SAS® Viya® 3.4 Administration: Jobs

Jobs: Overview . . . 1

Jobs: How To . . . 2
- Monitor Jobs . . . 2
- Schedule Jobs . . . 3

Scheduling Command Line Interface . . . 6
- Scheduling: How to (Command Line Interface) . . . 6
- Scheduling: Command-Line Interface Reference . . . 7

Jobs: Overview

The Jobs page enables you to monitor and schedule jobs from a variety of sources in SAS Viya.

The **Monitoring** tab enables you to view a table or a chart of jobs that are currently running and that have run in a specified time in the past. You can filter the jobs to narrow the number of jobs displayed and change the time period for displaying jobs. You can also rerun jobs and delete jobs from the list.

The **Scheduling** tab enables you to schedule jobs to run at a particular time or in response to a specific trigger. You can run a job immediately, or you can specify a time interval (from daily to yearly) to control when the job runs. You can also unschedule, delete, and view the properties of jobs.

Jobs that are available for scheduling are from these sources:

SAS Data Explorer
: Creates jobs that you can schedule using SAS Environment Manager.

SAS Data Studio
: Creates jobs that you can schedule using SAS Environment Manager.

SAS Visual Analytics
: Creates jobs that are scheduled in SAS Visual Analytics. You can view and modify the schedules in SAS Environment Manager.

CAS table state management
: 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

You can schedule these jobs, but you cannot delete them, and 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*.

To access the Jobs page, click **Jobs** in the SAS Environment Manager navigation menu.

Jobs: How To

Monitor Jobs

View a Table of Job Executions

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 job name, the start and end date and time, the run time, the job status, and the user that submitted the job. If the job has completed, the table also includes a link to download the log for the job, if one was created. You can also list the environment in which the job ran, although this column 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: 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.

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

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.

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.

To filter by job status, select one or more check boxes in the **Status** list that you want to display.

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. You can filter by creator only if you opted in to the SAS Administrators group when you signed in to SAS Environment Manager.

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.

To remove a filter, deselect its check box and click **Apply**. To remove all filters in either the **Status** or **Created By** list, click **Reset** next to the list. To remove all filters, click **Reset all**.

Rerun a Job from the Monitoring Tab

To rerun a job, right-click the job in the **Jobs** table and select **Run Now** from the pop-up menu. A copy of the job is created and is displayed in the list.

Delete a Job from the Monitoring Tab

Details about jobs remain in the list on the **Monitoring** tab unless you delete the entry. To delete the entry for a job execution, right-click the entry in the **Jobs** table and select **Delete** from the pop-up menu.

View 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. Click **Download** in the **Log** column to save or open a local copy of the log file. The specific behavior depends on your browser.

Schedule Jobs

Schedule a Job

1. On the **Scheduling** tab, click ▦ to display the **Jobs** table. The table is displayed by default when you open the **Scheduling** tab. By default, the table displays the job name, scheduled status, description, and the date on which the job was created. You can also choose to display the ID of the user that created the job, 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, and the job type. These columns are 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: Report distribution jobs 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.

2. Select a job in the **Jobs** table.
3. Click ⏲ in the toolbar or select **Schedule** from the pop-up menu.
4. (Optional) To run the job under credentials other than your own, in the Schedule Job window, specify the user ID under which the job should be run in the **Run as** field. Click to select from specified identities. The user that you select must have previously signed in to SAS since it was installed.
5. Activate the **Enabled** control for one or more triggers in the **Available triggers** table. A trigger controls when the job runs. See "Create a Time Trigger" on page 3 to define a new trigger. You can use a trigger only with the job for which it was created.

   Note: Currently, **Time Event** is the only supported trigger type.

6. Click **Save**.
7. Verify that the listing for the job in the **Jobs** table contains ⏲ in the **Scheduled** column.

Create a Time Trigger

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. The name is specified as **New trigger** by default.

3 Use the **Frequency** field to specify how often the trigger should be repeated (such as a specified number of minutes, hours, or days).

4 Depending on your choice for the frequency interval, different fields appear in the window to enable you to completely specify a frequency for the trigger. For example, if you 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). If you specify **Minutes** in the **Frequency** field, you can specify that the job runs every 5, 10, 15, 20, or 30 minutes. Use these fields to specify the criteria for the trigger interval.

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

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

   For example, if you use the **Frequency** fields to specify that the job runs every hour, and you specify **10:15** in the **Start time** field, the job runs at 10:15, 11:15, 12:15, and so on. If you use the **Frequency** fields to specify that the job runs every 20 minutes, and you specify **09** in the **Start time** field, the job runs at 9:00, 9:20, 9:40, and so on.

6 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 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.

8 Click **Save**.

9 Repeat these steps to create other triggers for the job.

Edit a Scheduled Job

After a job is scheduled, you can edit the schedule for the job. Follow these steps:

1 Select a scheduled job in the **Jobs** table on the **Scheduling** tab. Scheduled jobs with at least one enabled trigger contain ⏲ in the **Scheduled** column. Scheduled jobs with disabled triggers contain a disabled icon ⏸ in the **Scheduled** column.

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

View a Graph 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. The sliders below the graph enable you to zoom into a specific time window within the selected time period.

Each scheduled job is listed on a separate line in the graph. Bars in the chart represent each scheduled execution of a job.

Place your cursor over a bar to display the job name, status, and the date and time of the scheduled execution.

Disable the Schedule for a Job

To prevent a job from running its specified schedule, you can either unschedule the job or disable the triggers. Unscheduling the job prevents the job from running on the defined schedule and also removes the triggers that

are specified for the job. Disabling the triggers prevents the job from running the schedule but preserves the defined triggers.

To unschedule a job, select a scheduled job in the **Jobs** table in the **Scheduling** tab. Click ⊘ from the toolbar or select **Unschedule** from the pop-up menu.

CAUTION! When you unschedule a job, any enabled triggers that are associated with the job are deleted. To unschedule a job and keep the triggers, instead of selecting **Unschedule**, edit the schedule and manually disable the triggers.

To disable the triggers, select a scheduled job in the **Jobs** table in the **Scheduling** tab. Click 🕒 from the toolbar or select **Edit Schedule** from the pop-up menu. In the Edit Schedule dialog box, disable all slider controls in the **Enabled** column of the **Available triggers** table. If you disable all triggers for a job, the disabled icon ⏸ appears in the **Scheduled** column of the **Jobs** table. A 🕒 appears in the column if any of the triggers for the job are enabled.

Run a Job

1. To run a job from the **Scheduling** tab, right-click a job in the **Jobs** table in the **Scheduling** tab.
2. To run the job under your own credentials, click ► in the toolbar or select **Run** from the pop-up menu.
3. To run the job under credentials other than your own, select **Run As** from the pop-up menu.

   The Select Identities window appears, and you can select the user ID under which the job should run.

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

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

View Execution History for a Job

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

1. In the **Scheduling** tab, right-click a job in the **Jobs** table.
2. Select **Execution history** in the pop-up menu or select 📄 from the toolbar.
3. Information about previous runs of the selected job is displayed in the **Monitoring** tab.

View Job Properties

To view properties for a job, select a job in the **Jobs** table and click ▦ in the toolbar or select **Properties** from the pop-up menu. The information in the Job properties window is read-only.

Delete a Job from the Schedule Tab

Jobs remain in the list on the **Scheduling** tab unless you delete them. To delete a scheduled job, follow these steps.

1. Select a job in the **Jobs** 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).

Scheduling Command Line Interface

Scheduling: How to (Command Line Interface)

Run a Job

In order to run a job using the command-line interface, follow these steps:

1 Create a template file for the job definition. This file contains the fields that are needed for a job definition.

```
sas-admin job definitions generate-template --template-filename
```

Here is an example of the job definition template:

```
template:
{
    "name": "Replace with name of the Job Definition",
    "type": "Replace with type of the Job Definition",
    "code": "Replace with code of the Job Definition"
}
```

2 Modify the job definition template file to supply information for the job that you want to run.

3 Use the job definition file that you created in the previous step to create the job definition.

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

This command returns a URI for the job definition.

4 Create a template file for the job request. This file contains the fields that are needed for a job request.

```
sas-admin job requests generate-template --template-filename
```

Here is an example of the job request template file:

```
template:
{
    "version": 0,
    "name": "Replace with name of the Job Request",
    "description": "Replace with description of the Job Request",
    "jobDefinitionUri": "(Mutually exclusive with Definition) Replace with uri to the Job Definition",
    "arguments": null,
    "properties": null
}
```

5 Modify the generated job request template file to supply information for the job that you want to run.

6 Use the job request file that you created in the previous step to create the job request.

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

The command returns an ID for the job request.

7 Execute the request. The job runs immediately.

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

Schedule a Job

In order to schedule a single job using the command-line interface, follow these steps:

1 Create a template file for the job definition. This file contains the fields needed for a job definition.

```
sas-admin job definitions generate-template --template-filename
```

Here is an example of the job definition template:

```
template:
{
    "name": "Replace with name of the Job Definition",
    "type": "Replace with type of the Job Definition",
    "code": "Replace with code of the Job Definition"
}
```

2 Modify the job definition template file to supply information for the job that you want to run.

3 Use the job definition file that you created in the previous step to create the job definition.

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

This command returns a URI for the job definition.

4 Create a template file for the job request. This file contains the fields needed for a job request.

```
sas-admin job requests generate-template --template_filename
```

Here is an example of the job request template:

```
template:
{
    "version": 0,
    "name": "Replace with name of the Job Request",
    "description": "Replace with description of the Job Request",
    "jobDefinitionUri": "(Mutually exclusive with Definition) Replace with uri to the Job Definition",
    "arguments": null,
    "properties": null
}
```

5 Modify the generated job request template file to supply information for the job that you want to run.

6 Use the job request file that you created in the previous step to create the job request.

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

The command returns an ID of the job request.

7 Create a JSON file and include the time triggers for the job. See “Time-Based Triggers” on page 12 for information about specifying the triggers.

8 Schedule the job, and specify the file that contains the time triggers.

```
sas-admin job requests schedule --triggers-file
```

Scheduling: Command-Line Interface Reference

Scheduler Commands

These are the commands to work with schedulers:

Create a scheduler

```
sas-admin job schedulers create [--file_in] [-]
```

The file *file_in* is a JSON file containing the definition for the scheduler. Specify - to provide the scheduler details through stdin.

Update a scheduler

```
sas-admin job schedulers update --id [--file_in] [-]
```

id specifies the ID of the scheduler to update.

file_in is a JSON file containing the updated definition for the scheduler. Specify - to provide the scheduler details through stdin.

Delete a scheduler

```
sas-admin job schedulers delete --id
```

id specifies the ID of the scheduler to delete.

List schedulers

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

Specify `start record` to start displaying schedulers from the specified record. The default is to start from the first record.

Specify `limit limit_number` to limit the number of schedulers to display. the default value is 10.

Specify `sort-by resource_name[, resource_name` to sort the displayed schedulers 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.

Specify `filter expression` to filter the results using the specified regular expression.

Display details for a scheduler

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

id specifies the ID of the scheduler to display.

Generate a scheduler template

```
sas-admin job schedulers generate-template --file_out
```

file_out specifies the template file to create. The file is in JSON format.

Job Template Definition Commands

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

Create a job definition

```
sas-admin job definitions create --file_in [-]
```

The file *file_in* is a JSON file containing the definition for the job template. Specify - to provide the template details through stdin.

Update a job template definition

```
sas-admin job definitions update --id [--file_in] [-]
```

id specifies the ID of the definition to update.

file_in is a JSON file containing the updated definition for the job template. Specify - to provide the definition details through stdin.

Delete a job template definition

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

id specifies the ID of the definition to delete.

List job template definitions

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

Specify `start record` to start displaying definitions from the specified record. The default is to start from the first record.

Specify `limit limit_number` to limit the number of definitions to display. the default value is 10.

Specify `sort-by resource_name[, resource_name` to sort the displayed definitions 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.

Specify `filter expression` to filter the results using the specified regular expression.

Display details for a job template definition

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

id specifies the ID of the definition to display.

Generate a job definition template

```
sas-admin job definition generate-template --file_out
```

file_out specifies the template file to create. The file is in JSON format.

Job Request Commands

Use these commands to manage job requests, which make jobs available for execution and scheduling.

Create a job request

```
sas-admin job requests create --file_in [-]
```

The file *file_in* is a JSON file containing the definition for the job request. Specify - to provide the request details through stdin.

Update a job request

```
sas-admin job requests update --id [--file_in] [-]
```

id specifies the ID of the job request to update.

file_in is a JSON file containing the updated job request. Specify - to provide the request details through stdin.

Delete a job request

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

id specifies the ID of the job request to delete.

List job requests

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

Specify `start record` to start displaying job requests from the specified record. The default is to start from the first record.

Specify `limit limit_number` to limit the number of job requests to display. the default value is 10.

Specify `sort-by resource_name[, resource_name` to sort the displayed job requests 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.

Specify `filter expression` to filter the results using the specified regular expression.

Display details for a job request

```
sas-admin job requests show --id
```

id specifies the ID of the job request to display.

Generate a job definition template

```
sas-admin job requests generate-template --file_out
```

file_out specifies the job request template file to create. The file is in JSON format.

Execute a job request

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

id specifies the ID of the job request to execute.

Schedule a job request

```
sas-admin job requests schedule --id --format --file_in
```

id specifies the ID of the job request to schedule.

format specifies the format of the trigger events in the JSON schedule file. Valid values are passThrough or portable.

file_in is a JSON file containing the event triggers. Specify - to provide the triggers through stdin. You must specify a file if you specify a value for *format*.

Unschedule a job request

```
sas-admin job requests unschedule --id --format --file_in
```

id specifies the ID of the job request to unschedule.

List a job request's history

```
sas-admin job requests list-history --id [--scheduled]
```

id specifies the ID of the job request whose history you want to view.

Specify `scheduled` to view the instances when the job ran as a scheduled job, rather than every instnace when the job executed.

List the details for a job request's history

```
sas-admin job requests show-history --instance-id --sch-id
```

instance-id specifies the ID of the job request instance whose history you want to view.

sch-id specifies the ID of the scheduled job instance.

Action Commands

Use these commands to manage actions which have been registered with job flow scheduling, which then makes them available to be invluded in a job flow.

Create an action

```
sas-admin job actions create [--file_in] [-]
```

The file *file_in* is a JSON file containing the details of the action. Specify - to provide the action details through stdin.

Update an action

```
sas-admin job actions update --id [--file_in] [-]
```

id specifies the ID of the action to update.

file_in is a JSON file containing the updated definition for the action. Specify - to provide the action details through stdin.

Delete an action

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

id specifies the ID of the action to delete.

List actions

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

Specify `start record` to start displaying actions from the specified record. The default is to start from the first record.

Specify `limit limit_number` to limit the number of actions to display. the default value is 10.

Specify `sort-by resource_name[, resource_name` to sort the displayed actions 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.

Specify `filter expression` to filter the results using the specified regular expression.

Display details for an action

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

id specifies the ID of the action to display.

Generate an action template

```
sas-admin job actions generate-template --file_out
```

file_out specifies the template file to create. The file is in JSON format.

Flow Commands

Use these commands to manage flows that are registered with jobflow-scheduling.

Create a flow

```
sas-admin flows create --file_in [-]
```

The file *file_in* is a JSON file containing the definition for the flow. Specify - to provide the request details through stdin.

Update a flow

```
sas-admin flows update --id [--file_in] [-]
```

id specifies the ID of the flowto update.

file_in is a JSON file containing the updated flow. Specify - to provide the flow details through stdin.

Delete a flow

```
sas-admin flows delete --id
```

id specifies the ID of the flow to delete.

List flows

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

Specify `start record` to start displaying flows from the specified record. The default is to start from the first record.

Specify `limit limit_number` to limit the number of flows to display. the default value is 10.

Specify `sort-by resource_name[, resource_name` to sort the displayed flows 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.

Specify `filter expression` to filter the results using the specified regular expression.

Display details for a flow

```
sas-admin flows show --id
```

id specifies the ID of the flowto display.

Generate a flow template

```
sas-admin flows generate-template --file_out
```

file_out specifies the flow template file to create. The file is in JSON format.

Execute a flow

```
sas-admin flows execute --id
```

id specifies the ID of the flow to execute.

Schedule a flow

```
sas-admin flows schedule --id
```

id specifies the ID of the flow to schedule.

Reschedule a flow

```
sas-admin flows reschedule --id --format --file_in
```

id specifies the ID of the flow to reschedule.

Unschedule a flow

```
sas-admin flows unschedule --id --format --file_in
```

id specifies the ID of the flow to unschedule.

List a flow's history

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

id specifies the ID of the flow whose history you want to view.

List the details for a flow's history

```
sas-admin flows show-history --instance-id --sch-id
```

id specifies the ID of the flow whose history you want to view.

sch-id specifies the ID of the scheduled flow instance.

Time-Based Triggers

Use the following syntax when specifying a time-based trigger to schedule a flow or a job.

Here is the general form of the syntax:

```
"triggers":[
              {"type" : "timeevent",
              "active":true,
              "event": {"recurrence":{"type":"recurrence-type"}, options,
                        "hours":hours,
                        "minutes":minutes,
                        "duration":duration,
                        "timeZone":zone,
                        "maxOccurrence":occurrences,}
              }
              ],
```

This list identifies the options that are used for each type of `recurrence` interval.

Minutes

Type

```
"type":"minutely"
```

Options

- `startDate` (specifies when to start the recurrence)
- `endDate` (specifies when to stop the recurrence)

- `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.
- `minutes` (specifies the offset from the beginning of the hour for when the executions start). The maximum value is `skipCount`-1. For example, `"minutes":"10"` specifies that the timing for `skipCount` starts at 10 minutes past the hour.

Here is an example that specifies a trigger that starts at six minutes past the hour and then causes a job to run every 15 minutes (06, 21, 36, 51, and so on).

```
"triggers":[
              {"type" : "timeevent",
              "active":true,
              "event": {"recurrence":{"type":"minutely"}, "skipCount":"15"}
              "minutes":"6", }
              ],
```

Hours

Type

`"type":"hourly"`

Options

- `startDate` (specifies when to start the recurrence)
- `endDate` (specifies when to stop the recurrence)
- `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.

Here is an example that specifies a trigger that starts on June 13, 2018 and causes a job to run every 4 hours:

```
"triggers":[
              {"type" : "timeevent",
              "active":true,
              "event": {"recurrence":{"type":"hourly","startDate":"2018'-'06'-'13",
                   "skipCount":"4"}}
              }
              ],
```

Days

Type

`"type":"daily"`

Options

- `startDate` (specifies when to start the recurrence)
- `endDate` (specifies when to stop the recurrence)
- `skipCount` (specifies how many days pass between executions). For example, a value of 3 specifies that the job runs every 3 days.
- `daysOfWeek` (specifies the days on which the job runs). For example, `"daysOfWeek":"thursday"` specifies that the job runs every Thursday. Valid values are names of days: monday–sunday.

Here is an example of a trigger:

```
"triggers":[
              {"type" : "timeevent",
              "active":true,
              "event": {"recurrence":{"type":"daily"}, "hours":"12","minutes":"0"}
              }
```

```
                ],
```

Weeks

Type

`"type":"weekly"`

Options

- `startDate` (specifies when to start the recurrence)
- `endDate` (specifies when to stop the recurrence)
- `skipCount` (specifies how many weeks pass between executions). For example, `"skipCount": "3"` specifies that the job runs every 3 weeks.
- `daysOfWeek` (specifies the days on which the job runs). For example, `"daysOfWeek":"thursday"` specifies that the job runs every Thursday. Valid values are names of days (monday–sunday).

Example:

```
"triggers":[
                {"type" : "timeevent",
                "active":true,
                "event": {"recurrence":{"type":"weekly"}, "startDate":"2017'-'08'-'15",
"skipCount":"4", "daysOfWeek":"tuesday"}
                }
                ],
```

Months

Type

`"type":"monthly"`

Options

- `startDate` (specifies when to start the recurrence)
- `endDate` (specifies when to stop the recurrence)
- `skipCount` (specifies how many months pass between executions). For example, `"skipCount": "3"` specifies that the job runs every 3 months.
- `daysOfWeek` (specifies the days on which the job runs). For example, `"daysOfWeek":"thursday"` specifies that the job runs on Thursday. Valid values are names of days (monday-sunday). `daysOfWeek`and `dayOfMonth` are mutually exclusive. If `daysOfWeek` is specified, then `occurrence` is required.
- `occurrence` (used with `daysOfWeek` to specify the days on which the job runs) Valid values are first, second, third, fourth, and last.
- `dayOfMonth` (specifies the day of the month on which to run. Valid values are 1–31. A value of 32 specifies the last day of the month. `dayOfMonth` and `daysOfWeek` are mutually exclusive.

Example 1:

```
"triggers":[
                {"type" : "timeevent",
                "active":true,
                "event": {"recurrence":{"type":"monthly"}, "startDate":"2017'-'05'-'14",
"daysOfWeek":"sunday", "occurrence":"second"}
                }
                ],
```

Example 2:

```
"triggers":[
                {"type" : "timeevent",
                "active":true,
```

```
                "event": {"recurrence":{"type":"monthly"}, "dayOfMonth":"15"}
                }
                ],
```

Years

Type

`"type":"yearly"`

Options

- `startDate` (specifies when to start the recurrence)
- `endDate` (specifies when to stop the recurrence)
- `skipCount` (specifies how many years pass between executions)
- `daysOfWeek` (specifies the days on which the job runs). For example, `"daysOfWeek":"thursday"` specifies that the job runs on Thursday. Valid values are names of days (monday-sunday). `daysOfWeek` and `dayOfMonth` are mutually exclusive. If `daysOfWeek` is specified, then `occurrence` is required.
- `occurrence` (used with `daysOfWeek` to specify the days on which the job runs). Valid values are first, second, third, fourth, and last.
- `dayOfMonth` (specifies the day of the month on which to run). Valid values are 1–31. A value of 32 specifies the last day of the month. `dayOfMonth` and `daysOfWeek` are mutually exclusive.
- `monthOfYear` (specifies the month in which the job run)Valid values are january–december.

Example 1:

```
"triggers":[
                {"type" : "timeevent",
                "active":true,
                "event": {"recurrence":{"type":"yearly"}, "daysOfWeek":"friday",
                   "occurrence":"last", "monthOfYear":"june"}
                }
                ],
```

Example 2:

```
"triggers":[
                {"type" : "timeevent",
                "active":true,
                "event": {"recurrence":{"type":"yearly"}, "dayOfMonth":"32", "monthOfYear":"may"}
                }
                ],
```

Specified dates

Type

`"type":"dateList"`

Options

- `startDate` (specifies when to start the recurrence)
- `endDate` (specifies when to stop the recurrence)
- array of dates in the form `yyyy '-' mm '-' dd`

Example:

```
"triggers":[
                {"type" : "timeevent",
                "active":true,
                "event": {"recurrence":{"type":"dateList"}, "2017 '-' 06 '-' 13",
                   "2017 '-' 08 '-' 02", "2017 '-' 10 '-' 05"}
```

```
                    }
                    ],
```

For the `"hours":hours` parameter, specify a set of hours when the job runs. You can specify a list of hours separated by commas (1,2,3), a range of hours (2–4), a combination of a range and a list (1–3,5,7), or an asterisk to specify all hours. If you specify a recurrence of hourly, only the first value is used, and it must be equal to or less than the value of skipCount. If you specify a recurrence of minutely, the hours parameter is ignored.

For the `"minutes":minutes` parameter, specify a set of minutes when the job runs. You can specify a list of minutes separated by commas (0,10,30), a range of minutes (20–25), a combination of a range and a list (0,10–15), or an asterisk to specify every minute. If you specify a recurrence of minutely, only the first value is used, and it must be equal to or less than the value of skipCount.

For the `"duration":duration` parameter, specify the number of minutes the event is to remain true.

For the `"timeZone":zone` parameter, specify the time zone to use when evaluating the time trigger. Specify the value using the Olson time zone database, in the form `region/city`. For example, `America/New_York`.

For the `"maxOccurrence":occurrences` parameter, specify the maximum number of times the job can execute.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. Copyright © 2018, SAS Institute Inc., Cary, NC, USA. All Rights Reserved. August 2019 3.4-P1:caljobs

THE POWER TO KNOW®