option, see [“Capture Results from a Program or Command” on page 54](#).

Syntax

```
sas-admin job requests execute --id job_request_id [--wait-results] [--domain-id domain_name]
```

TIP See [“Obtain IDs That Are Used as Input Values” on page 58](#).

Input options

- *job_request_id* specifies the ID of the job request.
- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See [“Using a Group-Managed Service Account” on page 117](#).

Note: The `--domain-id` option is available as of release 2.6.12 of the job CLI plug-in.

Output

- **Success:** Executes the specified job request.

When `--wait-results` is specified, the results are returned from the program associated with the job request. The results are written to stdout in the following format:

```
"State: state
```

Where `state` is the termination status from the program. This format enables the parsing of the results.

- **Failure:** Returns a message indicating incorrect or missing input values.

job requests schedule

About This Command

Schedules a job request. The command sends the job request to the Scheduler Service. For details about this process, see [“Schedule a Job Request” on page 7](#).

This command requires a time-based trigger for input. See [“Time-Based Trigger Definition” on page 111](#).

Syntax

```
sas-admin job requests schedule --id job_request_id [--format event_format] --file-in filename.json [--domain-id domain_name]
```


TIP See “Obtain IDs That Are Used as Input Values” on page 58.

Input options

- *job_request_id* specifies the ID of the job request.
- *event_format* specifies the format of the trigger events in the JSON file.
 - Valid values are `passThrough` or `portable`.
 - If you do not specify the `--format` option, `portable` is the default.
 - If you specify a value for `--file-in`, you must specify the `--format`.
- *filename.json* specifies a JSON file that contains the definition for one or more time-based triggers. For details, see “Define a Time-Based Trigger” on page 46. Alternatively, you can provide the details through Stdin. See “Specify Standard Input” on page 62.

Note: You must specify a file name when you specify a value for *event_format*.

- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See “Using a Group-Managed Service Account” on page 117.

Note: The `--domain-id` option is available as of release 2.6.12 of the job CLI plug-in.

Output

- **Success:** Schedules the specified job request. The output includes the scheduled-job-request ID.

Note: The scheduled-job-request ID is another ID that is separate and distinct from the job-request ID.

- **Failure:** Returns a message indicating incorrect or missing input values.

job requests unschedule

Unscheduled a job request.

```
sas-admin job requests unschedule --id scheduled_job_request_id [--domain-id domain_name]
```

TIP See “Obtain IDs That Are Used as Input Values” on page 58.

Input options

- *scheduled_job_request_id* specifies the ID of the scheduled job request to unschedule.
- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See “Using a Group-Managed Service Account” on page 117.

Note: The `--domain-id` option is available as of release 2.6.12 of the job CLI plug-in.

Output

- **Success:** Unschedules the specified job request.
- **Failure:** Returns a message indicating incorrect or missing input values.

job requests list-history

Lists the history of past executions of a job request. You can specify that the command generate output for either of these items:

- a job request that is defined. It was executed, but not via a schedule.
- a job request that is scheduled for execution. Its schedule can be in process or complete.

```
sas-admin job requests list-history [--scheduled --id scheduled_job_request_id] [--id
job_request_id] [--start record] [--limit limit_number] [--sort-by resource_name
[,resource_name]] [--filter expression] [--domain-id domain_name]
```

Note: You must specify either *job_request_id* or *scheduled_job_request_id*, but not both.

The scheduled-job-request ID is another ID that is separate and distinct from the job-request ID.

TIP See “Obtain IDs That Are Used as Input Values” on page 58.

Input options

- `--scheduled` indicates to return information about job requests that are scheduled for execution instead of job requests that are defined.

Note: As of release 2.6.35 of the job CLI plug-in, use the `--scheduled` input option. It replaces the `scheduled` subcommand that is deprecated.

- *scheduled_job_request_id* specifies the ID of the scheduled job request. The scheduled-job-request ID is another ID that is separate and distinct from the job-request ID.

Note: When `--scheduled` is specified, the scheduled-job-request ID is required.

- *job_request_id* specifies the ID of the defined job request whose history you want returned.

Note: When `--scheduled` is not specified, the job-request ID is required.

- *record* specifies the record from which to start displaying items. The default is to start from the first record.
- *limit_number* specifies the maximum number of items to display. The default value is 10.
- *resource_name* [, *resource_name*] specifies to sort the displayed items by one or more resource names.

- Separate multiple values with commas.
- The default sort order is ascending.
- To sort in descending order, put the ~ (tilde) character before the resource name.
- *expression* specifies to filter the output using the specified regular expression. For more information, see [“Specify a Filter for Command Output” on page 60](#).
- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See [“Using a Group-Managed Service Account” on page 117](#).

Note: The `--domain-id` option is available for this command as of release 2.6.29 of the job CLI plug-in.

Output

- **Success:** If `--scheduled` is not specified, returns the history of the specified defined job request.
- **Success:** If `--scheduled` is specified, returns the history of the specified scheduled job request.
- **Failure:** Returns a message indicating incorrect or missing input values.

job requests show-history

Lists the history details (including the state) for either of the following items:

- an instance of a job request that can be in any of the following states: running (executing), pending, canceled, failed, complete, suspended, or on hold.
- an instance of a job request that is scheduled for execution. The instance can be running, successfully complete, or failed.

```
sas-admin job requests show-history --instance-id job_request_instance_id [--sch-id
scheduled_job_request_instance_id] [--domain-id domain_name]
```

TIP See [“Obtain IDs That Are Used as Input Values” on page 58](#).

Input options

- *job_request_instance_id* specifies the ID of the job-request instance whose history details you want to view.
- *scheduled_job_request_instance_id* specifies the ID of the scheduled job-request instance whose history you want to view.

Note: The `--sch-id` option is available as of release 2.6.5 of the job CLI plug-in.

- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See [“Using a Group-Managed Service Account” on page 117](#).

Note: The `--domain-id` option is available for this command as of release 2.6.29 of the job CLI plug-in.

Output

- **Success:** If `--sch-id` is not specified, returns the details about the history of the specified job-request instance. The output can include the ID of the history record, state, results, error, job request, and more.
- **Success:** If `--sch-id` is specified, returns the details about the history of the specified scheduled-job-request instance. The output can include the ID of the history record, state, exit code, job status, and more.
- **Failure:** Returns a message indicating incorrect or missing input values.

job requests show-status

Shows the detailed status for a job-request instance. You can specify that the command generate output for either of these items:

- a job-request instance that is defined. It was executed, but not via a schedule.
- a job-request instance that is scheduled for execution. Its schedule can be in process or complete.

Note: This command is available as of release 2.6.17 of the job CLI plug-in.

```
sas-admin job requests show-status [--scheduled --id scheduled_job_request_id --instance-id
scheduled_job_request_instance_id] [--instance-id job_request_instance_id] [--domain-id
domain_name]
```

TIP See “Obtain IDs That Are Used as Input Values” on page 58.

Input options

- `--scheduled` indicates to return information about job requests that are scheduled for execution instead of job requests that are defined.

Note: The `--scheduled` input option is available as of release 2.6.40 of the job CLI plug-in.

- `scheduled_job_request_id` specifies the ID of the scheduled job request. The scheduled-job-request ID is another ID that is separate and distinct from the job-request ID.

Note: When `--scheduled` is specified, the scheduled-job-request ID is required.

- `scheduled_job_request_instance_id` specifies the ID of a job-request instance that is scheduled for execution and whose status you want to view.

Note: When `--scheduled` is specified, the scheduled-job-request-instance ID is required.

- *job_request_instance_id* specifies the ID of the job-request instance whose status you want to view.

Note: When `--scheduled` is not specified, the job-request-instance ID is required.

- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See [“Using a Group-Managed Service Account” on page 117](#).

Note: The `--domain-id` option is available for this command as of release 2.6.29 of the job CLI plug-in.

Output

- **Success:** If `--scheduled` is not specified, returns the status of the specified instance of the specified job request. The output can include the ID, message, state, exit code, and more.
- **Success:** If `--scheduled` is specified, returns the status of the specified instance of the specified scheduled job request. The output can include the ID, message, state, exit code, and more.
- **Failure:** Returns a message indicating incorrect or missing input values.

Actions Commands

Use these commands to manage actions that have been registered with the Job-Flow-Scheduling Service, which then makes the actions available for inclusion in a job flow.

job actions generate-template

Generates a template for an action definition. Use the generated template to specify the action that you want to create with the command `job actions create`. For details, see [“Job-Action Definition” on page 99](#).

```
sas-admin job actions generate-template [--file-out filename.json]
```

Input options

filename.json specifies the name of an optional JSON file to contain the generated template. For details, see [“Create a Job Action for Use in a Job Flow” on page 40](#).

TIP The command saves the file on the system that is running the CLI. When you specify the file name, include a path for its location on that system. If you do not specify a path, the file is written to the system’s working directory. You can identify the working directory by issuing the `pwd` command.

Output

- by default, writes the generated template to Stdout
- if `--file-out` is specified, returns the generated template in the specified JSON file

TIP The file is saved to the system that is running the CLI, not the server to which the CLI commands are issued.

job actions create

Creates an action. For details, see [“Create a Job Action for Use in a Job Flow”](#) on page 40.

```
sas-admin job actions create [--file-in filename.json] [--folder-id parent_folder_id]
```

Input options

- *filename.json* specifies an optional JSON file that contains the definition. Alternatively, you can provide the details through Stdin. See [“Specify Standard Input”](#) on page 62.
- *parent_folder_id* specifies an ID to an optional parent folder on the SAS Viya platform in which to save the job-action definition.

Note: The `--folder-id` option is available for this command as of release 2.7.1 of the job CLI plug-in. To acquire the folder ID to use for *parent_folder_id*, use the **Content** page in SAS Environment Manager or the folders CLI plug-in. See [“Folders: How To”](#) in *SAS Viya Administration: Folders*.

Output

- **Success:** Returns an ID for the action.
- **Failure:** Returns a message indicating incorrect or missing input values.

job actions update

Updates an action.

```
sas-admin job actions update --id action_id [--file-in filename.json]
```

Input options

- *action_id* specifies the ID of the action.
- *filename.json* specifies an optional JSON file that contains the updated definition. Alternatively, you can provide the details through Stdin. See [“Specify Standard Input”](#) on page 62.

Output

- **Success:** Updates the specified action.
- **Failure:** Returns a message indicating incorrect or missing input values.

job actions delete

Deletes an action.

```
sas-admin job actions delete --id action_id
```

Input options

action_id specifies the ID of the action.

Output

- **Success:** Deletes the specified action.
- **Failure:** Returns a message indicating incorrect or missing input values.

job actions list

Lists actions.

```
sas-admin job actions list [--start record] [--limit limit_number] [--sort-by resource_name
[,resource_name]] [--filter expression]
```

Input options

- *record* specifies the record from which to start displaying items. The default is to start from the first record.
- *limit_number* specifies the maximum number of items to display. The default value is 10.
- *resource_name* [, *resource_name*] specifies to sort the displayed items by one or more resource names.
 - Separate multiple values with commas.
 - The default sort order is ascending.
 - To sort in descending order, put the ~ (tilde) character before the resource name.
- *expression* specifies to filter the output using the specified regular expression. For more information, see [“Specify a Filter for Command Output” on page 60](#).

Output

Returns a list of actions. The output can include each action’s ID, name, description, and more.

job actions show

Displays details about an action.

```
sas-admin job actions show --id action_id
```

Input options

action_id specifies the ID of the action.

Output

- **Success:** Returns details about the specified action. The output can include ID, name, description, request, and more.
- **Failure:** Returns a message indicating incorrect or missing input values.

Flows Commands

Use these commands to manage flows that are registered with Job-Flow-Scheduling Service.

job flows generate-template

Generates a template for a job-flow definition. Use the template to specify the job flow that you want to create with the command `job flows create`. For details, see [“Job-Flow Definition Template” on page 103](#).

```
sas-admin job flows generate-template [--file-out filename.json]
```

Input options

filename.json specifies the name of an optional JSON file to contain the generated template. For details, see [“Create a Job Flow and Its Definition” on page 41](#).

TIP The command saves the file on the system that is running the CLI. When you specify the file name, include a path for its location on that system. If you do not specify a path, the file is written to the system’s working directory. You can identify the working directory by issuing the `pwd` command.

Output

- by default, writes the generated template to Stdout
- if `--file-out` is specified, returns the generated template in the specified JSON file

TIP The file is saved to the system that is running the CLI, not the server to which the CLI commands are issued.

job flows create

Creates a job flow. The job flow is specified by the definition that you create and submit with this command. For details, see [“Create a Job Flow and Its Definition” on page 41](#).

```
sas-admin job flows create [--file-in filename.json] [--folder-id parent_folder_id]
```

Input options

- *filename.json* specifies an optional JSON file that contains the definition. Alternatively, you can provide the details through Stdin. See [“Specify Standard Input” on page 62](#).
- *parent_folder_id* specifies an ID to an optional parent folder on the SAS Viya platform in which to save the job-flow definition.

Note: The `--folder-id` option is available for this command as of release 2.7.1 of the job CLI plug-in. To acquire the folder ID to use for *parent_folder_id*, use the **Content** page in SAS Environment Manager or the folders CLI plug-in. See [“Folders: How To” in SAS Viya Administration: Folders](#).

Output

- **Success:** Returns an ID for the job flow.
- **Failure:** Returns a message indicating incorrect or missing input values.

job flows update

Updates a job flow. The job flow is specified by the definition that you update and submit with this command. For details, see [“Update a Job Flow” on page 43](#).

Note:

- A job flow can be updated only by its creator or an administrator.
- A member of a group-managed service account can update a flow that was scheduled by that account. However, the update is limited to the triggers.

```
sas-admin job flows update --id job_flow_id [--file-in filename.json] [--domain-id domain_name]
```

TIP See [“Obtain IDs That Are Used as Input Values” on page 58](#).

Input options

- *job_flow_id* specifies the ID of the job flow.
- *filename.json* specifies an optional JSON file that contains the updated definition. For details, see [“An Existing Job-Flow Definition” on page 105](#). Alternatively, you can provide the details through Stdin. See [“Specify Standard Input” on page 62](#).
- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See [“Using a Group-Managed Service Account” on page 117](#).

Note: The `--domain-id` option is available for this command as of release 2.6.29 of the job CLI plug-in.

Output

- **Success:** Updates the specified job flow.
- **Failure:** Returns a message indicating incorrect or missing input values.

job flows delete

Deletes a job flow.

```
sas-admin job flows delete --id job_flow_id
```

TIP See [“Obtain IDs That Are Used as Input Values” on page 58](#).

Input options

- *job_flow_id* specifies the ID of the job flow.

Output

- **Success:** Deletes the specified job flow.
- **Failure:** Returns a message indicating incorrect or missing input values.

job flows list

Lists all job flows.

```
sas-admin job flows list [--start record] [--limit limit_number] [--sort-by
resource_name[,resource_name]] [--filter expression]
```

Input options

- *record* specifies the record from which to start displaying items. The default is to start from the first record.
- *limit_number* specifies the maximum number of items to display. The default value is 10.
- *resource_name* [*, resource_name*] specifies to sort the displayed items by one or more resource names.
 - Separate multiple values with commas.
 - The default sort order is ascending.
 - To sort in descending order, put the ~ (tilde) character before the resource name.
- *expression* specifies to filter the output using the specified regular expression. For more information, see [“Specify a Filter for Command Output” on page 60](#).

Output

Returns a list of job flows including each associated job-flow ID, name, version, and description.

job flows show

Displays details about a job flow.

```
sas-admin job flows show --id job_flow_id
```

TIP See [“Obtain IDs That Are Used as Input Values” on page 58](#).

Input options

job_flow_id specifies the ID of the job flow.

Output

- **Success:** Returns the existing job-flow definition for the specified job flow. The definition is returned in JSON format via Stdout. The output includes the job flow ID, name, description, trigger type, triggers, trigger conditions, jobs, and more. For details, see [“An Existing Job-Flow Definition” on page 105](#).

TIP Pipe the definition into a file to save the JSON.

Note: When a job flow is scheduled, the returned JSON contains the job-flow-definition ID in the `definitionId` field. If the job flow is not scheduled, this field is empty and omitted from the returned JSON.

- **Failure:** Returns a message indicating incorrect or missing input values.

job flows execute

About This Command

Executes a job flow. The command sends the job flow to the Job-Flow-Scheduling Service (JFS). For details about this process, see [“Submit a Job Flow” on page 7](#) and [“Execute a Job Flow Immediately” on page 8](#).

Note: The `--wait-results` option is available as of release 2.6.44 of the job CLI plug-in.

This option enables the job CLI plug-in to wait for the job flow to reach a terminated state. The CLI receives the results from the program associated with a job request or command-line command in the job flow. The CLI polls the program until the CLI receives the program results. The results are written to stdout. For an example of using this option, see [“Capture Results from a Program or Command” on page 54](#).

Syntax

```
sas-admin job flows execute --id job_flow_id [--wait-results] [--domain-id domain_name]
```

TIP See [“Obtain IDs That Are Used as Input Values” on page 58](#).

Input options

- *job_flow_id* specifies the ID of the job flow.

Note: If you specify a job flow with a trigger, the trigger is ignored.

- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See [“Using a Group-Managed Service Account” on page 117](#).

Note: The `--domain-id` option is available as of release 2.6.12 of the job CLI plug-in.

Output

- **Success:** Executes the specified job flow.

When `--wait-results` is specified, the results are returned from the program associated with a job request within the job flow. The results are written to stdout in the following format:

```
"ExitCode: n - State: state
```

Where `n` is the return code from the program and `state` is the termination status from the program. This format enables parsing of the results.

- **Failure:** Returns a message indicating incorrect or missing input values.

job flows schedule

About This Command

Schedules a job flow. The command sends the job flow to the Job-Flow-Scheduling Service (JFS). For details about this process, see [“Submit a Job Flow” on page 7](#) and [“Schedule a Job Flow with a Time-Event Trigger” on page 8](#).

Syntax

```
sas-admin job flows schedule --id job_flow_id [--domain-id domain_name]
```

TIP See [“Obtain IDs That Are Used as Input Values” on page 58](#).

Input options

- *job_flow_id* specifies the ID of the job flow.
- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See [“Using a Group-Managed Service Account” on page 117](#).

Note: The `--domain-id` option is available as of release 2.6.12 of the job CLI plug-in.

Output

- **Success:** Schedules the specified job flow.
- **Failure:** Returns a message indicating incorrect or missing input values.

job flows reschedule

Reschedules a job flow.

```
sas-admin job flows reschedule --id job_flow_id [--domain-id domain_name]
```

TIP See [“Obtain IDs That Are Used as Input Values” on page 58](#).

Input options

- *job_flow_id* specifies the ID of the job flow.
- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See [“Using a Group-Managed Service Account” on page 117](#).

Note: The `--domain-id` option is available as of release 2.6.12 of the job CLI plug-in.

Output

- **Success:** Reschedules the specified job flow.
- **Failure:** Returns a message indicating incorrect or missing input values.

job flows unschedule

Unscheduler a job flow.

Note: Unlike SAS Environment Manager, this CLI command does not delete the triggers associated with the job flow. Use this command as an alternative to the `unschedule` option in SAS Environment Manager.

This command does delete the job-flow history.

```
sas-admin job flows unschedule --id job_flow_id [--domain-id domain_name]
```

TIP See “Obtain IDs That Are Used as Input Values” on page 58.

Input options

- *job_flow_id* specifies the ID of the job flow.
- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See “Using a Group-Managed Service Account” on page 117.

Note: The `--domain-id` option is available as of release 2.6.12 of the job CLI plug-in.

Output

- **Success:** Unscheduler the specified job flow.
- **Failure:** Returns a message indicating incorrect or missing input values.

job flows list-history

About This Command

Lists the history of past executions of a job flow. You can specify that the command generate output for either of these scenarios:

- a history of only the instances of scheduled flows that have been executed
- a combined history of the following instances:
 - instances of the scheduled flows that have been executed
 - instances of the flows that have been executed immediately (that is, not via a schedule).

TIP A job flow can be executed immediately by using either of these methods:

- the “[job flows execute](#)” command in the job CLI plug-in
- “[Execute a Job Flow Immediately](#)”

Syntax

```
sas-admin job flows list-history --id job_flow_id [--allExecutedInstances | --all] [--domain-id domain_name] [--start record] [--limit limit_number] [--sort-by resource_name [, resource_name]] [--filter expression]
```

TIP See “[Obtain IDs That Are Used as Input Values](#)” on page 58.

Input options

- *job_flow_id* specifies the ID of the job flow.
- `--allExecutedInstances` or `--all` indicates to return a combined history of flow instances that have been executed (either immediately or via a schedule).

Note: The option is available as of release 2.6.47 of the job CLI plug-in.

- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See “[Using a Group-Managed Service Account](#)” on page 117.

Note: The `--domain-id` option is available for this command as of release 2.6.29 of the job CLI plug-in.

- *record* specifies the record from which to start displaying items. The default is to start from the first record.
- *limit_number* specifies the maximum number of items to display. The default value is 10.
- *resource_name* [, *resource_name*] specifies to sort the displayed items by one or more resource names.
 - Separate multiple values with commas.
 - The default sort order is ascending.
 - To sort in descending order, put the ~ (tilde) character before the resource name.
- *expression* specifies to filter the output using the specified regular expression. For more information, see “[Specify a Filter for Command Output](#)” on page 60.

Output

- **Success:** If `--allExecutedInstances` is not specified, returns the history of only the instances of scheduled flows that have been executed. The output can include the job-flow-instance ID, the job-flow name, the start time and completed time of the job-flow instance, its state, and exit code.

- **Success:** If `--allExecutedInstances` is specified, returns a combined history of instances of the scheduled flows that have been executed and instances of the flows that have been executed immediately.
- **Failure:** Returns a message indicating incorrect or missing input values.

job flows show-history

Displays the history details for the specified instance of the specified job flow.

```
sas-admin job flows show-history --id job_flow_id --instance-id job_flow_instance_id [--domain-id domain_name]
```

TIP See “Obtain IDs That Are Used as Input Values” on page 58.

Input options

- *job_flow_id* specifies the ID of the job flow.
- *job_flow_instance_id* specifies the ID of the job-flow instance whose history details you want to view.

Note: This option was renamed to `--instance-id` as of release 2.6.5 of the job CLI plug-in. The prior name (`--sch-id`) is supported, but deprecated.

- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See “Using a Group-Managed Service Account” on page 117.

Note: The `--domain-id` option is available for this command as of release 2.6.29 of the job CLI plug-in.

Output

- **Success:** Returns the history details of the specified instance of the specified job flow. The output can include the job-flow ID, name, job-flow-instance ID, the start and end time of the job-flow instance, its state, exit code, and more.
- **Failure:** Returns a message indicating incorrect or missing input values.

job flows show-history-tree

Shows the flow history for a specified instance of a job flow.

- Shows all of the history records for the job actions and subflows that are associated with the specified instance of the job flow.
- Sorts the output by start time.

TIP Sorted output enables you to see whether dependencies have run in the correct order.

Note: This command is available as of release 2.6.35 of the job plug-in to the sas-admin CLI. See [“Obtain the History of an Executed Job Flow” on page 53](#) for an example of using this command.

```
sas-admin job flows show-history-tree --id job_flow_id --instance-id job_flow_instance_id
```

TIP See [“Obtain IDs That Are Used as Input Values” on page 58](#).

Input options

- *job_flow_id* specifies the ID of the job flow.
- *job_flow_instance_id* specifies the ID of the job-flow instance whose history records you want to view.

Output

- **Success:** Returns the history of the specified instance of the specified job flow.
 - The output includes all of the history records for the job actions and subflows that are associated with the specified instance of the job flow.
 - The output is sorted by start time.
- **Failure:** Returns a message indicating incorrect or missing input values.

job flows show-status

Shows the status for a job-flow instance.

Note: This command is available as of release 2.6.17 of the job CLI plug-in.

```
sas-admin job flows show-status --id job_flow_id --instance-id job_flow_instance_id [--domain-id domain_name]
```

TIP See [“Obtain IDs That Are Used as Input Values” on page 58](#).

Input options

- *job_flow_id* specifies the ID of the job flow.
- *job_flow_instance_id* specifies the ID of the job-flow instance whose status you want to view.
- *domain_name* specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See [“Using a Group-Managed Service Account” on page 117](#).

Note: The `--domain-id` option is available for this command as of release 2.6.29 of the job CLI plug-in.

Output

- **Success:** Returns the status of the specified instance of the specified job flow.
- **Failure:** Returns a message indicating incorrect or missing input values.

job flows trigger

Triggers an instance of a scheduled job flow. The `job flows trigger` command enables you to cause a job flow to run manually. This is the equivalent of the **Run Now** option on the **Jobs and Flows** page in SAS Environment Manager. See [“Trigger a Job Flow with a Manual Trigger” on page 44](#).
`sas-admin job flows trigger --id job_flow_definition_id --sch scheduler_id [--domain-id domain_name]`

Input options

- `job_flow_definition_id` specifies the ID of the job-flow definition. This ID is returned in the JSON that contains the job-flow definition. For details, see [“An Existing Job-Flow Definition” on page 105](#). You can use `job flows show` to retrieve the job-flow-definition JSON. See [“job flows show” on page 82](#).
- `scheduler_id` specifies the ID of the scheduler of the specified job flow.
- `domain_name` specifies the name of an authentication domain. This option enables a group-managed service account to use this command. See [“Using a Group-Managed Service Account” on page 117](#).

Note: The `--domain-id` option is available as of release 2.6.12 of the job CLI plug-in.

Output

- **Success:** Attempts to trigger the specified flow instance. To determine whether the flow instance was triggered, check the job-flow instance status or the job-flow instance history for the execution status. See [“job flows show-status” on page 88](#) and [“job flows show-history” on page 87](#).
- **Failure:** Returns a message indicating incorrect or missing input values.

Schedulers Commands

Considerations

Typically, you do not create a scheduler. The Job-Flow-Scheduling Service creates a scheduler as part of the service’s start-up process.

For details about the available scheduler, see [“How Jobs and Flows Work” on page 4](#).

job schedulers generate-template

Generates a template for a scheduler definition. Use the generated template to define the scheduler that you want to create with the command `job schedulers create`.

Note: Typically, you do not create a scheduler. The Job-Flow-Scheduling Service creates a scheduler as part of the service's start-up process.

```
sas-admin job schedulers generate-template [--file-out filename.json]
```

Input options

filename.json specifies the name of an optional JSON file to contain the generated template. For details, see [“Job-Scheduler Definition” on page 110](#).

TIP The command saves the file on the system that is running the CLI. When you specify the file name, include a path for its location on that system. If you do not specify a path, the file is written to the system's working directory. You can identify the working directory by issuing the `pwd` command.

Output

- by default, writes the generated template to Stdout
- if `--file-out` is specified, returns the generated template in the specified JSON file

TIP The file is saved to the system that is running the CLI, not the server to which the CLI commands are issued.

job schedulers create

Creates a scheduler.

Note: Typically, you do not create a scheduler. The Job-Flow-Scheduling Service creates a scheduler as part of the service's start-up process.

```
sas-admin job schedulers create [--file-in filename.json]
```

Input options

filename.json specifies an optional JSON file that contains the definition. Alternatively, you can provide the details through Stdin. See [“Specify Standard Input” on page 62](#).

Output

- **Success:** Creates the scheduler.
- **Failure:** Returns a message indicating incorrect or missing input values.

job schedulers update

Updates a scheduler definition.

```
sas-admin job schedulers update --id scheduler_id [--file-in filename.json]
```

Input options

- *scheduler_id* specifies the ID of the scheduler.
- *filename.json* specifies an optional JSON file that contains the updated definition. Alternatively, you can provide the details through Stdin. See [“Specify Standard Input” on page 62](#).

Output

- **Success:** Updates the specified scheduler.
- **Failure:** Returns a message indicating incorrect or missing input values.

job schedulers delete

Deletes a scheduler.

```
sas-admin job schedulers delete --id scheduler_id
```

Input options

scheduler_id specifies the ID of the scheduler.

Output

- **Success:** Deletes the specified scheduler.
- **Failure:** Returns a message indicating incorrect or missing input values.

job schedulers list

Lists details about schedulers.

```
sas-admin job schedulers list [--start record] [--limit limit_number] [--sort-by resource_name[,resource_name]] [--filter expression]
```

Input options

- *record* specifies the record from which to start displaying items. The default is to start from the first record.
- *limit_number* specifies the maximum number of items to display. The default value is 10.
- *resource_name* [, *resource_name*] specifies to sort the displayed items by one or more resource names.
 - Separate multiple values with commas.
 - The default sort order is ascending.
 - To sort in descending order, put the ~ (tilde) character before the resource name.
- *expression* specifies to filter the output using the specified regular expression. For more information, see [“Specify a Filter for Command Output” on page 60](#).

Output

Returns a list of schedulers including each associated scheduler ID.

job schedulers show

Displays details about a specified scheduler.

```
sas-admin job schedulers show --id scheduler_id
```

Input options

scheduler_id specifies the ID of the scheduler.

Output

- **Success:** Returns details about the specified scheduler including its ID, name, description, host, port, and more.
- **Failure:** Returns a message indicating incorrect or missing input values.

job schedulers pause

About This Command

Pauses the schedules of the following objects:

- all scheduled job-flow instances
- all scheduled job-request instances

Note: A job request is scheduled by using the `job requests schedule` command.

When scheduled, these objects were saved by the Scheduler Service in the SAS Viya platform. The pause affects instances whose schedules have not yet occurred.

IMPORTANT If other applications or SAS Viya platform services save jobs via the Scheduler Service, those jobs are affected too.

Note:

- You must be a SAS administrator to use this command.
 - This command is useful when performing a tenant migration. See [“Pause the Scheduler Before Migration” on page 56](#).
 - This command does not affect job flows that are scheduled via file-based events.
 - This command is available as of release 2.5.1 of the job CLI.
-

Syntax

```
sas-admin job schedulers pause
```

Output

- If successful, returns a message indicating success. The job CLI exits with a value of 0.
- If the Scheduler Service has an error, that error message is returned. The job CLI exits with a value of 1.

job schedulers resume

About This Command

Resumes the schedules of the following objects:

- all scheduled job-flow instances
- all scheduled job-request instances

Note: A job request is scheduled by using the `job requests schedule` command.

When scheduled, these objects were saved by the Scheduler Service in the SAS Viya platform. The resume affects instances whose schedules have not yet occurred.

IMPORTANT If other applications or SAS Viya platform services save jobs via the Scheduler Service, those jobs are affected too.

Note:

- You must be a SAS administrator to use this command.
 - This command is useful when performing a tenant migration. See [“Pause the Scheduler Before Migration” on page 56](#).
 - This command does not affect job flows that are scheduled via file-based events.
 - This command is available as of release 2.5.1 of the job CLI.
-

Syntax

```
sas-admin job schedulers resume
```

Output

- If successful, returns a message indicating success. The job CLI exits with a value of 0.
- If the Scheduler Service has an error, that error message is returned. The job CLI exits with a value of 1.

job schedulers restore

About This Command

Restores the schedules of the following objects:

- all scheduled job-flow instances
- all scheduled job-request instances

Note: A job request is scheduled by using the `job requests schedule` command.

When scheduled, these objects were saved by the Scheduler Service in the SAS Viya platform. The restoration affects all instances of these objects that are scheduled in the future. The instances that are complete are not affected.

IMPORTANT If other applications or SAS Viya platform services save jobs via the Scheduler Service, those jobs are affected too.

When you run the job CLI `job scheduler restore` command, the `job scheduler pause` and `job scheduler resume` commands are also performed. The scheduler is unaffected if you do either of the following actions:

- Explicitly run the `job scheduler pause` command before the `job scheduler restore` command.
- Explicitly run the `job scheduler resume` command after the `job scheduler restore` command.

Note:

- You must be a SAS administrator to use this command.
 - This command is useful when performing a tenant migration. See [“Pause the Scheduler Before Migration” on page 56](#).
 - This command is useful when adjusting schedules for daylight saving time. See [“Adjust for Daylight Saving Time” on page 57](#).
 - This command does not affect job flows that are scheduled via file-based events.
 - This command is available as of release 2.5.1 of the job CLI.
-

Syntax

```
sas-admin job schedulers restore
```

Output

- If successful, returns a JSON structure that contains the scheduled objects. The job CLI exits with a value of 0.

- If the Scheduler Service has an error, that error message is returned. The job CLI exits with a value of 1.

Webapp Command

About This Command

Issues a call to the SAS Job Execution REST API. This API provides functionality to the SAS Job Execution Web Application. This application executes a job request when it receives a parameter indicating the location of the job request. The webapp command adds the parameter to a URL in the following way:

```
http://host:port/SASJobExecution/?input_string
```

where *input_string* indicates the location of the job-request definition.

For detailed information, see these topics:

- [“Executing Jobs” in SAS Job Execution Web Application: User’s Guide](#)
- [“Using Job Input Parameters” in SAS Job Execution Web Application: User’s Guide](#)
- [“Working with the SAS Job Execution Web Application” in SAS Job Execution Web Application: User’s Guide](#)

Syntax

```
sas-admin job webapp execute --query-string input_string
```

Input options

input_string indicates the location of the job-request definition that you want to execute by using the SAS Job Execution REST API.

Output

The command returns the same output that the web application generates, including the raw HTML.

Job-Definition Definition

This topic contains detailed descriptions about the job-definition definition template, its key-value pairs, and their settings. For more information, see [“Create a Job Request” on page 38](#).

Job-Definition Definition Template

A template of the job-definition definition is returned by the [“job definitions generate-template”](#) on [page 65](#) command.

Note: When you view an existing definition in the SAS Viya platform, it might contain additional key-value pairs. These do not require input because of one or more of these reasons:

- They are deprecated.
- They are reserved for future use.
- They are assigned a permanent value.

Here is the definition template and its descriptions:

```
{
1  "name": "Replace this string with the name of the job definition.",
2  "description": "(Optional) Replace this string with a description of the job definition.",
3  "type": "Replace this string with the value that indicates the type of code used in the job
definition.",
4  "code": "Replace this string with the code to use in the job definition.",
5  "parameters": {
    "name": "(Optional) Replace this string with the name of the parameter.",
    "type": "(Optional) Replace this string with the parameter type.",
    "defaultValue": "(Optional) Replace this string with a default value for the parameter.",
    "label": "(Optional) Replace this string with a display label for the parameter.",
    "required": "(Optional) Replace with this string with one of these values: true, false."
  }
6  "properties": {
    "name": "(Optional) Replace this string with the name of the property.",
    "value": "(Optional) Replace this string with the value of the property."
  }
}
```

- 1 Specifies the name for the job definition.
- 2 (Optional) Specifies a description about the job definition.
- 3 Specifies the type of code that is specified in the `code` setting. See [“type”](#) on [page 97](#).
- 4 Specifies the code to use in the job definition.
- 5 (Optional) Specifies parameters that are required for configuration or the code. See [“parameters”](#) on [page 98](#).
- 6 (Optional) Specifies properties that are required for configuration or the code. See [“properties”](#) on [page 99](#).

TIP You can study existing job definitions as examples. New deployments of the SAS Viya platform already contain job definitions.

- 1 Use the “[job definitions list](#)” command to list the available job definitions.
- 2 Select the ID of a job definition whose details you want to study.
- 3 Submit the “[job definitions show](#)” command using that ID.

Example of a Job-Definition Definition

```
{
  "name": "example",
  "description": "Show the contents of sashelp.class using PROC PRINT",
  "type": "Compute",
  "parameters": [
    {
      "name": "_contextName",
      "defaultValue": "SAS Job Execution Compute context",
      "type": "CHARACTER",
      "label": "Context Name",
      "required": false
    },
    {
      "name": "param2",
      "defaultValue": "value",
      "type": "CHARACTER",
      "label": "param2 name",
      "required": false
    }
  ],
  "code": "ods html style=HTMLBlue;\nproc print data=sashelp.class; run; quit;\nods
html close;",
  "properties": [
    {
      "name": "class",
      "value": "admin"
    }
  ]
}
```

Job-Definition Settings and Objects

type

The `type` setting indicates the type of code that is specified in the `code` setting. For example, the `type Compute` indicates SAS code.

Note:

- The type value is case sensitive.
 - `compute` is the provider intended for use by users. Other providers are used by software only.
 - SAS solutions create their own job definitions. Some `type` values are associated with these solutions.
-

code

TIP Use line breaks to separate multiple lines of code.

parameters

The `parameters` key-value pair specifies parameter values that are required by configuration or the code specified in `code`. You can specify multiple parameter objects. For an example, see [“Example of a Job-Definition Definition” on page 97](#).

TIP If your code does not require parameter values, do not include the `parameters` object in the template.

Here is an example of how to specify the `parameters` object:

```
"parameters": {
  "name": "_contextName",
  "defaultValue": "SAS Job Execution compute context",
  "type": "CHARACTER",
  "label": "Context Name",
  "required": false
}
```

- **name** specifies the name of the parameter.
- **type** specifies the type of the parameter.

TIP This value is not validated by the job CLI plug-in. Make sure that this value is valid. If it is not and you run the job definition, the definition will fail.

- **defaultValue** specifies the default value for this parameter. If a value is not specified at the time of execution, this default value is used.
- **label** (Optional) specifies the display value for the parameter. For example, the parameter name is `maxLength` and the display label is `Maximum Length`.
- **required** (Optional) indicates whether the parameter is required by the configuration or code. The default value is `false`.

properties

The properties key-value pair specifies property values for a configuration or the code specified in code.

TIP If your code does not require properties, do not include the properties object in the template.

Here is an example of how to specify properties:

```
"properties": {  
  "name": "class",  
  "value": "admin"  
}
```

- **name** specifies the name of the property expected by the configuration or code.
- **value** specifies the value of the property.

Job-Action Definition

This topic contains detailed descriptions about the job-action definition, its key-value pairs, and their settings. For more information, see [“Create a Job Action for Use in a Job Flow” on page 40](#).

Job-Action Definition Template

A template of the job-action definition is returned by the [“job actions generate-template” on page 77](#) command.

Note: When you view an existing definition in the SAS Viya platform, it might contain additional key-value pairs. These do not require input because of one or more of these reasons:

- They are deprecated.
 - They are reserved for future use.
 - They are assigned a permanent value.
-

Here is the definition template and its descriptions:

```

{
1  "name": "Replace this string with the name of the job action.",
2  "description": "(Optional) Replace this string with a description of the job action.",
3  "jobRequestUri": "(Mutually exclusive with commandLine) Replace this string with the URI
associated with the job request.",
4  "commandLine": "(Mutually exclusive with jobRequestUri) Replace this string with a fully-
qualified command-line command."
}

```

- 1 Specifies the name for the action. See [“Unsupported Characters in Job Action and Job Flow Names” on page 116](#).
- 2 (Optional) Specifies a description for the action.
- 3 Specifies the URI of the job request associated with this action. You can set either `jobRequestUri` or `commandLine`, but not both.
- 4 Specifies the command-line command associated with this action. You can set either `jobRequestUri` or `commandLine`, but not both.

Note:

- A command-line command can fail if the command is not permitted by the SAS Launcher Service allow list. For more information about the allow list feature, see [“Security Considerations” in SAS Viya Administration: Programming Run-Time Servers](#).
 - SAS Launcher Service does not validate whether the command is a valid command to run in the pod. It is the responsibility of the creator of the job action that contains the command to ensure that the command is valid to run.
-

Job-Request Definition

This topic contains detailed descriptions about the job-request definition, its key-value pairs, and their settings. For more information, see [“Create a Job Request” on page 38](#).

Job-Request Definition Template

A template of the job-request definition is returned by the [“job requests generate-template”](#) command.

Note: When you view an existing definition in the SAS Viya platform, it might contain additional key-value pairs. These do not require input because of one or more of these reasons:

- They are deprecated.
 - They are reserved for future use.
 - They are assigned a permanent value.
-

Here is the definition template and its descriptions:

```
{
1 "name": "Replace this string with the name of the job request.",
2 "description": "(Optional) Replace this string with a description of the job request.",
3 "jobDefinitionUri": "(Mutually exclusive with jobDefinition) Replace this string with the URI of
a job definition.",
4 "jobDefinition": "(Mutually exclusive with jobDefinitionUri) Replace this string with a job
definition.",
5 "arguments": {
    "name": "(Optional) Replace this string with the name of an argument that is used when the
job request runs.",
    "value": "(Optional) Replace this string with the value of that argument."
  },
6 "properties": {
    "name": "(Optional) Replace this string with the name of a property associated with the
job request.",
    "value": "(Optional) Replace this string with the value of that property."
  }
}
```

- 1 Specifies the name for the job request. The maximum length permitted is 255 characters.
- 2 (Optional) Specifies a description about the job request.
- 3 Specifies the URI of the job definition to use in this job request. See [“Construct a URI for Use in a Definition” on page 60](#).

Note: If you specify `jobDefinitionUri`, do not specify `jobDefinition`.

- 4 Specifies the job definition to use in this job request. For more information, see [“Job-Definition Definition” on page 95](#).

Note:

- If you specify `jobDefinition`, do not specify `jobDefinitionUri`.
 - If you want to execute or schedule this job request as a member of a group-managed service account, then you must specify `jobDefinition`. It enables use of the `--domain-id` input option.
-

- 5 Specifies arguments that are used when the job request runs. See [“arguments” on page 102](#).
- 6 Specifies properties that are associated with the job request. See [“properties” on page 102](#).

Example of a Job-Request Definition

```
{
  "name": "exampleJob1",
  "jobDefinition": {
    "name": "example1",
    "type": "Compute",
    "code": "ods html style=HTMLBlue;\nproc print data=sashelp.class; run;
quit;\nods html close;"
  }
}
```

```

    "parameters": [
      {
        "name": "_contextName",
        "defaultValue": "SAS Job Execution Compute context",
        "type": "CHARACTER",
        "label": "Context Name",
        "required": false
      }
    ]
  },
  "arguments": {
    "_contextName": "SAS Job Execution Compute context"
  },
  "properties": [
    {
      "name": "class",
      "value": "admin"
    }
  ]
}

```

Job-Definition Objects

arguments

The arguments key-value pair enables you to specify an argument and its value that is required by the job request. You can specify multiple arguments objects.

TIP If your job request does not require arguments, do not include the arguments object in the template.

Here is an example of how to specify the arguments object:

```

"arguments": {
  "_contextName": "SAS Job Execution Compute context"
}

```

In the example shown in [“Example of a Job-Request Definition” on page 101](#), the parameters object assigns “_contextName” as the value of “name”. Then, the argument object passes a string via the _contextName argument. That string is then set as the name of the parameters object.

properties

The properties object can be a list of name-value pairs that specify anything applicable to the job-definition object that is specified in the job request.

TIP If your job definition does not require properties, do not include the properties object in the template.

Here is an example of how to specify properties:

```
"properties": [  
  {  
    "name": "class",  
    "value": "admin"  
  }  
]
```

- **name** specifies the name of the property expected by the configuration or code.
- **value** specifies the value of the property.

Job-Flow Definition

This topic contains detailed descriptions about the job-flow definition, its key-value pairs, and their settings. For more information, see these topics:

- [“Create a Job Flow and Its Definition” on page 41](#)
- [“Update a Job Flow” on page 43](#)

Job-Flow Definition Template

A template of the job-flow definition is returned by the [“job flows generate-template”](#) command.

Here is the definition template and its descriptions:

```

{
1  "name": "Replace this string with the name of the job flow.",
2  "description": "(Optional) Replace this string with a description of the job flow.",
3  "jobs": [ "Replace this string with a comma-separated-values list of job action IDs, subflow
IDs, or both." ],
4  "triggerType": "Replace this string with one of these values: runNow, event, or manual.",
5  "triggerCondition": "Replace this string with one of these values: any, all.",
6  "schedulerId": "Replace this string with the scheduler ID.",
7  "flowProperties": {
    "completionLevel": "(Optional) Replace this string with one of these values: default,
allSelected, anySelected.",
    "completionSelectedJobs": ["(Optional) Replace this string with a comma-separated-values
list of job actions or subflows."],
    "emailNotification": ["(Optional) Replace this string with a comma-separated-values list
of any or all of these values: start, end, abnormalExit, none."],
    "emailAddress": "(Optional) Replace this string with the email address to send each
notification.",
    "endBehavior": "(Optional) Replace this string with one of these values: stop, continue,
continueDelay.",
    "exitcodeStrategy": "(Optional) Replace this string with one of these values: last, sum."
  },
8  "defaultJobProperties": {
    "jobDefaultEmailNotification": ["(Optional) Replace this string with a comma-separated-
values list of any or all of these values: start, end, abnormalExit, none."],
    "jobDefaultEmailAddress": "(Optional) Replace this string with the email address to send
each notification.",
    "jobDefaultPriority": "(Optional) Replace this string with one of these values: top, high,
normal (the default setting), medium, low."
  }
}

```

- 1 Specifies the name for the flow. See [“Unsupported Characters in Job Action and Job Flow Names” on page 116](#).
- 2 (Optional) Specifies a description about the flow.
- 3 The jobs array. For information about this array, see [“Step 3: Update the Jobs Array in the Definition” on page 42](#).
- 4 Indicates how to run the job flow. See [“triggerType” on page 107](#).
- 5 Indicates how to evaluate conditions for a trigger. Required if `triggerType` is set to `event`. See [“triggerCondition” on page 108](#).
- 6 Specifies the ID of the scheduler.
- 7 (Optional) Specifies settings that are specific to the job flow. See [“flowProperties” on page 108](#).
- 8 (Optional) Specifies your selections for default properties that apply to all job actions and subflows in the job flow. See [“defaultJobProperties” on page 109](#).

IMPORTANT

- (Release 2.6.35 or later of the job CLI plug-in) The output from the `job flows generate-template` command contains the jobs array.
- (Release 2.6.34 or earlier of the job CLI plug-in) The output from the `job flows generate-template` command does not contain the jobs array.

An Existing Job-Flow Definition

An existing job-flow definition is returned by the “[job flows show](#)” command.

Note: When you view an existing definition in the SAS Viya platform, it might contain additional key-value pairs. These do not require input because of one or more of these reasons:

- They are deprecated.
- They are reserved for future use.
- They are assigned a permanent value.

Here is the definition and its descriptions:

```

{
  1 "version": 1,
  2 "id": "f2efcfae-9a12-4cea-8d58-228bce19da90",
  3 "name": "Example Flow",
  4 "links": [
    {
      "method": "GET",
      "rel": "self",
      "href": "/jobFlowScheduling/flows/f2efcfae-9a12-4cea-8d58-228bce19da90",
      "uri": "/jobFlowScheduling/flows/f2efcfae-9a12-4cea-8d58-228bce19da90",
      "type": "application/vnd.sas.schedule.flow"
    },
    {
      "method": "PUT",
      "rel": "update",
      "href": "/jobFlowScheduling/flows/f2efcfae-9a12-4cea-8d58-228bce19da90",
      "uri": "/jobFlowScheduling/flows/f2efcfae-9a12-4cea-8d58-228bce19da90",
      "type": "application/vnd.sas.schedule.flow"
    },
    {
      "method": "DELETE",
      "rel": "delete",
      "href": "/jobFlowScheduling/flows/f2efcfae-9a12-4cea-8d58-228bce19da90",
      "uri": "/jobFlowScheduling/flows/f2efcfae-9a12-4cea-8d58-228bce19da90"
    },
    {
      "method": "GET",
      "rel": "up",
      "href": "/jobFlowScheduling/flows",
      "uri": "/jobFlowScheduling/flows"
    },
    {
      "method": "GET",
      "rel": "scheduler",
      "href": "/jobFlowScheduling/schedulers/57534784-2ba4-49b0-9c1f-c350f29ee436",
      "uri": "/jobFlowScheduling/schedulers/57534784-2ba4-49b0-9c1f-c350f29ee436",
      "type": "application/vnd.sas.schedule.scheduler"
    },
    {
      "method": "PUT",
      "rel": "reschedule",
      "href": "/jobFlowScheduling/flows/f2efcfae-9a12-4cea-8d58-228bce19da90/scheduled",
      "uri": "/jobFlowScheduling/flows/f2efcfae-9a12-4cea-8d58-228bce19da90/scheduled",
      "type": "application/vnd.sas.schedule.flow"
    },
    {
      "method": "PATCH",
      "rel": "execute",
      "href": "/jobFlowScheduling/flows/f2efcfae-9a12-4cea-8d58-228bce19da90/scheduled",
      "uri": "/jobFlowScheduling/flows/f2efcfae-9a12-4cea-8d58-228bce19da90/scheduled",
      "type": "application/vnd.sas.schedule.flow"
    }
  ],
  5 "triggerType": "event",
  6 "triggerCondition": "any",
  7 "jobs": [
    "/jobFlowScheduling/jobs/31f96c40-71e4-4274-9962-948f6664b389"
  ],
  8 "definitionId": "b30c4d9e-184b-430f-9f25-366b0b7d8ddd",
  9 "submitted": true,
  10 "schedulerId": "57534784-2ba4-49b0-9c1f-c350f29ee436",
  11 "scheduledBy": "trmisu",
  12 "flowProperties": {
    "completionLevel": "default",
    "endBehavior": "stop",
    "exitcodeStrategy": "last"
  },
  13 "defaultJobProperties": {}
}

```

- 1 (Optional) The version of the job flow scheduling API. The current version is 1 and is the default. Specifying this key-value pair is not required.
- 2 The ID of the job flow.
- 3 The name specified for the job flow when it was created. You can update the name when you edit this file. See [“Unsupported Characters in Job Action and Job Flow Names” on page 116](#).
- 4 This is a list of links that are associated with the job flow. For example, the “scheduler” link gets the scheduler that is associated with the job flow.
- 5 Indicates how to run the job flow. See [“triggerType” on page 107](#).
- 6 Indicates how to evaluate conditions for a trigger. Required if `triggerType` is set to `event`. See [“triggerCondition” on page 108](#).
- 7 The jobs array. For information about this array, see [“Step 3: Update the Jobs Array in the Definition” on page 42](#).
- 8 The ID of the job-flow definition placed on the scheduler associated with the job flow. The placement occurs when you schedule or reschedule a job flow. If the job flow is not scheduled, this field is empty and omitted from the returned JSON.
- 9 Indicates whether a job flow is scheduled. `true` indicates the job flow is scheduled.
- 10 The ID of the scheduler.
- 11 The SAS user who scheduled the job flow.
- 12 (Optional) Specifies settings that are specific to the job flow. See [“flowProperties” on page 108](#).
- 13 (Optional) Specifies your selections for default properties that apply to all job actions and subflows in the job flow. See [“defaultJobProperties” on page 109](#).

Job-Flow Definition Settings

These settings are used in the job-flow definition. See [“Job-Flow Definition” on page 103](#).

triggerType

The `triggerType` key-value pair indicates how to run the job flow.

<code>runNow</code>	Runs the job flow immediately and only one time.
<code>event</code>	Runs the job flow whenever a specified event occurs. The event is a time-based trigger.
<code>manual</code>	Indicates that the job flow is scheduled, but the flow must be triggered manually and not by the scheduled trigger.

Note: For information about the `manual` setting, see [“Trigger a Job Flow with a Manual Trigger” on page 44](#).

triggerCondition

The `triggerCondition` key-value pair indicates how to evaluate conditions for any associated triggers. This setting is required when `triggerType` is set to `event`.

- `any` Any of the specified conditions can be true to trigger the flow (logical OR).
- `all` All of the specified conditions must be true to trigger the flow (logical AND).

flowProperties

The `flowProperties` key-value pair specifies property settings for the job flow. You can adjust these settings.

If you create a job flow without setting these properties, the default settings are used.

```
"flowProperties": {},
```

Use this structure to specify the properties.

```
"flowProperties": {
  "completionLevel": "default",
  "endBehavior": "stop",
  "exitcodeStrategy": "last"
},
```

- **completionLevel:** Indicates how to determine when a job flow is complete.

- `none` Indicates that no completion level is specified for the flow.
- `default` Uses the SAS Job Flow Scheduler default method for determining whether a job flow is completed. This is the default setting.
- `allSelected` Determines the flow is complete when all of the specified jobs are no longer running. These jobs are specified in the `completionSelectedJobs` list.
- `anySelected` Determines the flow is complete when any of the specified jobs are no longer running. These jobs are specified in the `completionSelectedJobs` list.

Note: For a description of the `completionSelectedJobs` list, see the next item in this list.

The default method that the SAS Job Flow Scheduler uses to determine whether a job flow is completed is based on these evaluations:

- 1 All job actions and subflows in the flow have completed successfully. The flow is marked as completed.
- 2 Any job action or subflow in the flow fails and there is no direct job-event dependency on that failed item. The flow is stopped and marked as exited. However, if a job action or subflow fails and there is a direct job-event dependency on that failed item, the flow is not stopped. Instead, the job action or subflow that depends on the job event is executed.
- 3 Any job action or subflow in the flow is killed. The flow is stopped and marked as killed.

- **completionSelectedJobs:** Specifies a list of jobs to evaluate when the `completionLevel` is either `allSelected` or `anySelected`. The list contains job actions and subflows that you select from the `jobs` array. However, they are identified by their names instead of their URI.

- **emailNotification:** Indicates when to send an email for each job action or subflow in the job flow. You can specify one or more values. The values must be separated by commas. If you specify `emailNotification`, you must set `emailAddress`.

<code>none</code>	Indicates not to send an email. This is the default setting.
<code>start</code>	Sends an email when the job action starts.
<code>end</code>	Sends an email when the job action ends due to a situation other than success or failure.
<code>abnormalExit</code>	Sends an email when the job action ends in failure.

- **emailAddress:** Specifies the email address to send the notification. If you specify `emailAddress`, you must set `emailNotification`.

- **endBehavior:** When the specified `completionLevel` setting is met, this setting specifies how the job flow ends any job actions or subflows that are still running.

<code>stop</code>	When the <code>completionLevel</code> setting is met, set the state to indicate how the job flow ended and stop executing any more items in the job flow. This is the default setting.
<code>continue</code>	When the <code>completionLevel</code> setting is met, set the state to indicate how the job flow ended and execute as much of the remaining items in the job flow as possible.
<code>continueDelay</code>	When the <code>completionLevel</code> setting is met, delay setting the state until as many remaining items are run in the job flow as possible.

- **Exclusive:** When set to `true`, indicates that only one instance of the job flow can execute at any given time. The default setting is `false`.

- **exitcodeStrategy:** Indicates how to determine the exit code for the job flow.

<code>last</code>	Use the exit code of the last item executed in the job flow as the exit code for the entire job flow. This is the default setting.
<code>sum</code>	Use the sum of the exit codes of all items in the job flow as the exit code for the entire job flow.

defaultJobProperties

These properties enable you to specify email notification, priority, and more. Each setting applies to all of the job actions and subflows in the job flow unless a job action or subflow explicitly sets its own property as an override.

If you create a job flow without setting any of these properties, the default settings are used.

```
"defaultJobProperties": {}
```

Update the structure to specify settings.

```
"defaultJobProperties": {
  "emailNotification": "end",
  "emailAddress": "trmisu@orion.com"
}
```

- **emailNotification:** Indicates when to send an email for each job action or subflow in the job flow. You can specify one or more values. The values must be separated by commas. If you specify emailNotification, you must set emailAddress.

none	Indicates not to send an email. This is the default setting.
start	Sends an email when the job action starts.
end	Sends an email when the job action ends due to a situation other than success or failure.
abnormalExit	Sends an email when the job action ends in failure.
- **emailAddress:** Specifies the email to send the notification. If you specify emailAddress, you must set emailNotification.
- **priority:** Specifies the level of priority for the items in the flow.
 - top
 - high
 - normal. This setting is the default.
 - medium
 - low
- **environmentVariables:** This setting is reserved.

Job-Scheduler Definition

This topic contains detailed descriptions about the job-scheduler definition, its key-value pairs, and their settings.

.....

Note: Typically, you do not create a scheduler. The Job-Flow-Scheduling Service creates a scheduler as part of the service's start-up process.

.....

Job-Scheduler Definition Template

A template of the job-scheduler definition is returned by the [“job schedulers generate-template”](#) command.

.....

Note: When you view an existing definition in the SAS Viya platform, it might contain additional key-value pairs. These do not require input because of one or more of these reasons:

- They are deprecated.
 - They are reserved for future use.
 - They are assigned a permanent value.
-

Here is the definition template and its descriptions:

```
{
  1 "name": "Replace this string with the name of the scheduler.",
  2 "description": "(Optional) Replace this string with a description of the scheduler.",
  3 "type": "sasjfs",
  4 "SASJobFlowSchedulerConfiguration": {
    "context": "Replace this string with the context for the SAS Launcher Service."
  }
}
```

- 1 Specifies the name for the scheduler.
- 2 (Optional) Specifies a description of the scheduler.
- 3 Indicates the required type of the scheduler. Do not change this value.
- 4 Specifies the context for the SAS Launcher Service. It is required by the job-flow-scheduling service.

Time-Based Trigger Definition

Considerations

- For details about adding a trigger, see [“Add a Trigger” on page 45](#).
- When creating or updating any JSON structure, use a JSON validation utility to ensure that the structure is valid.

Syntax

Use the following JSON syntax to define one or more time-based triggers to schedule a job flow or a job request.

```
[{
  "type": "timeevent",
  "name": "trigger-name",
  "active": true,
  "event": {
    "recurrence": {
      "type": "recurrence-type",
      recurrence-options
    }
    "hours": "hours",
    "minutes": "minutes",
    "duration": duration,
    "timeZone": "zone",
    "maxOccurrence": occurrences
  }
}]
```

where:

- *trigger-name* is the name that you specify for the trigger.
- *recurrence_type* is the interval of the time trigger. Types include `minutely`, `hourly`, `daily`, `weekly`, `monthly`, `yearly`, and `dateList`.
For examples of how to use each recurrence type, see [“Define a Time-Based Trigger” on page 46](#).
- *recurrence-options* are the applicable options (such as `startDate` and `endDate`) for the selected recurrence type. See [“Recurrence Types and Their Options” on page 112](#).
- *hours*, *minutes*, *duration*, *zone*, and *occurrences* can be used to specify aspects of the time of day when each job occurs (or the time of the first job occurrence if they run every minute). See [“Time-of-Day Options” on page 115](#).

Recurrence Types and Their Options

Minutely Recurrence

Use the following options to specify recurrence intervals when using `"type": "minutely"`:

- `startDate` specifies when to start the recurrence in the form `yyyy-mm-dd`.
- `endDate` specifies when to stop the recurrence in the form `yyyy-mm-dd`.
- `skipCount` specifies how many minutes pass between executions. For example, `"skipCount": 15` specifies that the job runs every 15 minutes.
 - Valid values are 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, and 30.
 - The default is 1.

Note: See [“minutes” on page 115](#) for an optional interaction with `skipCount`.

Hourly Recurrence

Use the following options to specify recurrence intervals when using `"type": "hourly"`:

- `startDate` specifies when to start the recurrence in the form `yyyy-mm-dd`.
- `endDate` specifies when to stop the recurrence in the form `yyyy-mm-dd`.
- `skipCount` specifies how many hours pass between executions. For example, `"skipCount": 3` specifies that the job runs every 3 hours.
 - Valid values are 1, 2, 3, 4, 6, 8, and 12.
 - The default is 1.

Note: See [“hours” on page 115](#) for an optional interaction with `skipCount`.

Daily Recurrence

Use the following options to specify recurrence intervals when using `"type": "daily"`:

- `startDate` specifies when to start the recurrence in the form `yyyy-mm-dd`.
- `endDate` specifies when to stop the recurrence in the form `yyyy-mm-dd`.
- `skipCount` specifies how many days pass between executions. For example, a value of 3 specifies that the job runs every 3 days.
 - `skipCount` and `DaysOfWeek` are mutually exclusive.
 - The default is 1.
- `daysOfWeek` specifies the days on which to run the job. For example, `"daysOfWeek": ["thursday"]` specifies that the job runs every Thursday.
 - Valid values are `monday-sunday`.
 - `skipCount` and `DaysOfWeek` are mutually exclusive.
 - This setting is optional.

Weekly Recurrence

Use the following options to specify recurrence intervals when using `"type": "weekly"`:

- `startDate` specifies when to start the recurrence in the form `yyyy-mm-dd`.
- `endDate` specifies when to stop the recurrence in the form `yyyy-mm-dd`.
- `skipCount` specifies how many weeks pass between executions. For example, `"skipCount": 3` specifies that the job runs every 3 weeks. The default is 1.
- `daysOfWeek` specifies the days on which to run the job. For example, `"daysOfWeek": ["thursday"]` specifies that the job runs every Thursday.
 - Valid values are `monday-sunday`.
 - This setting is required.

Monthly Recurrence

Use the following options to specify recurrence intervals when using `"type": "monthly"`:

- `startDate` specifies when to start the recurrence in the form `yyyy-mm-dd`.
- `endDate` specifies when to stop the recurrence in the form `yyyy-mm-dd`.
- `skipCount` specifies how many months pass between executions. For example, `"skipCount": 3` specifies that the job runs every 3 months. The default is 1.
- `occurrence` is used with `daysOfWeek` to specify the days on which to run the job. Valid values are `first`, `second`, `third`, `fourth`, and `last`.
- `daysOfWeek` specifies the days on which to run the job. For example, `"daysOfWeek": ["thursday"]` specifies that the job runs every Thursday.
 - Valid values are `monday-sunday`.
 - If `daysOfWeek` is specified, then `occurrence` is required.
 - `daysOfWeek` and `dayOfMonth` are mutually exclusive.
 - If both `daysOfWeek` and `dayOfMonth` are specified, then both are ignored.
- `dayOfMonth` specifies the day of the month on which to run the job.

- ❑ Valid values are 1–31.
- ❑ A value of 32 specifies the last day of the month.
- ❑ `dayOfMonth` and `daysOfWeek` are mutually exclusive.
- ❑ If both `daysOfWeek` and `dayOfMonth` are specified, then both are ignored.

Yearly Recurrence

Use the following options to specify recurrence intervals when using `"type": "yearly"`:

- `startDate` specifies when to start the recurrence in the form `yyyy-mm-dd`.
- `endDate` specifies when to stop the recurrence in the form `yyyy-mm-dd`.
- `skipCount` specifies how many years pass between executions. The default is 1.
- `occurrence` is used with `daysOfWeek` to specify the days on which to run the job. Valid values are `first`, `second`, `third`, `fourth`, and `last`.
- `daysOfWeek` specifies the days on which to run the job. For example, `"daysOfWeek": ["thursday"]` specifies that the job runs every Thursday.
 - ❑ Valid values are `monday`–`sunday`.
 - ❑ If `daysOfWeek` is specified, then `occurrence` is required.
 - ❑ `daysOfWeek` and `dayOfMonth` are mutually exclusive.
 - ❑ If both `daysOfWeek` and `dayOfMonth` are specified, then both are ignored.
- `dayOfMonth` specifies the day of the month on which to run the job.
 - ❑ Valid values are 1–31.
 - ❑ A value of 32 specifies the last day of the month.
 - ❑ `dayOfMonth` and `daysOfWeek` are mutually exclusive.
 - ❑ If both `daysOfWeek` and `dayOfMonth` are specified, then both are ignored.
- `monthOfYear` specifies the month in which to run the job. Valid values are `january`–`december`.

Date List Recurrence

Use the following options to specify recurrence intervals when using `"type": "dateList"`:

- `startDate` specifies when to start the recurrence in the form `yyyy-mm-dd`.
- `endDate` specifies when to stop the recurrence in the form `yyyy-mm-dd`.
- `dates` specifies an array of dates in the form `yyyy-mm-dd`. The job is run on the specified dates.

TIP `dateList` does not use `skipCount`. If you include `skipCount`, it is ignored.

Time-of-Day Options

Use the following options to specify aspects of the time of day when each job occurs (or the time of the first job occurrence if they run every minute).

minutes

For recurrence types other than `minutely`

`minutes` specifies a set of minutes when to run the job. Remember these key points to specify a valid value:

- Valid values for `minutes` are 0–59.
- You can specify a list of minutes in the following ways:
 - separated by commas (0,10,30)
 - a range of minutes (20–25)
 - a combination of a range and a list (0,10–15)
 - specify every minute with an asterisk (*)

For the `minutely` recurrence type

`minutes` specifies the offset from the top of the hour. That offset is used to start the time event.

For example, if `skipCount` is set to 15, the time event runs four times an hour. A valid offset is 0–14. If `minutes` is set to 10, the offset is 10. The time event runs at 10, 25, 40, and 55 minutes past the hour.

To specify a valid value for `minutes` with `minutely` recurrence type, remember these key points:

- If a set of minutes is specified, only the first value in the set is used.
- The value must be less than the value of `skipCount`.
- If the value is not valid, then 0 is the default.

hours

For recurrence types other than `hourly` and `minutely`

`hours` specifies a set of hours when to run the job. Remember these key points to specify a valid value:

- Valid values for `hours` are 0–23.
- You can specify a list of hours in the following ways:
 - separated by commas (1,2,3)
 - a range of hours (2–4)
 - a combination of a range and a list (1–3,5,7)
 - specify all hours with an asterisk (*)

For the `hourly` recurrence type

`hours` specifies the offset from the start of the day. That offset is used to start the time event.

For example, if `skipCount` is set to 6, the time event runs four times a day. A valid offset is 0–5. If `hours` is set to 4, the offset is 4. The time event runs at 4, 10, 16, and 22 hours in the day.

To specify a valid value for `hours` with the `hourly` recurrence type, remember these key points:

- If a set of hours is specified, only the first value in the set is used.
- The value must be less than the value of `skipCount`.
- If the value is not valid, then 0 is the default.

For the `minutely` recurrence type
the `hours` option is ignored.

duration

Specifies the number of minutes that the event remains true. The default is 1 minute.

This option is useful when evaluating multiple conditions. For example, you have two events that are evaluated together using the condition AND. The first is a time event and the second is a file event. The time event is set daily at midnight. However, at midnight the file event is not true. Therefore, the AND condition is not true.

By using the `duration` option, you can set the time event to have a duration of 15 minutes. Now if the file event is true anytime between midnight and 12:15 am, the AND condition is true.

timeZone

Specifies the time zone to use when evaluating the time event.

- Specify the value using the Olson time zone database, in the form `region/city`. An example is `America/New_York`. For more information, see the [Time Zone Database](#).
- The default is the time zone of the system executing the job.

maxOccurrence

Specifies the maximum number of times the job can execute. This option is ignored by default.

Unsupported Characters in Job Action and Job Flow Names

When you specify a name for a job action or job flow definition, you enter a value for the `name` key-value pair. However, you must not use these unsupported characters:

- & (ampersand)
- ' (apostrophe)
- * (asterisk)
- @ (at sign)
- \ (backslash)
- ` (backtick)
- ^ (caret)

- : (colon)
- , (comma)
- \$ (dollar sign)
- = (equal sign)
- ! (exclamation mark)
- > (greater-than sign)
- < (less-than sign)
- ((parenthesis, opening)
-) (parenthesis, closing)
- % (percent sign)
- . (period)
- | (pipe)
- + (plus sign)
- ? (question mark)
- " (quotation mark)
- ; (semi-colon)
- / (slash)
- [(square bracket, opening)
-] (square bracket, closing)
- ~ (tilde)

Using a Group-Managed Service Account

Group-Managed Service Account: About

What Is a Group-Managed Service Account

A group-managed service account enables members of a scheduling group to execute or schedule a job or job flow that was created by another member of that group. When executed or scheduled, the job or job flow is initiated by the service account.

IMPORTANT The group-managed service account function is available by applying the latest hot fixes. See [“Managing Your Software” in SAS Viya for Linux: Deployment Guide](#) and

“Managing Your Software” in *SAS Viya for Windows: Deployment Guide* for information about how to apply hot fixes to your SAS Viya deployment.

Note:

After the hot fixes are applied, the group-managed service account feature must be configured. Until the configuration is complete, one of these error messages appears when attempting to use the feature:

- No service account available.
 - The Run As functionality requires a group-managed service account. None are available. Contact your SAS administrator.
-

Why Use a Group-Managed Service Account?

Uninterrupted Support

Group-managed service accounts enable a group of users to share the responsibility for executing, scheduling, and monitoring jobs and job flows. This is useful in production environments where scheduled jobs and job flows must continue to execute successfully. The dependency on any single user responsible for those jobs and job flows is eliminated.

Note: A group-managed service account does not have any special privileges or unique features other than it enables a group of users to share responsibility for jobs and job flows.

By using non-personal accounts (a group-managed service account), you have the following advantages:

- A group of users are enabled to share the responsibility for scheduling, executing, and monitoring jobs and job flows. The dependency on one person is eliminated.
- The requirement to assign SAS administrator privileges for the sole purpose of enterprise scheduling is eliminated.

Improved Security

As of the installation of a set of hot fixes, the functionality and security of the Run As feature is improved with the implementation of the group-managed service account. The Run As feature is available in the **Scheduling** tab of the Jobs and Flows page in SAS Environment Manager. The improved functionality enables the responsibility of scheduling jobs and job flows to be shared by a group of individuals.

Prior to the introduction of the group-managed service account, the Run As functionality enabled SAS administrators to schedule jobs and flows using the credentials of another SAS user.

However, in some organizations, the Run As functionality presented the following concerns:

- SAS users who are not SAS administrators need to be responsible for scheduling jobs.

- The ability to schedule jobs and flows using another SAS user's credentials is considered a security flaw.

To improve security and enable SAS users other than SAS administrators to schedule jobs and flows, the Run As functionality has been redesigned as the group-managed service account. Now, SAS users other than SAS administrators can schedule jobs and flows to run. By associating non-administrators with a service account via a domain, they can manage and schedule jobs and flows with improved security.

Expanded Support

Compared to the prior Run As functionality, the group-managed service account provides the following expanded support:

- It supports both job requests and job flows, not just job requests.
- It can have group members who are not SAS administrators.

How a Group-Managed Service Account Works

A group-managed service account enables a scheduled job or job flow to be executed and managed by an entity other than the user. The actual execution is by a service account. To accomplish this goal securely, SAS Viya uses the following functionalities:

- A group-managed service account. The service account is assigned the permissions and authorizations necessary to execute or schedule any job or job flow of interest to all members of a scheduling group. This includes all of the permissions and authorizations required to use any of the resources used by the scheduled jobs and job flows, such as folders, files, databases, and so on.
- A scheduling group whose members share responsibility for executing, scheduling, and monitoring jobs and job flows.
- A token authentication domain that enables the following functions:
 - the association between the service account and the scheduling group
 - authorization rules that enable the group-managed service account to function as required.

Scheduled job flows and jobs that use a group-managed service account are executed by the service account using token authentication.

When the job flow or job is executed, members of the scheduling group associated with the service account can monitor and view the results of the executed job flow or job.

How to Transition the Run As Function

When a SAS Viya installation is upgraded or the applicable hot fixes are applied, the prior Run As functionality is removed and the group-managed service account functionality becomes available. The **Run As** option remains in SAS Environment Manager, but it now provides the group-managed service account functionality.

IMPORTANT If your organization uses the prior Run As functionality, you must update any existing scheduled job requests that use the functionality. Job flows could not use the prior Run As functionality, but can use the group-managed service account functionality.

Perform the following tasks to transition to using a group-managed service account:

- 1 Identify any scheduled job requests that use the prior Run As functionality. See [“Identify a Job Request That Uses Run As” on page 124](#).

IMPORTANT If your SAS Viya platform environment does not use the prior Run As functionality with any scheduled job requests, then you must perform step 2 only.

- 2 Configure the prerequisites. See [“Group-Managed Service Account: Prerequisites” on page 120](#).
- 3 Update the job requests identified in step 1 to use the group-managed service account instead of an individual’s account. See [“Update a Job Request to Run Using a Group-Managed Service Account” on page 122](#).

Note: If you do not want to update the scheduled job requests that use the prior Run As functionality, you have these choices:

- Ask an administrator or the creator of each affected job request to share the job request ID with the members of the group-managed service account.

Any user can execute a job request and view its history as long as the user knows the job request ID. However, the job request ID is visible only to administrators and the creator of the job request. The administrator or creator can opt to share the job request ID with the members of the group-managed service account. However, other users cannot view the job request instances.

- Ask the creator of each affected job request to share the job request definition with the members of the group-managed service account.

By using the job request definition, the job request can be re-created by the members of the group-managed service account.

Group-Managed Service Account: Prerequisites

Note: The following functional areas require the latest hot fixes to provide the added functionality.

The group-managed service account uses the following functions in SAS Viya:

- 1 To enable the use of a group-managed service account, configure an authentication domain that uses a token.
 - For information about token domains, see [“Authentication Domain” in SAS Viya Administration: External Credentials](#).

- To configure the group-managed service account to work with a token domain, see [“Allow a Group-Managed Service Account to Manage the Credentials for a Token Domain” in SAS Viya Administration: External Credentials](#).
- 2 The group-managed service account must be assigned all of the permissions and authorizations required to use any of the resources that the scheduled jobs and job flows use. Resources can include folders, files, databases, and so on.

Group-Managed Service Account: How To

Use SAS Environment Manager with a Group-Managed Service Account

Members of the scheduling group can perform the following tasks with jobs and job flows submitted via a group-managed service account.

Jobs:

- [“View a Table of Job Executions” on page 14](#)
- [“Filter Details about Jobs” on page 16](#)
- [“Rerun a Job from the Monitoring Tab” on page 22](#)
- [“Delete a Job from the Monitoring Tab” on page 24](#)
- [“Viewing the Job Log” on page 17](#)
- [“View a Chart of Scheduled Jobs” on page 16](#)
- [“Schedule a Job” on page 18](#)
- [“View the Execution History for a Job” on page 15](#)
- [“View Job Properties” on page 23](#)

Job flows:

- [“Schedule a Job Flow” on page 32](#)
- [“Disabling or Uncheduling the Schedule for a Job Flow” on page 36](#)
- [“Execute a Job Flow Immediately” on page 35](#)
- [“View the Execution History for a Job Flow and Status of Its Objects” on page 25](#)
- [“Modify Job Flow Properties and Attributes” on page 31](#)

Use the Job Command-Line Interface with a Group-Managed Service Account

The job plug-in to the sas-admin CLI supports using a group-managed service account. The following commands provide support.

Job request commands:

- [Execute a job request on page 71](#)
- [Schedule a job request on page 72](#)
- [Unschedule a job request on page 73](#)
- [List the history for a job request on page 74](#)
- [List the details about the history for a job request on page 75](#)
- [Show the status for a job request instance on page 76](#)

Job flow commands:

- [Update a job flow on page 81](#)
- [Execute a job flow on page 83](#)
- [Schedule a job flow on page 84](#)
- [Reschedule a job flow on page 84](#)
- [Unschedule a job flow on page 85](#)
- [List the history for a job flow on page 85](#)
- [List the details about the history for a job flow on page 87](#)
- [Show the status for a job flow instance on page 88](#)

Note:

The following commands support the group-managed service account function as of release 2.6.26 of the job plug-in to the sas-admin CLI:

- List the history for a job request
 - List the details about the history for a job request
 - Show the status for a job request instance
 - List the history for a job flow
 - List the details about the history for a job flow
 - Show the status for a job flow instance
-

Update a Job Request to Run Using a Group-Managed Service Account

About This Procedure

When a SAS Viya installation is upgraded or the applicable hot fixes are applied, the prior Run As functionality is removed and the group-managed service account functionality becomes available. The **Run As** option remains in SAS Environment Manager, but it now provides the group-managed service account functionality.

IMPORTANT

If your organization uses the prior Run As functionality, you must update any existing scheduled job requests that use the functionality. See [“How to Transition the Run As Function” on page 119](#).

If your organization does not use the prior Run As functionality, you can ignore this procedure.

When the group-managed service account feature is available, job requests can use the group-managed service account instead of an individual's account.

Scenario 1

This scenario uses the following criteria:

- User 1 is a SAS administrator who scheduled the job request using the Run As functionality.
- User 1 is a member of the group that is associated with the group-managed service account.

User 1 completes all of these steps:

- 1 Unschedule the job request. Use either of these methods:
 - SAS Environment Manager. See [“Unschedule a Job” on page 24](#).
 - The `sas-admin job requests unschedule` command. See [Unschedule a job request on page 73](#).
- 2 Schedule the job request again and specify the group-managed service account. Use either of these methods:
 - SAS Environment Manager. See [“Schedule a Job” on page 18](#).
 - The `sas-admin job requests schedule` command. See [Schedule a job request on page 72](#). The value for the `--domain-id` option must be specified.

Scenario 2

This scenario uses the following criteria:

- The job request was scheduled using the Run As functionality by User X who is no longer available.
- User 2 is a SAS administrator.
- User 2 is not a member of the group that is associated with the group-managed service account.

User 2 completes all of these steps:

- 1 Unschedule the job request. Use either of these methods:
 - SAS Environment Manager. See [“Unschedule a Job” on page 24](#).
 - The `sas-admin job requests unschedule` command. See [Unschedule a job request on page 73](#).
- 2 Add User 2's account to the group-managed service account. See [“Add or Remove Custom Group Members” in SAS Viya Administration: Identity Management](#).

- 3 Schedule the job request again and specify the group-managed service account. Use either of these methods:
 - SAS Environment Manager. See [“Schedule a Job” on page 18](#).
 - The `sas-admin job requests schedule` command. See [Schedule a job request on page 72](#). The value for the `--domain-id` option must be specified.
- 4 Remove User 2's account from the group-managed service account. See [“Add or Remove Custom Group Members” in SAS Viya Administration: Identity Management](#).

Identify a Job Request That Uses Run As

If your organization uses the prior Run As functionality and you want to change the job request to use the group-managed service account, you must identify the job requests that use the prior Run As functionality.

Note: The prior Run As functionality is available in an installation without the applicable hot fixes.

Using SAS Environment Manager

Note: You must be a SAS administrator to perform this task in SAS Environment Manager.

Complete these steps for each job request that is scheduled:

- 1 Note the user who created the scheduled job request. See [“View Scheduled Jobs” on page 15](#).

TIP The ID of the user that created the job is not shown by default. You might need to add this column to your view.

- 2 For each scheduled job request, open the Edit Schedule window to view the value in the **Run As** field. See [“Edit a Scheduled Job” on page 23](#).

If the value is a user who is not the creator of the job request and the value is not a service account, then the job request is still using the prior Run As functionality.

Using the Job Command-Line Interface

Note: You must have SAS administrator privileges to receive output for all job requests.

Complete these steps for each job request that is scheduled:

- 1 Identify the scheduled job requests (scheduled-job-request ID) that have an assigned Run As value:

```
sas-viya job requests list --scheduled
```

- 2 From this list, eliminate any scheduled job requests where the Run As value is a service account. Save the list of remaining job requests.
- 3 List all job requests:

```
sas-viya job requests list
```
- 4 From the output, eliminate the job requests that are not scheduled. Save this list.
- 5 From this list, identify the creator value of each job request:

```
sas-viya job requests show --id job_request_id
```

Where *job_request_id* is the ID of a job request.
- 6 Compare the Run As value with the creator value. If the values do not match, then the job request is still using the prior Run As functionality.

Jobs and Flows: Troubleshooting

After Adjusting for DST, Schedules Are Still Off by One Hour

Problem:

After you adjust for Daylight Saving Time (DST), scheduled job requests and job flows are still off by one hour.

Explanation 1:

When you adjusted for DST, you specified a default time zone by setting the `timeZone` configuration property in SAS Environment Manager. Scheduled job requests and job flows can still be off by one hour if the following criteria are met:

- The specified time zone has not adopted Daylight Saving Time.
- The Jobs and Flows Scheduler in SAS Environment Manager or the jobs CLI is used in a time zone that has adopted Daylight Saving Time.

Explanation 2:

If you did not specify a default time zone by setting the `timeZone` configuration property in SAS Environment Manager, the operating-system setting for the time-zone value on the system that hosts the SAS Viya platform is used. Scheduled job requests and job flows can still be off by one hour if the following criteria are met:

- the operating-system time zone has not adopted Daylight Saving Time.
- The Jobs and Flows Scheduler in SAS Environment Manager or the jobs CLI is used in a time zone that has adopted Daylight Saving Time.

Solution:

Specify the default time zone by setting the `timeZone` configuration property in SAS Environment Manager. For details, see [“Adjust for Daylight Saving Time” on page 57](#).

IMPORTANT

- Verify that the selected time zone has adopted Daylight Saving Time. Coordinated Universal Time (UTC) is not a valid selection.
- It is recommended that you specify the time zone by using the TZ identifier (for example, `America/New_York`) instead of the TZ abbreviation (EST/EDT). For more information, see [Time Zone Database](#).

Command-Line Command Fails in a Job Flow

Problem:

A command-line command fails in a job flow.

Explanation:

A command-line command can fail if the command is not permitted by the SAS Launcher Service allow list. If the command is not in the allow list, SAS Launcher Service rejects the call.

Complete these steps to determine whether the command is in the allow list:

- 1 Retrieve the launcher context that is used by the scheduler. See [“Retrieve the Launcher Context Name” on page 59](#).
- 2 In SAS Environment Manager, select the **Context** page.
- 3 In the **View** list, select **Launcher contexts**.
- 4 Under the **View** list, select the launcher context that you retrieved in step 1.
- 5 In the right pane, expand **Commands**.

The commands that are allowed by the allow list are displayed.

Solution:

Add the command to the SAS Launcher Service allow list. You can add the command on the **Context** page in SAS Environment Manager. For more information about the allow list feature, see [“Security Considerations” in SAS Viya Administration: Programming Run-Time Servers](#).

Note: SAS Launcher Service does not validate whether the command is a valid command to run in the pod. It is the responsibility of the creator of the job action that contains the command to ensure that the command is valid to run.

Determine If the Job CLI Plug-In Is Installed

The jobs and flows command-line interface (CLI) is available via the job plug-in to the sas-admin CLI.

Note: If the CLI is not installed, see [“Command-Line Interface: Preliminary Instructions”](#) in *SAS Viya Administration: Using the Command-Line Interfaces*.

After you sign in to the sas-admin CLI, enter the following command:

```
sas-admin plugins list
```

If job is not listed in the output, the plug-in is not installed. To install the plug-in, see [“Download the CLI Executable”](#) in *SAS Viya Administration: Using the Command-Line Interfaces*.

The Filter Option on CLI Commands Fails

Problem:

In the job plug-in to the CLI, the `--filter` option is available on some commands. That option can fail due to unexpected syntax errors in the filter expression. These are examples of possible error messages:

- The request could not be understood by the server due to malformed syntax.
 - The requested resource could not be found.
 - Unexpected token in expression or statement.
-

Note: Messages might vary depending on the operating system on which the CLI is running.

Resolution:

For information about how to correct the filter expression, see [“Known Limitations”](#) on page 61.

No Service Account Available

Problem:

The Run As function is available in SAS Environment Manager on the **Scheduling** tab of the Jobs and Flows page. When selecting **Run As**, the following error message is displayed: No service account available.

Note: Updated releases of SAS Viya 3.5 display this error message: The Run As functionality requires a group-managed service account. None are available. Contact your SAS administrator.

Explanation:

The Run As function has been redesigned. It is now called the group-managed service account. When a SAS Viya installation is upgraded or the applicable hot fixes are applied, the group-managed service account becomes available. The error message is displayed because the function now requires a service account.

Resolution:

Configure a group-managed service account. For more information, see [“Group-Managed Service Account: Prerequisites”](#) on page 120.

Error: The job action or job flow name is not valid.

Problem:

You receive this error when using either of the following commands:

- `job actions create`
- `job flows create`

Explanation:

When you use one of these commands, you specify the associated definition. Each definition includes a key-value pair called `name`. There are unsupported characters that you cannot use in `name`. Any unsupported characters cause the error.

Resolution:

Remove any unsupported characters from the name. See [“Unsupported Characters in Job Action and Job Flow Names”](#) on page 116.