



SAS[®] Viya[™] 3.1 Administration: Logging

Logging Overview

The logs that are produced for SAS Viya depend on the components that are active in your environment. In addition, different components use different methods for determining the level of messages that are logged. Here are the SAS Viya components that produce logs and the methods used to control logging:

SAS Viya servers

SAS Viya servers use the SAS logging facility to generate and process log messages. Aspects of logging are defined in the logging configuration file (by default, `logconfig.xml`). Details about logging configuration include the level of messages that are written to the log, the message format, and the output destination for the log. You enable logging for a server by using the `logconfigloc` option to identify the logging configuration file to use. An example is `-logconfigloc=logconfig.xml`). For a CAS server, the logging configuration file is specified by the option `cas.logcfgloc = config.loc .. '/logconfig.xml'`. Specify the applicable option in the configuration file for the server in order to start logging whenever the server starts. See ([SAS 9.4 Logging: Configuration and Programming Reference](#)) for complete information about the SAS logging facility. The Note that the information that refers to SAS Management Console does not apply to a SAS Viya environment. The following SAS Viya servers create logs:

- CAS Server
- SAS Workspace Server
- SAS object spawner
- SAS/CONNECT server
- SAS/CONNECT spawner

Some SAS Viya applications such as SAS Event Stream Processing also use the SAS logging facility. Refer to the application's documentation for information about application logging.

SAS Configuration Server (Consul)

Logging for the SAS Configuration Server is controlled by a configuration file. However, the format of the configuration file and logging process is different from that used by other SAS Viya servers. The SAS Configuration Server is used only by SAS Visual Investigator.

SAS Visual Investigator

Messages from SAS Visual Investigator are aggregated by Cloud Foundry and can be processed by third-party log management services, which you must provide, install, and maintain.

Logs for all components other than SAS Visual Investigator are stored in the location `/opt/sas/deployment-name/config/var/log/product-name`.

How To

SAS Viya Server Logging

Enabling Logging for SAS Viya Servers

Specify the `-logconfigloc=config-file.xml` option in order to start logging for a SAS Viya server. The default name for `config-file.xml` is `logconfig.xml`. The `logconfigloc` option is specified by default for SAS Viya servers in the configuration files for the servers. If you need to change the logging configuration (in order to specify a different logging configuration file, for example), you must modify the usermods configuration file if one is present. Here are the locations and names of the files:

CAS Server

- Configuration directory: `/opt/sas/deployment-name/config/etc/cas/default/`
- Server configuration file: `casconfig.lua`

SAS Workspace Server

- Configuration directory: `/opt/sas/deployment-name/config/etc/workspaceserver/default/`
- Default configuration file: `sasv9.cfg`
- User modification configuration file: `sasv9_usermods.cfg`

SAS object spawner

- Configuration directory: `/opt/sas/deployment-name/config/etc/spawner/default`
- Default configuration file: `spawner.sh`
- User modification configuration file: `spawner_usermods.sh`

SAS/CONNECT server

- Configuration directory: `/opt/sas/deployment-name/config/etc/connectserver/default`
- Default configuration file: `sasv9.cfg`
- User modification configuration file: `sasv9_usermods.cfg`

SAS/CONNECT spawner

- Configuration directory: `/opt/sas/deployment-name/config/etc/connect/default`
- Default configuration file: `connect.sh`
- User modification configuration file: `connect_usermods.cfg`

By default, the SAS Workspace Server and the SAS/CONNECT server send log messages to the operating system, rather than to a log file. If you need to capture messages from these servers in a log file for troubleshooting, you can specify a TRACE-level logging configuration file. See [Specify TRACE-Level Logging for SAS Viya Servers on page 3](#) for the steps required.

You can also use the LOG system option to send messages to a log file from the SAS session running the server process. Specify `LOG file-specification`, where `file-specification` is the destination for the log file. If you do not specify the destination, the file is created in the current directory. The default name is `filename.log`, where `filename` is the name of your SAS job.

You can specify this option in a configuration file, on a SAS command, or as a `SASV9_OPTIONS` environment variable. Other system options affect system logging, such as `LOGPARM` and `MSGLEVEL`. See [“Using SAS System Options” in SAS Viya System Options: Reference](#) for information about setting system options.

Modify the Threshold Level for SAS Viya Servers

The logging configuration file for the servers that are used by SAS Viya control how messages from each server are logged. One of the configuration settings is the threshold for each logger. Messages at the threshold level or higher are processed by the logger. Messages that are lower than the threshold are ignored. For example, specifying a logging threshold of DEBUG means that only events with a level of debug or higher are saved to the log.

When you are trying to isolate the cause of a problem, it can be useful to temporarily set the logging threshold to a lower level. Lowering the threshold adds more detailed messages to the log, and can help you locate the actions that are causing a problem. However, you should change the level back to a higher level after fixing the problem so that your log does not fill up with unneeded messages.

To modify the logging level for a SAS Viya server logger, follow these steps:

- 1 Edit the logconfig.xml file for the server. Here are the locations of the files for the following servers:

CAS Server

```
/opt/sas/deployment-name/config/etc/cas/default
```

SAS Workspace Server

```
/opt/sas/deployment-name/config/etc/workspaceserver/default
```

SAS object spawner

```
/opt/sas/deployment-name/config/etc/spawner/default
```

SAS/CONNECT server

```
/opt/sas/deployment-name/config/etc/connectserver/default
```

SAS/CONNECT spawner

```
/opt/sas/deployment-name/config/etc/connect/default
```

- 2 Locate the logger whose value you want to change. Each logger processes the messages from a specified category. See [CAS Server Loggers on page 6](#) for information about the defined loggers for the CAS Server. See [SAS 9.4 Logging: Configuration and Programming Reference](#) for information about logger definitions for other servers.

The logger entries have the following format:

```
<logger name="Admin">
  <level value="Info" />
</logger>
```

- 3 Change the value of the level parameter to the level that you want to use. Valid values are TRACE, DEBUG, INFO, WARN, ERROR, and FATAL (ordered lowest to highest). See [Logging Thresholds on page 7](#) for details about the logging thresholds.
- 4 Save and close the file.

Specify TRACE-Level Logging for SAS Viya Servers

If you are troubleshooting a problem with a SAS Viya server, you might want to temporarily change the logging threshold level for the server to TRACE. This level is optimal for troubleshooting, but produces too many messages for normal use.

By default, logging for SAS Viya servers is specified in the logconfig.xml configuration file for each server type. However, you can point to the alternative configuration file logconfig.trace.xml while you are troubleshooting and return to the default file after you have finished. Pointing to the alternative file reduces the potential errors that might occur if you modify the default file.

To change the logging level, follow these steps:

4

- 1 Modify the server configuration file to point to the logconfig.trace.xml file. Here are the modifications to make for each server type:

CAS Server

Modify the file `/opt/sas/deployment-name/config/etc/cas/default/casconfig.lua`.

Add the line `cas.logconfig = config_loc .. '/logconfig.trace.xml'`

SAS Workspace Server

Modify the file `/opt/sas/deployment-name/config/etc/workspaceserver/default/sasv9_usermods.cfg`.

Add the line `-logconfigloc=/opt/sas/deployment-name/config/etc/workspaceserver/default/logconfig.trace.xml`.

SAS object spawner

Modify the file `/opt/sas/deployment-name/config/etc/spawner/default/spawner_usermods.sh`.

Add the line `spawner_options="-logconfigloc ${config_home}/logconfig.trace.xml"`.

SAS/CONNECT server

Modify the file `/opt/sas/deployment-name/config/etc/connectserver/default/connectserver_usermods.cfg`.

Add the line `-logconfigloc=/opt/sas/deployment-name/config/etc/connectserver/default/logconfig.trace.xml`.

SAS/CONNECT spawner

Modify the file `/opt/sas/deployment-name/config/etc/connect/default/connect_usermods.sh`.

Add the line `spawner_options="-logconfigloc ${config_home}/logconfig.trace.xml"`.

- 2 After you have finished troubleshooting the problem, modify the server configuration file and remove the line that you had added.

Locate the SAS Viya Server Logs

Each SAS Viya server produces a separate log file for each day.

The logs for each SAS Viya server are stored in the following locations:

CAS Server

`/opt/sas/deployment-name/config/var/log/cas/default`

SAS Workspace Server

`/opt/sas/deployment-name/config/var/log/workspaceserver/default`

SAS object spawner

`/opt/sas/deployment-name/config/var/log/spawner/default`

SAS/CONNECT server

`/opt/sas/deployment-name/config/etc/connectserver/default`

SAS/CONNECT spawner

`/opt/sas/deployment-name/config/etc/connect/default/.`

CAS has several types of server processes such as the main control node, a session control node, and the main worker nodes. Log messages from the session control nodes and the worker nodes are forwarded to their main processes. A separate log file is produced for the main control node and for each main worker node.

If you specify the parameter `-onelog` when you start CAS, all log messages are sent to the main control node. A single log file is produced for all of the CAS server processes.

Each entry in a log file for a CAS server displays the node name and the user ID from which the message originated.

Add a Logger for a SAS Viya Server

The default logging configuration file includes entries for several loggers, but other valid loggers can be added to the file. The addition of a logger enables you to specify how a specific type of log event is processed. For example, you can add a logger for a specific CAS action set. Also, you can set the threshold level to a low level so that you can isolate a problem with that action set.

To add a logger, follow these steps:

- 1 Edit the logconfig.xml file for the server.
- 2 Locate the existing logger entries in the file.
- 3 Add the new logger entry.

The logger entries have the following format:

```
<logger name="logger-name">
  <level value="threshold" />
</logger>
```

SAS Configuration Server Logging

Locate the Log Files for the SAS Configuration Server

The log files for the SAS Configuration Server are stored in the location `/opt/sas/deployment-name/config/var/log/consul`.

Note: The SAS Configuration Server (Consul) is used only by SAS Visual Investigator.

Modify the Logging Level for the SAS Configuration Server

The default logging level for the SAS Configuration Server is INFO. To change the level, follow these steps:

- 1 Modify the file `/opt/sas/deployment-name/config/etc/consul.d/config-consul.json`.
- 2 Change the line `"log_level": "INFO",`, replacing INFO with the log level that you want to use. Valid values are TRACE, DEBUG, INFO, WARN, and ERROR.

Note: The SAS Configuration Server (Consul) is used only by SAS Visual Investigator.

SAS Visual Investigator Logging

SAS Visual Investigator uses two systems for logging.

- Logs for the services cas, consul, elasticsearch, postgresql, pgpool, and rabbitmq are sent using TCP/IP as rsyslog messages to a remote rsyslog server. After you install and configure the rsyslog server, you specify the name and the fully qualified domain name (FQDN) of the server in the manifest file for the service.
- Logs for microservices that are used by SAS Visual Investigator are enabled to use Loggregator, which is a component of Cloud Foundry. In order to access these logs, you must install, configure, and deploy a Cloud Foundry Nozzle for the Loggregator system. For more information about using Loggregator and Nozzles, see the [Cloud Foundry documentation](#).

Reference

CAS Server Loggers

A logger identifies how different categories of messages are processed, including the level of messages that are processed and the appender to which messages are sent. For example, an Admin logger specifies how administration-related log messages are processed.

In the logging configuration file, a logger has this format:

```
<logger name="logger-name">
  <level value=threshold/>
  <appender-ref ref="appender-name"?>
</logger>
```

The `level value=threshold` parameter for a logger specifies the lowest level of messages that is processed by the logger. For example, a level of WARN specifies that only messages with a level of WARN, ERROR, and FATAL are included in the log.

The `appender-ref ref=appender-name` parameter for a logger specifies the appender, or the destination, for log messages.

See [SAS 9.4 Logging: Configuration and Programming Reference](#) for complete information about logger definitions.

The following loggers are defined with a default threshold of INFO in the default logging configuration file for the CAS server:

Admin

processes the administration events. The log messages are sent to the UNIX system log.

App

processes the events from applications.

App.cas.actions

processes the events from CAS actions.

Audit

processes the events that are used for auditing. These events include the user authentication requests and administration of access controls.

Logging

processes the events from the logging system. Log messages are sent to the UNIX system log.

The following loggers are not defined by default:

App.cas

processes the events from the CAS server.

App.cas.actions.action-setname.action-name

processes the events from a specified CAS action set and a CAS action.

App.cas.driver

processes the events from start-up of the CAS server.

App.cas.tkcasa

processes the events from internal processes.

App.cas.io.cache

processes the messages that are related to running out of space in the CAS_DISK_CACHE volumes.

App.cas.datastep

processes the output and the events from the DATA step PUT statements, as well as the messages that are sent to the SAS log.

CAS Server Appenders

The default configuration file for the CAS server defines the following appenders:

RollingFileAppender

writes log messages to a time-based rolling log file. By default, the file rolls over at midnight. Also, messages from the App logger, the App.cas.actions logger, and the Audit logger are sent to this appender.

UNXFacilityAppender

writes log messages to the syslogd logging facility in the UNIX operating system. It discards messages that have already been logged by the appender. By default, messages from the Admin and Logging loggers are sent to this appender.

Logging Thresholds

The logging configuration file sets a threshold for each logger. Messages at the threshold level or higher are processed by the logger. Messages that are lower than the threshold level are ignored. The following threshold levels (ordered lowest to highest) are available:

Trace

produces the most detailed log messages. This level might be useful when isolating the cause of a problem, but it produces too many messages for normal use.

Debug

produces detailed log messages, but are less detailed than the Trace threshold. This level might be useful when isolating the cause of a problem, but it produces too many messages for normal use.

Info

produces messages that show an application's progress.

Warn

produces messages that identify areas of potential problems.

Error

produces messages when errors occur, but the application might continue to run.

Fatal

produces messages when severe errors occur. The application probably ends.

