

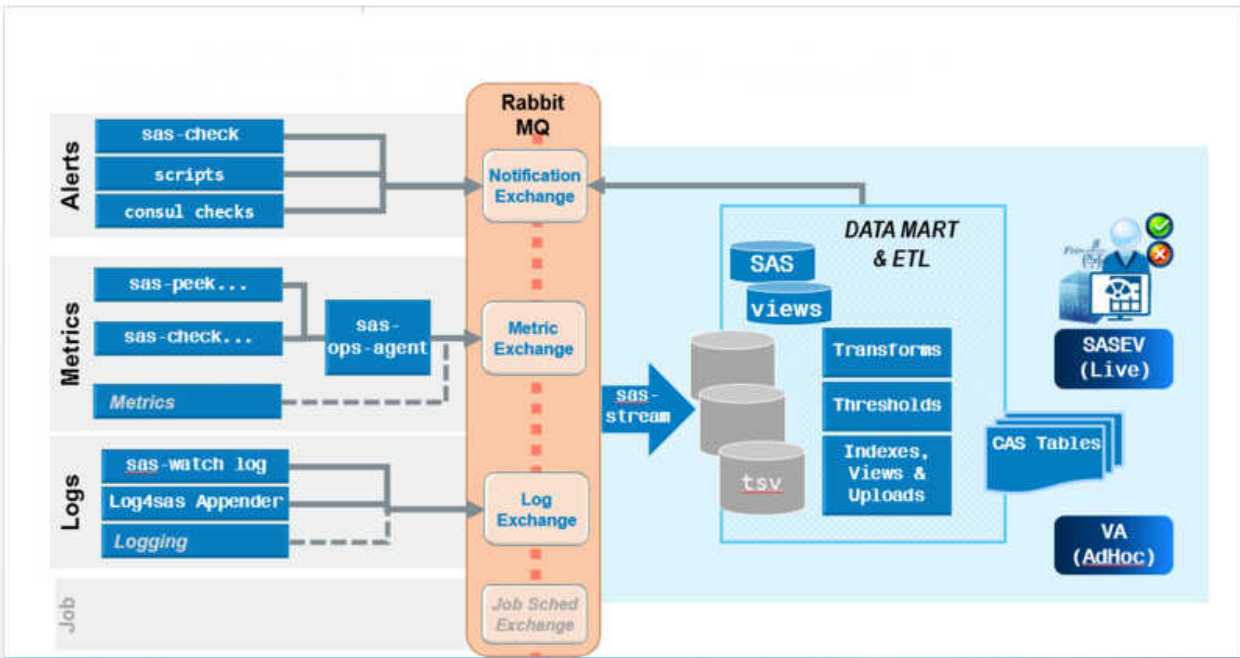


SAS[®] Viya[®] 3.3 Administration: Operations Infrastructure

Operations Infrastructure: Overview

The operations infrastructure implements an event-driven architecture that underlies several areas of SAS Viya administration, most notably monitoring and logging. An event in this architecture represents some unit of information, such as a metric reading (for example, CPU usage at a particular time) or a log message (for example, a microservice has failed). The architecture is flexible and adaptable, because it keeps the producer of the events and the consumers of the events separate. The producer of an event collects information such as a system metric or a log message that publishes that information to a message exchange without knowledge about the consumer of the information. The information consumer looks for specified types of information and retrieves the information when it is found. Types of consumers include extract, transform, and load (ETL) processes or a data mart. Likewise, the consumer has no knowledge about the source of the information.

The operations infrastructure implements the producer portion of this architecture by using components such as `sas-peek`, `sas-check`, and `sas-watch` to collect system performance data from resources and log files. These producers then publish events to the appropriate RabbitMQ message exchange. Another component, `sas-stream`, performs the consumer role by reading events from the message exchange and writing them to the data mart. An ETL process runs periodically and loads data from the data mart into the CAS server, where it is then surfaced in SAS Environment Manager. For example, both the Logging view and the Machines view of SAS Environment Manager surface data that is collected through the operations infrastructure.



The infrastructure manages these event types:

Metrics events

measurements describing the performance of a resource, component, or subsystem

Check events

simple threshold checks of a metric, with return codes to indicate whether the metric passed the threshold

Log message events

messages written to a log that is monitored by the infrastructure

Notification events

important events that require attention

The `sas-peek` component is a collector that gathers performance metric data for a specific resource or service. For example, `sas-peek cpu` collects CPU metrics, and `sas-peek cas` collects metrics for the CAS service.

The `sas-watch` component monitors an object such as a log. For example, `sas-watch log` monitors the state of the log files for SAS Viya resources and services.

The `sas-check` component captures metric data and compares it to a defined threshold value. Then the component reports whether the metric value passed or exceeded the warning or critical threshold.

You do not have to run these components (`sas-peek`, `sas-watch`, and `sas-check`) manually. They are configured and run automatically.

How To: Operations Infrastructure Command Line

Overview

Operations infrastructure tasks are performed through the `sas-ops` command. Although you can run these commands manually, they are configured and run automatically as part of the operations infrastructure processes. The functions that are provided by this command follow:

- streaming of operations information, including notifications, alerts, metric data, and log messages
- validation of the SAS Viya environment and the operations infrastructure
- information about the SAS Viya environment, including the services, the machines, and the environment

Here is the format of the command:

```
sas-ops command --option
```

You must be the SAS install user in order to run the command.

Stream the Operations Information

Stream the Alert Messages

Use the `sas-ops alerts` command to stream the alert messages from SAS applications and components.

The default behavior is to stream alert messages until you stop the command. You can use the option `--timeout duration` to stream messages only for a specified time. The duration is specified using the format `0h0m0s0ms`, although subsets of this format are also allowed. See [“Time Format” on page 17](#) for details and examples. For example, `--timeout 5m30s` specifies that alert messages are streamed for 5 minutes and 30 seconds.

Use the `--last n` option to display only a specified number (specified as *n*) of the most recent messages.

Use the `--format` option to specify the format for the data.

```
--format json
```

Streams the alerts in JSON format, with the data and parameters on one line.

```
--format pretty
```

Streams the alerts in JSON format, with the data and parameters on separate lines.

```
--format line
```

Streams the alerts on a single line. The timestamp is listed first.

```
--format block
```

```
--format message
```

Stream the Log Events

Use the `sas-ops logs` command to stream the log messages that are generated by SAS Viya applications and services.

The default behavior is to stream log messages until you stop the command. You can use the option `--timeout duration` to stream messages only for a specified time. The duration is specified using the format `0h0m0s0ms`, although subsets of this format are also allowed. See [“Time Format” on page 17](#) for details and examples. For example, `--timeout 5m30s` specifies that messages are streamed for 5 minutes and 30 seconds.

The default behavior is to stream messages to the terminal on which the command was issued. However, you can use the `--format` option to specify a different format for the messages.

```
--format json
```

Streams the log messages in JSON format, with the message text and parameters on one line.

```
--format pretty
```

Streams the log messages in JSON format, with the message text and parameters on separate lines.

```
--format line
```

Streams the log message text and parameters on a single line. The timestamp is listed first.

4

- `--format file`
Streams the log message text and parameters on a single line.
- `--format term`
Streams the log message text and parameters in terminal format. The message level is listed first.
- `--format plain`
Streams the log message text and parameters with no tagging to indicate different parts of the message. The abbreviated message level is listed first.
- `--format logfmt`
Identifies the parts of the message using the format *message-part=string*. An example is `timestamp=2017-10-20T11:14:56.000000-04:00`. The messages are color-coded, depending on their level.
- `--format template`
Streams the log message text and parameters with no tagging to indicate different parts of the message.
- `--format event`
Streams the log messages in JSON format as used by the event service.

Because many log messages are produced in a typical environment, you can use these options to filter the message stream to include only those messages of interest.

- `--match regular-expression`
Streams the messages that match the specified regular expression.
- `--match-file file`
Streams the messages that match the regular expressions that are contained in the specified file.
- `--min-level level`
Streams the messages that are at the specified level or a higher level.

Valid values for *level* are trace, debug, info, warn, error, fatal, and none.
- `--source source`
Streams the messages only from the specified source.

Stream the Metric Data

Use the `sas-ops metrics` command to stream the metric data that is generated by SAS Viya applications and services.

The default behavior is to stream data until you stop the command. You can use the option `--timeout duration` to stream data only for a specified time. The duration is specified using the format `0h0m0s0ms`, although subsets of this format are also allowed. See [“Time Format” on page 17](#) for details and examples. For example, `--timeout 5m30s` specifies that data is streamed for 5 minutes and 30 seconds.

The default behavior is to stream metric data to the terminal on which the command was issued. However, you can use the `--format` option to specify a different format for the data.

- `--format json`
Streams the metric data in JSON format, with the data and parameters on one line.
- `--format pretty`
Streams the metric data in JSON format, with the data and parameters on separate lines.
- `--format line`
Streams the metric data on a single line. The timestamp is listed first.
- `--format property`
Streams the metric data on a single line.
- `--format event`
Streams the metric data in JSON format as used by the event service.

Stream the Notification Messages

Use the `sas-ops notifications` command to stream notification messages from SAS applications and components.

The default behavior is to stream notifications until you stop the command. You can use the option `--timeout duration` to stream notifications only for a specified time. The duration is specified using the format `0h0m0s0ms`, although subsets of this format are also allowed. See [“Time Format” on page 17](#) for details and examples. For example, `--timeout 5m30s` specifies that notifications are streamed for 5 minutes and 30 seconds.

Publish a Notification Message

Use the `sas-ops notify message` command to publish a notification message. You can use the `--level` option to specify the level of the message. Supported level values are `info`, `warn`, or `alert`.

An example command is `sas-ops notify The server will be rebooted. --level alert`.

Verify the Status of Your SAS Viya Environment

The Operations infrastructure provides the `validate` command to enable you to perform checks on your Viya environment in order to locate problems. To use the command, SAS Viya does not have to be running. However, the SAS Configuration Server and RabbitMQ must be running.

The syntax of this command is

```
sas-ops validate --level level --json --tags string --verbose.
```

The value for *level* specifies the complexity of the validation checks. Each level performs the checks both at its own level and the previous level. For example, specifying `--level 2` causes both the level 1 and level 2 checks to be performed.

Three levels of validation are available, in increasing order of complexity.

- 1
 - Verifies a connection to the SAS Configuration Server (Consul) and to the RabbitMQ exchanges `sas.application`, `sas.log`, `sas.metric`, and `sas.notification`. This level of validation ensures that you can perform validation checks at levels 2 and 3.
- 2
 - Verifies that the operation infrastructure is operating properly. These checks are performed:
 - Verifies the SAS Configuration Server (Consul) services on each machine. The checks verify the following: the disk space used is not greater than 95%, the memory used is not greater than 95%, and the SAS Configuration Server is running.
 - Verifies that the operations data mart ETL is running properly by performing checks on the standard regularly scheduled ETL jobs. The information displayed is the same information that is generated by the `sas-ops datamarts` command.
 - Verifies the status of the operations services, including all instances of the `ops-agent`, `alert-track`, `watch-log`, `ops-agentsrv`, and `stream-evdm` services.
- 3
 - Verifies the status of CAS, the HTTP service, and the authorization service. These checks are performed:
 - Verifies the status of the CAS servers by locating the defined servers and verifying that they are running.
 - Verifies HTTP connectivity by attempting to connect to the base HTTP address.
 - Verifies that the authorization service is working by attempting to obtain an OAuth token for the `sas-ops` command.

Display the Environment Information

Display Information about the Data Mart

Use the `sas-ops datamart` command to display metric and status information about the data mart. Here is typical information that is returned:

```
evdm
  status
    etl_driver
      casLogLoad_SYSCCRC : 0
      casLogSearchLoad_SYSCCRC : 4
      casLogconnect_SYSCCRC : 0
      casMetricConnect_SYSCCRC : 0
      casMetricLoad_SYSCCRC : 0
      endtime : 2017-10-18T14:11:29.72-04:00
      jobExitRC : 0
      osLogRC : 1
      osMetricRC : 0
      readMetricTransform_SYSCCRC : 0
      starttime : 2017-10-18T14:11:05.881639-04:00
      status : ok
      statusRC : 4
    rolloff
      endtime : 2017-10-18T02:00:10.69-04:00
      jobExitRC : 42
      starttime : 2017-10-18T02:00:00.034636-04:00
      status : error
    ziptsv
      deleteOldZips_SYSCCRC : 0
      endtime : 2017-10-18T03:01:27.89-04:00
      jobExitRC : 0
      osZipTSVRC : 0
      starttime : 2017-10-18T03:00:00.023883-04:00
      status : ok
      statusRC : 0
      updateInventory_SYSCCRC : 0
      zipTSV_SYSCCRC : 0
```

The results include information about the following three jobs that are used by the data mart:

- The `etl_driver` job, which processes the metric data and the log data, and loads the data into the data mart

`casLogLoad_SYSCCRC`

Return code from SAS for loading the log data into CAS for the log phase of the ETL job

`casLogSearchLoad_SYSCCRC`

Return code from SAS for updating the CAS search index for the log phase of the ETL job

`casLogconnect_SYSCCRC`

Return code from SAS for connecting to CAS for the log phase of the ETL job

`casMetricConnect_SYSCCRC`

Return code from SAS for connecting to CAS for the metric phase of the ETL job

`casMetricLoad_SYSCCRC`

Return code from SAS for loading the metric data into CAS for the metric phase of the ETL job

endtime

End time of the ETL job

jobExitRC

Return code from the operating system that is written by emi-runsasjob

osLogRC

Maximum return code (either 0,1, or 2) from SAS for the log phase

osMetricRC

Maximum return code (either 0,1, or 2) from SAS for the metric phase

readMetricTransform_SYSCCRC

Maximum return code from SAS for reading the raw TSV file for the metric phase

starttime

Start time of the ETL job

status

Status message (ok, error, or warning) that is written by emi-runsasjob at the end of the job

statusRC

Maximum return code from SAS

- Nightly rolloff job, which removes old data mart data from CAS

endtime

End time of the rolloff job

jobExitRC

Return code from the operating system that is written by emi-runsasjob

starttime

Start time of the rolloff job

status

Status message (ok, error, or warning) that is written by emi-runsasjob at the end of the job

- Nightly ZIPTSV job, which archives old TSV files into ZIP format, removes old archive files, and updates the data mart inventory

deleteOldZips_SYSCCRC

Return code from SAS for deleting old zipped TSV files

endtime

End time of the ZIPTSV job

jobExitRC

Return code from the operating system, written by emi-runsasjob

osZipTSVRC

Maximum return code (either 0,1, or 2) from SAS for archiving the TSV files

starttime

Start time of the ZIPTSV job

status

Status message (ok, error, or warning) that is written by emi-runsasjob at the end of the job

statusRC

Maximum return code from SAS

updateInventory_SYSCCRC

Maximum return code from SAS for updating all inventory files (one per resource type)

zipTSV_SYSCCRC

Maximum return code from SAS to Zip TSVs and delete old ZIP files, written by SAS

Access Information about Your Environment

Use the `sas-ops env` command to display information about the machine (on which you run the command), SAS environment variables, and the SAS Viya deployment.

Here is typical information that is returned:

```
Host Information:
  Full hostname           : full_hostname
  Short hostname         : short_hostname
  Consul node name       : full_hostname

SAS environment variables:
  CONSUL_HTTP_ADDR = https://localhost:8501

SAS Viya Deployment:
  Install user           : sas
  Deployment ID          : viya
  SAS root                : /opt/sas
  Deployment root        : /opt/sas/viya
  Home directory         : /opt/sas/viya/home
  Config directory       : /opt/sas/viya/config
  Log directory          : /opt/sas/viya/config/var/log
  SPRE directory         : /opt/sas/spre
```

View Machine Information

Use the `sas-ops info` command to obtain information about each machine in your SAS Viya environment. For each machine in your environment, the command returns this information:

- machine identity
- packages installed on the machine
- system metrics
- system limits

The information returned by this command is the same information that is displayed on the Machines page in SAS Environment Manager.

View Information about Services

Use the `sas-ops services` command to view information about the services in your environment.

Run the command `sas-ops services` with no options to display a list of all SAS Viya services that are currently active in your environment.

Run the command `sas-ops services --detail service-name` to view detailed information about a specified service. Here is typical information that is returned:

```
{
  "ID": "e77ab2dc-b6c6-4a4d-af4e-bf3712de3c98",
  "Node": "vdmml-tue-17w47-ud.uda.sashq-r.openstack.sas.com",
  "Address": "10.104.29.192",
  "Datacenter": "",
  "TaggedAddresses": {
    "lan": "10.104.29.192",
    "wan": "10.104.29.192"
  }
},
```



```

"NodeMeta": {},
"ServiceID": "compute-10-104-29-192",
"ServiceName": "compute",
"ServiceAddress": "vdmm1-tue-17w47-ud.uda.sashq-r.openstack.sas.com",
"ServiceTags": [
  "proxy",
  "rest-commons",
  "https",
  "jobExecution-provider",
  "jobExecution-provider-Compute",
  "dataSources-provider",
  "dataSources-provider-Compute",
  "dataTables-provider",
  "dataTables-provider-Compute",
  "rowSets-provider",
  "rowSets-provider-Compute",
  "contextPath=/compute"
],
"ServicePort": 39504,
"ServiceEnableTagOverride": false,
"CreateIndex": 16797,
"ModifyIndex": 16797
}

```

Run the command `sas-ops services --health service-name` to perform the health checks on each instance of the specified service:

Disk utilization of SAS Config filesystem

The check passes if disk usage does not exceed 95%.

Memory percent free

The check passes if memory usage does not exceed 95%.

Serf Health Status

The check passes if the SAS Configuration service is running.

Service '*service-name*' check

The check passes if the service is running,

View Information about Metric Tasks

The operations agent (`sas-ops-agent`) runs a defined set of tasks to collect system metrics and to publish the metric data to RabbitMQ. Use the `sas-ops tasks` command to view a list of the tasks that are performed by the agent and the frequency of the task that is run. For more information about the agent and the tasks performed by the agent, see [“How To: Operations Infrastructure Agent Command Line” on page 10](#).

Here is an example of the information that is returned by the `sas-ops tasks` command:

Task Name	Description	Frequency
-----	-----	-----
CASMetrics	CAS performance metrics (level=2)	1m0s
CheckCpu	Check CPU activity less than 95% busy	1m0s
CheckFileSystem	Check file system space less than 90% used	1m0s
CheckMemory	Check memory less than 95% used	1m0s
EmiSweeper	Retry publishing any payloads that failed to publish earlier	1h0m0s
FileSystemMetrics	Host file system metrics (level=2)	1m0s
HostEnvSnapshot	Host environment snapshot	02:25
LogFileArchive	Archive daily	04:00
NetworkInterfaceMetrics	Host network interface metrics (level=2)	1m0s

OpsAgentActivity	Internal sas-ops-agent activity monitor	2m0s
OpsAgentTaskStatistics	Internal sas-ops-agent task statistics activity monitor	4m0s
PostgresMetrics	Postgres metrics (level=2)	1m0s
RabbitmqMetrics	RabbitMQ performance metrics (level=2)	1m0s
SpringBootMetrics	Spring Boot performance metrics (level=2)	1m0s
SpringBootMetricsLevel3	Spring Boot performance metrics (level=3)	4h0m0s
SystemMetrics	Host system metrics (level=2)	1m0s
TopProcessMetrics	Top CPU process consumers (level=2)	1m0s
genAudit	Extract audit records. Generate a CSV files for given applications	2h0m0s
registerOpsAgentServiceTask	Register Ops-Agent service task	5m0s
registerOpsServiceTask	Register Ops service task	5m0s

How To: Operations Infrastructure Agent Command Line

Overview

The operations infrastructure agent runs a defined set of tasks. Each task is defined as a combination of a command to execute and information about how to publish the output of the command. Most tasks invoke the sas-peek or sas-check components and publish the output as an event to RabbitMQ.

The list of tasks for the agent to run are provided from the SAS Infrastructure Data Server or from a file to be read. The task definition also includes other attributes such as how often the task should be run.

Summary of the Default Task List

The following tasks are provided by default on the agent server:

Task name	Command	Description
CASMetrics	<code>sas-peek cas -level 2</code>	Collects the CAS performance metrics (at level 2) every minute.
CheckCPU	<code>sas-check cpu -warning 95 -metric percentCpuBusy</code>	Checks the CPU activity and verifies that it is lower than 95% busy. It runs every minute.
CheckFileSystem	<code>sas-check filesystem -warning 90 -metric percentUsedBytes -inctype xfs</code>	Checks the file system and verifies that it is less than 90% used. It runs every minute.
CheckMemory	<code>sas-check memory -warning 95 -metric percentUsed</code>	Checks the system memory and verifies that it is less than 95% used. It runs every minute.
EmiSweeper (server)	<code>emi-event-sweep run -delete -verbose</code>	Attempts to publish any component outputs that were not published. It runs every hour.
FileSystemMetrics	<code>sas-peek filesystem -level 2</code>	Collects the host file system metrics (level 2). It runs every minute

Task name	Command	Description
EvdmDatamartEtl (server)	<code>emi-runsasjob -pgm etl_driver.sas -datamart evdm</code>	Specifies the ETL driver for the data mart. It runs every 5 minutes.
EvdmDatamartRollOff (server)	<code>emi-runsasjob -pgm rolloff.sas -datamart evdm</code>	Rolls off old data from the data mart. It runs at 2 AM every day.
evdmDatamartzipTSV (server)	<code>emi-runsasjob -pgm ziptsv.sas -datamart evdm</code>	Zips data from the data mart to a TSV file. It runs at 3 AM every day.
HostEnvSnapshot	<code>emi-sysinfo</code>	Takes a snapshot of the host environment at 2:25 AM every day.
LogfileArchive	<code>sas-archive</code>	Archives the logs from the previous day. It runs at 4 AM every day.
NetworkInterfaceMetrics	<code>sas-peek network -level 2</code>	Collects the host network interface metrics. It runs every minute.
OpsAgentActivity (server)	<code>sas-event-pub-exchange sas.metric</code>	Monitors the activity of the operations infrastructure agent. It runs every 2 minutes.
OpsAgentTaskStatistics (server)	<code>sas-event-pub-exchange sas.metric</code>	Monitors the activity of the operations infrastructure agent task statistics. It runs every 4 minutes.
PostgresMetrics	<code>sas-peek postgres -level 2</code>	Collects the metrics from the Infrastructure Data Server (level 2). It runs every minute.
RabbitmqMetrics	<code>sas-peek rabbitmq -level 2</code>	Collects the metrics from RabbitMQ (level 2), It runs every minute.
SpringBootMetrics	<code>sas-peek springboot -level 2</code>	Collects the metrics from Spring Boot (level 2). It runs every minute.
SpringBootMetricsLevel3	<code>sas-peek springboot -level 3</code>	Collects the metrics from Spring Boot (level 3). It runs every minute.
SystemMetrics	<code>sas-peek system -level 2</code>	Collects the host system metrics (level 2). It runs every minute.
TopProcessMetrics	<code>sas-peek top -level 2</code>	Collects the top consumers of CPU processes (level 2). It runs every minute.
registerOpsAgentSvrServiceTask (server)	<code>emi-util register -id sas.ops-agentsrv</code>	Registers the ops-agentSvr service task. It runs every 5 minutes.
genAudit	<code>genAudit.sh -a reports,folders,dataPlans,casManagement,casAccessManagement,--user-id -l 1000 -d 7</code>	Extracts the audit records for reports, folders, data plans, CAS management, and CAS access management, and generates a CSV file. It runs every 2 hours.

Controlling the Tasks

List the Tasks

To list the tasks that are in the current task list and that are loaded to the SAS Configuration Server, run the command `sas-ops-agent list`. The command returns a list of the tasks that are in the current list. The command also displays the frequency and a brief description for each task. Here is typical output:

```
Task name: CASMetrics
  Freq.....: 1m0s
  Description..: CAS performance metrics (level=2)
Task name: CheckCpu
  Freq.....: 1m0s
  Description..: Check CPU activity less than 95% busy
Task name: CheckFileSystem
  Freq.....: 1m0s
  Description..: Check file system space less than 90% used
Task name: CheckMemory
  Freq.....: 1m0s
  Description..: Check memory less than 95% used
Task name: EmiSweeper
  Freq.....: 1h0m0s
  Description..: Retry publishing any payloads that failed to publish earlier
Task name: FileSystemMetrics
  Freq.....: 1m0s
  Description..: Host file system metrics (level=2)
Task name: HostEnvSnapshot
  Freq.....: 02:25
  Description..: Host environment snapshot
```

Add a Task

You can define a new task to include in a task list. Each task is specified by a task attribute list, which is a part of a task list. Here is the format of a task attribute list:

```
{
  "version": 1,
  "taskName": "TaskExample",
  "description": "Non-persistent task",
  "hostType": "any",
  "runType": "once",
  "frequency": "0s",
  "maxRunTime": "2m0s",
  "timeOutAction": "cancel",
  "errorAction": "cancel",
  "command": "emi-stress",
  "commandArgs": "-version",
  "commandType": "sas",
  "publisherType": "amqp",
  "publisherCommand": "sas-event-pub",
  "publisherArgs": "-exchange sas.metric"
}
```

Here are the attributes to specify when defining a task:

version

The version of the task attribute list.

taskName

A name that you assign to the task. The name must be a single string, and it must start with an alphabetic character or an underline (), followed by one or more alphanumeric characters.

description

(Optional) A description of the task.

hostType

The operating system on which the task can run. Here are the valid values:

linux

Specifies that the task can run only on a 64-bit Linux system.

windows

Specifies that the task can run only on a 64-bit Windows system.

any

Specifies that the task can run on either a Linux or Windows system. If you specify "hostType": "any", you must use a command name that does not have an extension.

Note: If you do not specify a value for this attribute, the value is the current operating system.

runType

How often the task runs. Here are the valid values:

once

Specifies that the command for the task is run only one time, after the agent server starts.

time

Specifies that the command runs at the time that is provided in the frequency attribute.

start_time

Specifies that the command starts when the agent server starts, and also runs at the time that is provided in the frequency attribute.

periodic

Specifies that the command runs according to the time interval that is specified in the frequency attribute. This is the default value.

periodic_aligned

Specifies that the command runs according to the time interval that is specified in the frequency attribute. The interval starts at midnight.

frequency

When the command in the task should run. See [“Time Format” on page 17](#) for details about specifying times. Here are the valid values:

0h0m0s0ms

Specifies a time interval.

YYYY-MM-DDTHH:MM:SS.ssssssZhh:mm

Specifies a specific time.

maxRunTime

How long to wait after the task starts before taking the action that is specified in the `timeOutAction` attribute.

timeOutAction

The action to take if the task times out (after the interval that is specified in the `maxRunTime` attribute). Here are the valid values:

cancel

Specifies that the task processes are killed and the task is not rescheduled. This is the default.

quiesce

Specifies that the task is completed, but it is not scheduled to run again until the agent server is restarted.

restart

Specifies that the task processes are killed and then the task is restarted.

errorAction

The action to take if the task completes with a nonzero return code. Here are the valid values:

cancel

Specifies that the task is not rescheduled. This is the default.

restart

Specifies that the task restarts at the next scheduled time.

command

The command to run. Do not include a path or command arguments. If you specify `windows` for the `hostType` attribute, the command should include the extension `.exe`. If it does not include the extension, it will automatically be added.

commandArgs

The arguments for the command. If you specify an invalid argument and the command end with a nonzero return code, the task is disabled and will not be rescheduled until the agent server restarts.

commandType

The type of command. Here are the valid values:

ext

Specifies that the command is an external, user-supplied command. The output for this type of command must be written to `stdout`. If the output is to be published, it must use the JSON metric event format.

sas

Specifies that the command is provided by SAS. These commands include `sas-peek`, `sas-check`, `sas-logwatch`, and so on.

int

Specifies an internal task.

Note: Do not use this value.

publisherType

The type of publisher program for the task. Here are the valid values:

amqp

Specifies that the collector data is published using `sas-event-pub` to the AMQP server. This the default value if the `publishParm` attribute is set to `-exchange`.

http

Specifies that the collector data is published using `sas-event-pub` to an HTTP end point. This the default value if the `publishParm` attribute is set to `-url`.

file

Specifies that the collector data is published using `sas-event-pub` to a file. This the default value if the `publishParm` attribute is set to `-file`.

olog

Specifies that the collector data is published using `emi-outlog` to a file. This the default value if the `publishPgm` attribute is set to `emi-outlog`.

ext

Specifies that the collector data is published to an external program other than SAS.

none

Specifies that the collector data is not published. This is the default.

publisherCommand

Specifies the name of the publishing program that receives the output from the program that is specified in the task definition. The publishing program receives input on its stdin from the stdout of the task command. Do not include a path or command arguments. If the host system for the publisher is Windows, the command should include the extension `.exe`. If it does not include the extension, it will automatically be added.

publisherArgs

Specifies the arguments for the publisher program.

Change the Frequency of a Task

You can change how often a defined and recurring task runs by modifying the task definition. Some tasks are defined to run at a specified interval (for example, every two minutes), and some are defined to run at a specified time (for example, 2 AM). To make the change, you export the current task list, make the modifications, and import the task list back to the server. Follow these steps.

- 1 Export the current task list. Use this command:

```
sas-ops-agent export -tasks yourExportedTasksFileName.json
```

- 2 Modify the task file.

The definition for a task that runs at a specified time interval contains the line `"runType": "periodic",`. To change how often this type of task runs, change the line

```
"frequency": "Ns"
```

Substitute your time specification for `Ns`. Use the format

```
NhNmNsNms
```

For example, `2m30s` specifies that the task runs every 2 minutes and 30 seconds.

The definition for a task that runs at a specified time contains the line `"runType": "time",`. To change how often this type of task runs, change the line `"frequency": "YYYY-MM-DDTHH:MM:SS",` to use your time specification. For example, `03:30` specifies that the task runs at 3:30 AM.

- 3 Import the modified task list, overwriting the existing list. Use this command:

```
sas-ops-agent import -tasks yourNewTasksFileName.json -f
```

- 4 Restart the `sas-ops-agent` service. Use this command:

```
sas-viya-ops-agent-default restart
```

How To: ETL and Data Mart Operations

Modify Property Values

You can manually change the property values that are used by the ETL and data mart processes. Use the command `dm-admin --datamart evdm set property-name=property-value`

Here are the properties that you can specify:

EMI_CAS_LOAD

Specifies that the collected alert and notification messages are loaded into CAS. The default value is Y.

EMI_CAS_LOAD_LOGS

Specifies that the collected log messages are loaded into CAS. The default value is Y.

EMI_CAS_LOAD_METRICS

Specifies that the collected metric data is loaded into CAS. The default value is Y.

EMI_CAS_RETAIN_DAYS

Specifies the number of days that the metric and log data are retained in CAS. The default value is 3.

EMI_DELETE_TSVZIP_DAYS

Specifies the number of days to keep the metric and log data on disk in the data mart. The default value is 30.

EMI_ZIP_TSV_DAYS

Specifies the number of days before raw TSV files are compressed into a ZIP file. It also applies to the number of days to keep SAS log files that are generated by standard data mart batch jobs. The default value is 1.

Summary of ETL Return Codes

Here are possible return codes for ETL jobs:

Return code	Meaning
93	A required CAS table could not be initialized because the specified initialization job does not exist.
94	A required CAS table does not exist after the specified initialization job has been run.
95	A required SAS table could not be initialized because the specified initialization job does not exist.
96	A required SAS table does not exist after the specified initialization job has been run.
156	A lock on a data set could not be obtained within the specified time limit.
195	CAS configuration information has not been provided or located.
196	A CAS connection could not be established.
197	The contents of the data mart lock file could not be released when releasing the data mart lock.
198	The data mart lock could not be released.
199	The data mart lock could not be released because it is not locked by this process.
298	A data mart lock could not be obtained because the data mart is locked by another process.
299	A data mart lock could not be obtained because the data mart lock file cannot be renamed.

Operations Infrastructure Reference

Time Format

Time can be specified in various formats. Here are the general forms of time specifications:

- An amount of time, specified in the format 0h0m0s0ms
- A specific time and date, specified in the format YYYY-MM-DDTHH:MM:SS.ssssssZhh:mm

Subsets and variants of each form are allowed. Here are some examples:

Specification	Time specified
"5m20s"	5 minutes and 20 seconds
"2h"	2 hours
"426s"	426 seconds
"3:55:05.29754"	3:55 and 5.29754 seconds, in the current time zone, on any day Example: 0000-01-01 03:55:05.29754 -0500 EST
"14:18:34"	14:18 and 34 seconds, in the current time zone, on any day Example: 0000-01-01 14:18:34 -0500 EST
"14:18:34.38"	14:18 and 34.38 seconds, in the current time zone, on any day Example: 0000-01-01 14:18:34.38 -0500 EST
"14:28"	14:28, in the current time zone, on any day Example: 0000-01-01 14:28:00 -0500 EST
"5T14:25:04.54731"	14:25 and 4.54731 seconds, in the current time zone, on the 5th day of any month. Example: 0000-01-05 14:25:04.54731 -0500 EST
"05T09:37"	9:37, in the current time zone, on the 5th day of any month. Example: 0000-01-05 09:37:00 -0500 EST
"05T09"	9:00, in the current time zone, on the 5th day of any month. Example: 0000-01-05 09:00:00 -0500 EST
"05-23T17:26:09.237"	17:26 and 9.237 seconds, in the current time zone, on 23 May of any year. Example: 0000-05-23 17:26:09.237 -0500 EST
"04-05T09:37"	9:37, in the current time zone, on 5 April of any year. Example: 0000-04-05 09:37:00 -0500 EST
"04-16T21"	21:00, in the current time zone, on 16 April of any year. Example: 0000-04-16 21:00:00 -0500 EST

Specification	Time specified
"03-15"	15 March, in the current time zone, of any year. Example: 0000-03-15 00:00:00 -0500 EST
"2015-04-05T14:18"	14:18, in the current time zone, on 5 April 2015. Example: 2015-04-05 14:18:00 -0400 EDT
"2015-04-05T18"	18:00, in the current time zone, on 5 April 2015. Example: 2015-04-05 18:00:00 -0400 EDT
"2018-01-16"	16 January, 2018, in the current time zone. Example: 2018-01-16 00:00:00 -0500 EST
"05:21:19Z"	5:21 and 19 seconds UTC, on any day. Example: 0000-01-01 05:21:19 +0000 UTC
"08:22:00.21Z"	8:22 and 0.21 seconds UTC, on any day. Example: 0000-01-01 08:22:00.21 +0000 UTC
"14:25:04.54731Z"	14:25 and 4.54731 seconds UTC, on any day Example: 0000-01-01 14:25:04.54731 +0000 UTC
"3:55Z"	3:55 UTC, on any day Example: 0000-01-01 03:55:00 +0000 UTC
"17T14:25:04.54731Z"	14:25 and 4.54731 seconds UTC, on the 17th day of any month Example: 0000-01-17 14:25:04.54731 +0000 UTC
"05T09:37Z"	9:37 UTC, on the 5th day of any month Example: 0000-01-05 09:00:00 +0000 UTC
"05T09Z"	9:00 UTC, on the 5th day of any month Example: 0000-01-05 09:00:00 +0000 UTC
"05-23T17:26:09.23731Z"	17:26 and 9.23731 seconds UTC, on 23 May of any year Example: 0000-05-23 17:26:09.23731 +0000 UTC
"11-10T06:44Z"	6:44 UTC, on 10 November of any year Example: 0000-11-10 06:44:00 +0000 UTC
"09-21T16Z"	16:00 UTC, on 21 September of any year Example: 0000-09-21 16:00:00 +0000 UTC
"2019-11-07T13:21Z"	13:21 UTC, on 7 November, 2019 Example: 2019-11-07 13:21:00 +0000 UTC
"2019-11-07T11Z"	11:00 UTC, on 7 November, 2019 Example: 2019-11-07 11:00:00 +0000 UTC

Specification	Time specified
"05T09+12"	9:00, UTC +12 hours, on any day Example: 0000-01-05 09:00:00 +1200 +1200
"2018-01-16+12:00"	16 January 2018, UTC +12 hours Example: 2018-01-16 00:00:00 +1200 +1200
"01-16+06:30"	16 January of any year, UTC +6 hours and 30 minutes Example: 0000-01-16 00:00:00 +0630 +0630

