SAS® Viya™ 3.1
Administration: SAS® Cloud Analytic Services 3.1

Servers: Overview

SAS Viya contains these servers:
- SAS Cloud Analytic Services
- SAS Workspace Server and Object Spawner
- SAS/Connect Server and Spawner
- embedded web application server
Cloud Analytic Services: How To (CAS Server Monitor)

Start and Stop Cloud Analytic Services

You can manage and view the running state of the CAS server using the `sas-viya-cascontroller-default` service on the same host as the controller.

**TIP** Your site’s Linux administrator might want to create a regular account (for example, sas-service-admin) and give that account the sudo permissions to manage the SAS services. Make sure that the services are defined as “start on reboot” so that the CAS server automatically starts when the machine is rebooted.

Start:

```
sudo service sas-viya-cascontroller-default start
```

Stop:

```
sudo service sas-viya-cascontroller-default stop
```

Restart:

```
sudo service sas-viya-cascontroller-default restart
```
Obtain status:

```
sudo service sas-viya-cascontroller-default status
```

**Note:** For Red Hat 7, use `systemctl` instead of `service` in the preceding list: `sudo systemctl status | start | restart | stop service`.

---

## Starting, Stopping, and Checking the Status of Services

You run the `sas-viya-all-services` script to start, stop, or check the status of all the services in your SAS Viya environment.

**Note:** To run this script, you must have `sudo` privileges.

- Check the status of all services:
  ```
  command: sudo service sas-viya-all-services status
  ```
- Stop all services:
  ```
  command: sudo service sas-viya-all-services stop
  ```
- Start all services:
  ```
  command: sudo service sas-viya-all-services start
  ```

---

## View CAS Controller and System Information

1. **Sign in** to CAS Server Monitor with a valid user ID and password.
2. In CAS Server Monitor, beneath the **Cloud Analytic Services** banner, click 📈.
3. On the System State page, make sure that **Controller** is selected.

---

## View CAS Server Configuration

To use CAS Server Monitor to view the current list of **CAS Server options** and their values, follow these steps:

1. **Sign in** to CAS Server Monitor with a valid user ID and password.
2. In CAS Server Monitor, beneath the **Cloud Analytic Services** banner, click 📈.
3. On the Configuration page, make sure that **CAS Configuration** is selected.

---

## View CAS Start-up Options and Environment Variables

You can use CAS Server Monitor to view the option used when a CAS Server was started and to see the current list of CAS environment variables and their values.

To view CAS start-up options and environment variable values, follow these steps:

1. **Sign in** to CAS Server Monitor with a valid user ID and password.
2. In CAS Server Monitor, beneath the **Cloud Analytic Services** banner, click 📈.
3. On the System State page, select **Runtime Environment**.
Manage CAS Nodes

You can use CAS Server Monitor to view, add, and remove CAS nodes in your analytics cluster.

Note: In order to add CAS nodes, the requisite software must already have been deployed on the machines that you are adding. To add new machines, deploy SAS on them first, and then you can add them using the CAS Server Monitor.

To manage a node, follow these steps:

1. Sign in to CAS Server Monitor with a user ID that has CAS Administrator privileges.

2. In CAS Server Monitor, beneath the Cloud Analytic Services banner, click .


4. From the Nodes table, you can:
   - View information about all the nodes in your analytics cluster.
   - Add nodes to your analytics cluster:
     - Click Add Nodes.
     - On the Add Nodes dialog box, in Hostname, enter a simple host name, such as mygrid011 and click OK.
       Separate multiple host names with a comma.

       If your hosts are named in numeric order (for example, host002, host003, ...) you can enter a range of host names. Use the form, host[start-number-end-number] (for example, mygrid[002–030]).

   - Shut down nodes on your analytics cluster:
     - Next to the node that you want to shut down, click ✗ and select Terminate Server Instance.

   - Remove nodes from your analytics cluster:
     - Next to the node that you want to remove, click ✗ and select Remove Nodes.

View User Session Information

You can use CAS Server Monitor to view information about a user’s session such as connection port, length of connection time, and so on.

To view user session information, follow these steps:

1. Sign in to CAS Server Monitor with a valid user ID and password.

2. In CAS Server Monitor, beneath the Cloud Analytic Services banner, click .


Cancel CAS User Session

To cancel your CAS Server session, follow these steps:

1. Sign in to CAS Server Monitor with a valid user ID and password.

2. In CAS Server Monitor, beneath the Cloud Analytic Services banner, click .
3 On the System State page, select User Sessions.

4 At the end of the row for the session that you want to cancel, click  and select Cancel Session.

### Terminate CAS User Session

**TIP** Terminate a session only after having tried canceling a session. Using terminate might not release resources (for example, mapped memory and memory involving database connections, and so on).

To terminate your CAS Server session, follow these steps:

1 **Sign in** to CAS Server Monitor with a valid user ID and password.

2 In CAS Server Monitor, beneath the Cloud Analytic Services banner, click  

3 On the System State page, select User Sessions.

4 At the end of the row for the session that you want to terminate, click  and select Terminate Session.

### Adjust Caslib Management Privileges

To enable non-administrators to add global caslibs:

1 **Sign in** to CAS Server Monitor with a valid user ID and password that has administrator privileges.

2 In CAS Server Monitor, beneath the Cloud Analytic Services banner, click  

3 On the Configuration page, select Access Controls.

4 In the Caslibs list, select Global Caslib Creation.

**TIP** If the Global Caslib Creation caslib is not listed, you are not signed in as an administrator.

5 In the upper right, click Edit.

6 In the Edit Access Controls window, adjust values as needed.

<table>
<thead>
<tr>
<th>Intent</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable all users to add global caslibs.</td>
<td>In the existing row for Authenticated Users, select the Grant radio button.</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable a group to add global caslibs.</td>
<td>Click Add Row. Select Group, enter the group name, and select the Grant radio button.</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable an individual user to add global caslibs.</td>
<td>Click Add Row. Select User, enter the user name, and select the Grant radio button.</td>
</tr>
</tbody>
</table>

7 Click OK to save your changes.

8 Under Access Controls, review the results of your changes.

9 Verify that users who should be able to add global caslibs can do so.
Here are details:

- User and group names that you enter are not validated.
- Regardless of access controls, administrators can add and manage all caslibs.
- For the special caslibs (Global Caslib Creation and Session Caslib Creation), the only available value in the Activity column is Manage Access. The special caslibs are protected by role requirements, not by the ManageAccess permission. Granting or denying the ManageAccess permission on the special caslibs affects only the ability of non-administrators to manage other caslibs.
- If you want to restrict the ability to manage session caslibs, select Session Caslib Creation in the Caslibs list. Add direct denials as needed.

Cloud Analytic Services: Concepts

CAS Controller

Controller is one of two roles that can be assigned to a host for SAS Cloud Analytic Services. For both server architectures, distributed and single-machine, one machine is assigned the controller role. When the server starts, the controller process is started. This process is sometimes referred to as the server controller. The controller accepts a connection from a client.

Single-machine CAS Server

The single-machine architecture uses symmetric multiprocessing (SMP). The functionality for a single-machine server is nearly identical to MPP, except that there is no cluster communication. In this architecture, the server acts as a controller. Before a client connects, the server listens on a port for connections.

After a client connects, a session is created and the session connects back to the client. (This is identical to the method that is performed by a CAS server that uses MPP.)
**CAS Workers**

When a server is running in massively parallel processing (MPP) mode, in addition to a controller, the server also has multiple machines that are assigned the worker role.

The controller parses out work to each worker node. Each worker node sends the results of its computations back to the controller.

**Distributed CAS Server**

CAS can be co-located with Hadoop on a cluster of machines. This massively parallel processing (MPP) architecture is appropriate for analyzing large data sets. Analysis proceeds on tables that are already made available to the server (loaded) or on tables that are gathered or created by the server on demand.

**Session Processes**

When a user connects to the server with a client, the server starts a session process for the user. Afterward, the client communicates with the session process.
A server running in symmetric multiprocessing mode (SMP mode) consists of a controller only, and the server starts a session controller process only. It is the session controller process that operates on rows of data.

In a distributed server (MPP mode), a session process is created on each machine in the cluster. These processes are sometimes referred to as the session controller and session worker processes.

Even though the sessions have their own operating system processes, the server processes must continue to run. When the server process terminates, the session processes also terminate.

**Paths List**

From a CAS server, all access to file system paths (host, S3, and HDFS directories) is through caslibs. To limit the paths that are available to non-administrators when they created or edit a caslib, use one of the following approaches:

- Create a blacklist of paths that should not be available.
- Create a whitelist of paths that should be available.

Here are key points:

- Paths must be absolute.
- All subdirectories of each specified path are affected.
- Paths list constraints do not affect access to existing caslibs.
- If you do not define a blacklist or whitelist, no paths list constraints are in effect.
- Paths list constraints do not apply to users who assume the Superuser role or the Data role.
- Only users who assume the Superuser role for a server can see and manage that server’s paths list.

**Note:** Access to third-party databases is not affected by a server’s blacklist or whitelist.

**Caslib Management Privileges**

**Caslib Management Privileges**

<table>
<thead>
<tr>
<th>Task</th>
<th>Who Can Perform the Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add global casibs.</td>
<td>Superusers and Data administrators.</td>
</tr>
<tr>
<td></td>
<td>Users who have global caslib management privileges.</td>
</tr>
<tr>
<td>Add session casibs.</td>
<td>Superusers and Data administrators.</td>
</tr>
<tr>
<td></td>
<td>Users who have session caslib management privileges.</td>
</tr>
<tr>
<td>Delete global casibs.</td>
<td>Superusers and Data administrators.</td>
</tr>
<tr>
<td></td>
<td>Users who have global caslib management privileges can delete any global caslib for which they have the ReadInfo and ManageAccess permissions.</td>
</tr>
<tr>
<td>Delete session casibs.</td>
<td>Superusers and Data administrators.</td>
</tr>
<tr>
<td></td>
<td>Users who have session caslib management privileges can delete any session caslib for which they have the ReadInfo and ManageAccess permissions.</td>
</tr>
<tr>
<td>Adjust caslib management privileges.</td>
<td>Superusers and Data administrators.</td>
</tr>
</tbody>
</table>

*Global caslib management privileges correspond to the ManageAccess permission on the _GLOBAL caslib. Session caslib management privileges correspond to the ManageAccess permission on the _SESSION caslib.*
Cloud Analytic Services: Reference

Configuration File Options

How Do I Use CAS Configuration File Options?
SAS Cloud Analytic Services options are stored in a configuration file. During deployment, a server configuration file is created in /opt/sas/viya/config/etc/cas/default/casconfig.lua. When you start the server with the service sas-viya-cascontroller-default start command, the options are read.

For the order of precedence for server configuration options, see "How the Session Option Values Are Determined" in SAS Cloud Analytic Services: Language Reference.

TIP Remember that you can also set operating system environment variables in casconfig.lua. For example, env.HADOOP_HOME='/hadoop/hadoop-someversion' env.HADOOP_NAMENODE='name_node.example.com'.

Configuration File Options Reference

**cas.actionsetlist**
Reserved for internal use.

**cas.addfmtlib='table-name.sashdat <table-name.sashdat <...>>'**
Specifies the format libraries to add and promote during server start-up prior to any sessions starting.

table-name.sashdat is a persisted format library that was previously saved using the CAS mysess SAVEFMTLIB statement. Use the CAS statement SAVEFMTLIB to persist a format library to a caslib or a table.

Begin and end the string of one or more table names with single quotation marks. Separate multiple library names with a whitespace character.

Valid in CAS configuration file

Category Formats

Note During session start up each format library identified by cas.addfmtlib, is automatically added to the format search for the session.

See cas.fmtcaslib on page 13

Example cas.addfmtlib='mylib1.sashdat mylib2.sashdat mylib3.sashdat'

**cas.allowuser**
Reserved for internal use.

**cas.apptag='tag-string'**
Specifies an arbitrary string to prefix to log messages.

Using apptag helps determine which log messages are associated with an application.

Valid in CAS statement SESSOPTS option

CAS configuration file
The CAS server uses `apptag` when writing to its log.

See `cas.logcfgloc`

**Example**

`apptag='my_app'`

---

**cas.caslib**

Reserved for internal use.

**cas.cmopt=\`optimization-value <optimization-value <...>>\` | 'all' | 'none'**

Specifies the type of code generation optimizations to use in the SAS language compiler.

- **optimization-value**
  Specifies the type of optimization that the SAS compiler is to use. Specify one or more of the following as a space-delimited list enclosed in quotation marks:

  - 'extramath' | 'noextramath'
    Specifies whether the compiler is to retain or remove the extra mathematical operations that do not affect the outcome of a statement.

  - 'funcdifferencing' | 'nofuncdifferencing'
    Specify `funcdifferencing` to calculate numeric-differencing derivatives for user-defined functions. Specify `nofuncdifferencing` to calculate analytic derivatives for user-defined functions.

  - 'guardcheck' | 'noguardcheck'
    Specifies whether the compiler checks for array boundary problems.

  - 'misscheck' | 'nomisscheck'
    Specifies whether to check for missing values in the data.

  - 'precise' | 'noprecise'
    Specify `precise` to handle exceptions at the operation boundary. Specify `noprecise` to handle exceptions at the statement boundary.

- **'all'**
  Specifies that the compiler is to optimize the machine language code by using the `noextramath`, `nomisscheck`, `noprecise`, `noguardcheck`, and `nofuncdifferencing` optimization values.

  Note: 'all' cannot be specified with other values.

- **'none'**
  Specifies that the compiler is not set to optimize the machine language code by using the `extramath`, `misscheck`, `precise`, `noguardcheck`, and `funcdifferencing` optimization values.

  Note: 'none' cannot be specified with other values.

---

**Valid in**

- CAS statement SESSOPTS option
- CAS configuration file
Note If the data contains a significant amount of missing data, specify `misscheck` to optimize the compilation. Otherwise, specify `nomisscheck`.

Example In this example, the SAS compiler is set to retain the extra mathematical operations, check for missing values, and handle exceptions at an operation boundary:

```plaintext
cas.cmpopt='extramath misscheck precise'
```

**cas.collate='mva' | 'uca'**

Specifies the collating sequence for sorting.

- **mva** specifies SAS client collating.
- **uca** specifies a locale-appropriate collating sequence.

**Valid in**
- CAS statement `SESSOPTS` option
- CAS configuration file

**Category** Sort

**Default** 'uca'

**Example**

```plaintext
cas.collate='mva'
```

**cas.colocation='none' | 'hdfs'**

Specifies whether to create a personal caslib (`hdfs`) at CAS server start-up.

A server started in MPP mode defaults to `hdfs` because it assumes it is co-located with Hadoop. Specify `none` for the server running in MPP mode not to create a personal caslib at start-up.

**Valid in** CAS configuration file

**Category** Caslib

**Default** `cas.colocation='hdfs'`

**Requirement** Used with `cas.mode='mpp'` and `cas.hdfsuserloc`.

**Example** In this example, the CAS server is running in MPP mode and is not co-located with Hadoop. At start-up, the CAS server does not create a personal caslib for the user ID under which the server is run.

```plaintext
cas.colocation='none'
```

**cas.createuserloc**

Reserved for internal use.

**cas.datastepmsgsumlevel='all' | 'none' | 'put'**

Specify the DATA step message summary level. When the DATA step runs on multiple threads, the same message can be generated on each thread. This option controls the summary level of duplicate messages.

- **'all'**
  
The first occurrence of all message and put statements are sent to the client when they occur. Duplicate occurrences of all message and put statements are summarized and sent to the client when the DATA step exits. This is the default.

- **'none'**
  
  All message and put statements from every thread are written to the client log. No summarization occurs.

- **'put'**
The first occurrence of all message and put statements are sent to the client. Duplicate occurrences of messages are summarized and sent to the client when the DATA step exits. put statements are not summarized rather sent to the client when they occur.

Valid in CAS statement SESSOPTS option
CAS configuration file
Category DATA Step
Default All
Example In this example, all message and put statements from every thread are written to the client log. No summarization occurs.
cas.datastepmsgsumlevel='none'

cas.datastepreplacetable=true | false
Specifies whether a DATA step can replace an existing table.

Valid in CAS statement SESSOPTS option
CAS configuration file
Category DATA Step
Default true
Note The values true and false are case-sensitive.
Example cas.datastepreplacetable=true

cas.display
Reserved for internal use.

cas.elastic=true | false
Indicates that new machines are allowed to join the analytics cluster.

Valid in CAS configuration file
Category Administration
Default false
Requirement Used with cas.gcport.
Supports CAS servers running MPP.
Note The values true and false are case-sensitive.
See cas.gcport
Example In this example, the CAS controller allows new worker nodes to join the analytic cluster:
cas.elastic=true

cas.elasticssl=true | false
When elasticssl=true, new machines are allowed to join the cluster when the CAS controller can authenticate their identity.

Authentication data is contained in a key file that was used to start the CAS server. Any machine that can access this key can join the CAS controller as a worker node.
Valid in: CAS configuration file  
Category: Security  
Default: false  
Requirement: Used with `cas.gcport` and `cas.keyfile`.  
Supports: CAS servers running MPP.  
Note: The values `true` and `false` are case-sensitive.  
See: `cas.gcport` and `cas.keyfile`.  
Example: `cas.elasticssl=true`

**cas.fmtcaslib='caslib'**  
Specifies the caslib where persisted format libraries can be found.  
Enclose `caslib` in single quotation marks.  
Valid in: CAS statement SESSOPTS option  
Category: Formats  
Default: `'formats'`  
Note: Set by the system administrator.  
See: `cas.addfmtlib` on page 9  
Example: `cas.fmtcaslib='formats'`

**cas.gcport=port**  
Specifies the network port that is used on a distributed server for communication between the controller and its worker nodes.  
The commonly configured port is 5580.  
Valid in: CAS configuration file  
Category: Network  
Default: 0 (random port in the range 32678–61000)  
Supports: CAS servers running MPP.  
See: `cas.httpport` and `cas.port`  
Example: `cas.gcport=5580`

**cas.hdfsuserloc='hdfs-path/%USER'**  
For CAS servers running in MPP mode, specifies that the server create a personal caslib for each user at session start-up time in the specified HDFS path.  
`'hdfs-path/%USER'` refers to a directory named for the user’s user ID under the specified HDFS path.  
Enclose `hdfs-path` in single quotation marks.
### Valid in
CAS configuration file

### Category
Data

### Requirement
`cas.mode='mpp' on page 18` and `cas.colocation='hdfs' on page 11`

### Example
In this example, the user’s caslib directory is a subdirectory named for the user ID under `/user`:

```
    cas.hdfsuserloc='/user/%USER'
```

#### `cas.httpport=port | port-range`

The port (or range of ports) that SAS Cloud Analytic Services listens to for HTTP communication.

The commonly configured port is 8777.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Network</td>
</tr>
<tr>
<td>Default</td>
<td>0 (random port)</td>
</tr>
<tr>
<td>Note</td>
<td>If the first port in the range is already taken, CAS tries the next port until it finds a port that is free.</td>
</tr>
<tr>
<td>See</td>
<td><code>cas.httpportmax</code>, <code>cas.gcport</code>, and <code>cas.port</code></td>
</tr>
<tr>
<td>Examples</td>
<td><code>cas.httpport=8777</code></td>
</tr>
<tr>
<td></td>
<td><code>cas.httpport=8777-9000</code></td>
</tr>
</tbody>
</table>

#### `cas.httpportmax=maximum-port-range`

Specifies the maximum port range that SAS Cloud Analytic Services listens to for HTTP communication.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Network</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
<tr>
<td>Range</td>
<td>0–65535</td>
</tr>
<tr>
<td>See</td>
<td><code>cas.httpport</code></td>
</tr>
<tr>
<td>Example</td>
<td><code>cas.httpport=8777-9000</code></td>
</tr>
</tbody>
</table>

#### `cas.ignorereadtimeout`

Reserved for internal use.

#### `cas.invaliddata`

Reserved for internal use.

#### `cas.jitdb`

Reserved for internal use.

#### `cas.jitex`

Reserved for internal use.

#### `cas.jreoptions=('JRE-option <JRE-option> <...>)'`

Specifies the Java Virtual Machine (JVM) options that SAS Cloud Analytic Services uses at start-up. Separate JRE options with a whitespace character. Enclose any paths in quotation marks.
For the list of the valid Java options, and what they do, see [http://docs.oracle.com/javase/6/docs/technotes/tools/windows/java.html](http://docs.oracle.com/javase/6/docs/technotes/tools/windows/java.html)

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Java</td>
</tr>
<tr>
<td>Default</td>
<td>(null)</td>
</tr>
</tbody>
</table>
| Example  | In the following example, the initial and maximum sizes of the memory allocation pool are set to 256 and 1024MB, respectively. Also, the log4j configuration file path and Java classpath are set: cas.jreoptions = '(-Xms256m -Xmx1024m
-Dlog4j.configuration=' .. ' -Djava.class.path=' .. env.CAS_HOME .. '/lib/base-base-tkjni.jar')' |

**cas.keyfile='pathname'**

Identifies to the CAS controller the path and filename to the X.509 digital certificate file that is used to start the server. The certificate must be signed by a CA that is trusted by the CAS server.

Enclose *pathname* in single quotation marks.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Security</td>
</tr>
<tr>
<td>Requirement</td>
<td>Used with cas.elasticssl and cas.mode='mpp'.</td>
</tr>
<tr>
<td>Supports</td>
<td>CAS servers running MPP.</td>
</tr>
<tr>
<td>See</td>
<td>cas.gcport and cas.elastic</td>
</tr>
<tr>
<td>Example</td>
<td>cas.keyfile='/opt/TKGrid/certs/controller.pem'</td>
</tr>
</tbody>
</table>

**cas.lifetime=minutes**

Indicates the duration, in minutes, that a server remains running.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Administration</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
<tr>
<td>Example</td>
<td>In the following example, the server shuts itself down in 120 minutes: cas.lifetime=120</td>
</tr>
</tbody>
</table>

**cas.locale='POSIX-locale-string'**

Specifies the locale to use for sorting and formatting.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS statement SESSOPTS option, CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Localization</td>
</tr>
<tr>
<td>Default</td>
<td>'en_US'</td>
</tr>
<tr>
<td>See</td>
<td>valid POSIX locale strings</td>
</tr>
<tr>
<td>Example</td>
<td>cas.locale='fr_FR'</td>
</tr>
</tbody>
</table>
**cas.logcfgloc='pathname'**

Specifies the path to the SAS logging facility logging configuration file.

Enclose *pathname* in single quotation marks.

Valid in: CAS configuration file

Category: Log

See: Logging and *cas.apptag* on page 9

Example: `cas.logcfgloc='/opt/sas/cas1/etc/logconfig.xml'`

**cas.logflushtime=-1 | 0 | number**

Specifies the log flush time, in milliseconds.

- `-1`
  
  flushes logs after each action completes.

- `0`
  
  flushes logs as they are produced.

- `number`
  
  flushes logs in *number* milliseconds.

Valid in: CAS statement SESSOPTS option, CAS configuration file

Category: Log

Default: 100

Range: -1–86400

Example: In the following example, the CAS server writes buffered lines to the log every 500 milliseconds: `cas.logflushtime=500`

**cas.machinelist='path/machine-list-file'**

Identifies the path and filename on the controller machine that contains the list of machines in the CAS analytics cluster.

Enclose *path* in single quotation marks.

*machine-list-file* contains all of the machines in the analytics cluster in the form:

<fully-qualified-domain-name controller | worker>

Place each machine on a separate line. For example:

```
my_machine01.example.com controller
my_machine02.example.com worker
my_machine03.example.com worker
my_machine04.example.com worker
my_machine05.example.com worker
```

Valid in: CAS configuration file

Category: Administration

Requirement: Used with *cas.mode='mpp'*. 
Interaction  
Do not specify `cas.mode='smp'` when a valid machine list file is used.

Example  
cas.machinelist='/etc/my_machine_list'

**cas.maxtablemem=number | '[number k | m | g | t]'**

Specifies the maximum amount of physical memory to allocate for a table.

- **`number`**  
specifies the maximum amount of physical memory, in bytes, to allocate for a table.

- **`'[number k | m | g | t]'`**  
specifies the maximum amount of physical memory to allocate for a table in a unit other than bytes: k (kilobytes), m (megabytes), g (gigabytes), and t (terabytes).

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS statement SESSOPTS option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Caslib</td>
</tr>
<tr>
<td>Default</td>
<td>16M</td>
</tr>
<tr>
<td>Note</td>
<td>After this threshold is reached, the server uses temporary files and operating system facilities for memory management.</td>
</tr>
<tr>
<td>Example</td>
<td>In this example, the CAS server can allocate up to 32MB of physical memory for a table: cas.maxtablemem='32m'</td>
</tr>
</tbody>
</table>

**cas.memorysize=number | '[number k | m | g | t]'**

Specifies the maximum amount of physical memory, to allocate for the CAS CGroup. This limit also applies to the YARN request, when `cas.useyarn` is specified.

- **`number`**  
specifies the maximum amount of physical memory, in bytes, to allocate for the CAS CGroup and (optionally) the YARN request.

- **`'[number k | m | g | t]'`**  
specifies the maximum amount of physical memory to allocate for the CAS CGroup and (optionally) the YARN request in a unit other than bytes: k (kilobytes), m (megabytes), g (gigabytes), and t (terabytes).

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS statement SESSOPTS option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Administration</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
<tr>
<td>See</td>
<td><code>cas.useyarn on page 22</code></td>
</tr>
<tr>
<td>Example</td>
<td>In the following example, the maximum amount of physical memory allocated for the CAS CGroup and the YARN request is 256GB: cas.memorysize='256g'</td>
</tr>
</tbody>
</table>
**cas.messagelevel='all' | 'default' | 'error' | 'none' | 'note' | 'warning'**

 Specifies the log message level.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS statement SESSOPTS option</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CAS configuration file</td>
</tr>
<tr>
<td>Category</td>
<td>Log</td>
</tr>
<tr>
<td>Default</td>
<td>'all'</td>
</tr>
<tr>
<td>Example</td>
<td>cas.messagelevel='default'</td>
</tr>
</tbody>
</table>

**cas.metrics=true | false**

Causes CAS server metrics information to be displayed (true) or not displayed (false) in the SAS log.

When cas.metrics=true, you see information similar to the following displayed in the SAS log:

```
NOTE: Action 'nobs' used (Total process time):
NOTE:       real time               2.100185 seconds
NOTE:       cpu time                0.010999 seconds (0.52%)
NOTE:       total nodes             6 (192 cores)
NOTE:       total memory            1.11T
NOTE:       memory                  7.00K (0.00%)
```

The analytic server processed the request in 2.100185 seconds.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS statement SESSOPTS option</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CAS configuration file</td>
</tr>
<tr>
<td>Category</td>
<td>Log</td>
</tr>
<tr>
<td>Default</td>
<td>false</td>
</tr>
<tr>
<td>Note</td>
<td>The values true and false are case-sensitive.</td>
</tr>
<tr>
<td>See</td>
<td>CASLIB Statement</td>
</tr>
<tr>
<td>Example</td>
<td>In the following example, CAS server metrics information is not displayed in the SAS log: cas.metrics=false</td>
</tr>
</tbody>
</table>

**cas.mode='smp' | 'mpp'**

Forces a server to be started in symmetric multiprocessing mode (smp) or in massively parallel processing mode (mpp).

<table>
<thead>
<tr>
<th>Category</th>
<th>Administration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction</td>
<td>Used in mpp mode with cas.machinelist.</td>
</tr>
<tr>
<td>Note</td>
<td>The server returns an error when cas.mode='smp' is specified for a server with a valid machine list.</td>
</tr>
<tr>
<td>Example</td>
<td>In the following example, the CAS server is forced to start in massively parallel processing mode (MPP). cas.mode='mpp'</td>
</tr>
</tbody>
</table>

**cas.nworkers=number**

Specifies the number of worker nodes associated with this session.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Valid in</th>
<th>Category</th>
<th>Default</th>
<th>Range</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cas.nworkers=8</strong></td>
<td><strong>CAS statement</strong> SESSOPTS option</td>
<td><strong>CAS configuration file</strong></td>
<td><strong>0</strong></td>
<td><strong>0–5000</strong></td>
<td><strong>cas.nworkers=8</strong></td>
</tr>
<tr>
<td><strong>cas.oauthsigningkey=’key-string’</strong></td>
<td><strong>CAS configuration file</strong></td>
<td><strong>Security</strong></td>
<td><strong>0</strong></td>
<td><strong>0–5000</strong></td>
<td><strong>cas.oauthsigningkey=’zAcSGqF23Pu85e7q7ZN2U42RhfV3WpwPAoE327k8w&amp;SSiodoUaIvY81tyTt5jKr4J50vUPWVHaR7YPi5</strong></td>
</tr>
<tr>
<td><strong>cas.oauthsigningcertificate=’path/public-key-file.pem’</strong></td>
<td><strong>CAS configuration file</strong></td>
<td><strong>Security</strong></td>
<td><strong>false</strong></td>
<td><strong>false</strong></td>
<td><strong>cas.oauthsigningcertificate=’/opt/keys/mypublickey.pem’</strong></td>
</tr>
<tr>
<td>**cas.onelog=true</td>
<td>false**</td>
<td><strong>CAS configuration file</strong></td>
<td><strong>Log</strong></td>
<td><strong>false</strong></td>
<td><strong>cas.onelog=true</strong> is used with cas.logcfgloc and cas.loghost.**</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>The values true and false are case-sensitive.</strong></td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>In this example, each CAS worker node creates a log file:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>cas.onelog=false</strong></td>
</tr>
<tr>
<td><strong>cas.perms=’pathname’</strong></td>
<td><strong>CAS configuration file</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>Specifies the path and name of the CAS permissions file (an XML file).</strong></td>
</tr>
<tr>
<td><strong>Enclose pathname in single quotation marks.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The XML bootstrap file is created for you during deployment and is used only once. Do not edit the XML bootstrap file that SAS provides.</td>
</tr>
</tbody>
</table>

### Variables

- **cas.oauthsigningkey**
  - **Description:** Contains the text of the key used to sign OAuth tokens.
  - **Valid in:** CAS configuration file
  - **Category:** Security
  - **Example:**
    ```
    cas.oauthsigningkey='zAcSGqF23Pu85e7q7ZN2U42RhfV3WpwPAoE327k8w&SSiodoUaIvY81tyTt5jKr4J50vUPWVHaR7YPi5'
    ```

- **cas.oauthsigningcertificate**
  - **Description:** Contains the full path to the PEM file that contains the public key. The file must exist and be readable by the CAS process or CAS does not start.
  - **Valid in:** CAS configuration file
  - **Category:** Security
  - **Example:**
    ```
    cas.oauthsigningcertificate='/opt/keys/mypublickey.pem'
    ```

- **cas.onelog**
  - **Description:** Specifies that all server logging is written to a single file. When **cas.onelog=false**, each worker node creates its own log file.
  - **Valid in:** CAS configuration file
  - **Category:** Log
  - **Default:** false
  - **Interaction:** **cas.onelog=true** is used with cas.logcfgloc and cas.loghost.
  - **Note:** The values **true** and **false** are case-sensitive.
  - **Example:** In this example, each CAS worker node creates a log file:
    ```
    cas.onelog=false
    ```

- **cas.perms**
  - **Description:** Specifies the path and name of the CAS permissions file (an XML file).
  - **Valid in:** CAS configuration file
  - **Enclose pathname in single quotation marks.**
  - **Note:** The XML bootstrap file is created for you during deployment and is used only once. Do not edit the XML bootstrap file that SAS provides.
### cas.permstore='path'

Specifies the path to a directory where the CAS server stores permissions.

Enclose `path` in single quotation marks.

The server saves its caslib and access control information to the `cas.permstore` directory periodically and when it shuts down.

Each subsequent time the server starts, caslib and access control information is initialized from the server’s `cas.permstore` location.

**CAUTION! Backups of access controls are not automatically performed.** It is strongly recommended that you periodically back up each CAS server’s stored access control and caslib information. In particular, it is important to create a backup after you modify access controls or add, delete, or modify global caslibs. See *SAS Viya Administration: Backup and Recovery*.

**Valid in** CAS configuration file

<table>
<thead>
<tr>
<th>Category</th>
<th>Access Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note</td>
<td>Each CAS server should have its own <code>cas.permstore</code> location. To minimize the potential for network timing issues, it is recommended that each <code>cas.permstore</code> location be on the controller machine and not be on a network file system. The server creates a directory with the name of the fully qualified DNS name of the machine that the main controller is running on in the specified <code>permstore</code> directory. Do not directly edit the files in a <code>cas.permstore</code> location.</td>
</tr>
<tr>
<td>See</td>
<td><code>cas.permstore</code> on page 20</td>
</tr>
<tr>
<td>Example</td>
<td><code>cas.permstore='/opt/sas/viya/config/etc/cas/default/permstore'</code></td>
</tr>
</tbody>
</table>

### cas.pincode

Reserved for internal use.

### cas.port=port

Specifies the port to which the CAS server listens.

The maximum allowable port number is 65535. If you do not specify a valid port, then the server listens on a random port in the range 32678-61000.

The commonly configured port is 5570.

**Valid in** CAS configuration file

**Category** Network

<table>
<thead>
<tr>
<th>See</th>
<th><code>cas.gcport</code> and <code>cas.httpport</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td><code>cas.port=5570</code></td>
</tr>
</tbody>
</table>

### cas.precommand

Reserved for internal use.
cas.provlist='ext' | 'oauth'

Specify the authentication providers that the CAS Server uses to authenticate incoming user connections.

- **'ext'**
  
  The external provider provides support for an external PAM authentication method when root access is required for authentication.

- **'oauth'**
  
  OAuth provider is always loaded (even when not listed) to support REST endpoints and communications between CAS worker nodes and the controller.

Valid in: CAS configuration file

Category: Security

Default: oauth

Note: The CAS Server configures the specified providers and uses each in order until an authenticated connection is successful.

Example: In this example, an external provider provides support for an external PAM authentication method. Although not specified, OAuth is always loaded to support REST endpoints and communications between CAS worker nodes and the controller.

cas.provlist='ext'

cas.servicesbaseurl

Reserved for internal use.

cas.singleuser

Reserved for internal use.

cas.timeout=seconds

Specifies the SAS Cloud Analytic Services session time-out in seconds for a new or existing session.

Valid in: CAS statement SESSOPTS option

Category: Session

Default: In order of descending precedence:
1. CAS statement TIMEOUT= option value, if specified
2. SAS system option CASTIMEOUT=, if you explicitly set it in SAS to a value greater than 0
3. 60

Range: 0–31536000

Notes: The session time-out starts when the number of connections to the session becomes zero and no actions are executing.

If a connection is established before the time-out expires, the time-out is canceled. Otherwise, the session is automatically terminated when the time-out expires.

When set to 0, the session is terminated immediately when the connection count becomes zero.

See: “CASTIMEOUT= System Option” in SAS Cloud Analytic Services: Language Reference

Example: cas.timeout=100
**cas.timezone=offset**
Specifies the time zone offset, in hours, from UTC.

- **Valid in:** CAS statement SESSOPTS option
- **CAS configuration file**
- **Category:** Session
- **Default:** -1
- **Range:** -1–23
- **Example:** `cas.timezone=5`

**cas.useinstall**
Reserved for internal use.

**cas.userloc='%HOME' | 'pathname/%USER'**
Specifies that the CAS server create a personal caslib for each user at session start-up time in the specified location.

- `%HOME` equates to the user's operating system $HOME directory.
- `'pathname/%USER'` refers to a directory named for the user's user ID under the specified file system path.

Enclose `pathname` in single quotation marks.

- **Valid in:** CAS configuration file
- **Category:** Caslib
- **Examples:**
  - In this example, the personal caslib directory is the user's operating system $HOME directory: `cas.userloc='%HOME'`
  - In this example, the user's personal caslib directory is named for his or her user ID and is located under `/local`: `cas.userloc='/local/%USER'`

**cas.useyarn=true | false**
Adds a reservation request to YARN for CAS memory size.

The memory limit for the YARN request is set with `cas.memorysize`.

- **Valid in:** CAS configuration file
- **Category:** Administration
- **Default:** false
- **See:** `cas.memorysize on page 17`
- **Example:** `cas.useyarn=true`

**Grouped by Categories**

Access Control Options
- **cas.perms on page 19**
Action Options
- cas.cmopt on page 10

Administration Options
- cas.elastic on page 12
- cas.lifetime on page 15
- cas.machinelist on page 16
- cas.memorysize on page 17
- cas.mode on page 18
- cas.nworkers on page 18
- cas.useyarn on page 22

Caslib Options
- cas.colocation on page 11
- cas.maxtablemem on page 17
- cas.userloc on page 22

Data Options
- cas.hdfsuserloc on page 13

DATA Step Options
- cas.datastepmsgsumlevel on page 11
- cas.datastepreplacetable on page 12

Formats Options
- cas.addfmtlib on page 9
- cas.fmtcaslib on page 13

Java
- cas.jreoptions on page 14

Localization
- cas.locale on page 15

Log Options
- cas.logcfgloc on page 16
- cas.logflush on page 16
- cas.messagelevel on page 18
- cas.metrics on page 18
CAS Environment Variables

Where Do I Set CAS Environment Variables?

You can set SAS Cloud Analytic Services environment variables for individual servers in the server's configuration file (for example, casconfig.lua), in the cas.settings file, or on the operating system command line. (By default, casconfig.lua is located in /opt/sas/viya/config/etc/cas/default and cas.settings in /opt/sas/viya/home/SASFoundation.)

If you edit either of these files (casconfig.lua or cas.settings) on a single node, only that node is affected. If you want to set an environment variable on every node, you need to edit either of those files on every node.

CAUTION! SAS Cloud Analytic Services ignores any instance of LD_LIBRARY_PATH found in the server configuration file. Set LD_LIBRARY_PATH in the cas.settings file only.

CAS Environment Variables Reference

Note: For information about SAS Cloud Analytic Services TLS environment variables, see "CAS TLS Environment Variables" in Encryption in SAS Viya.

env.CAS_DISK_CACHE="path [[:path] ...]"

Specifies the disk paths to cache data.

Delimit multiple paths with a colon (:).
Valid in | server configuration file, cas.settings file, and operating system command line
--- | ---
Category | Data
Restriction | Do not set to /tmp.
Tip | There is an advantage to using multiple physical disks. When using multiple threads, mapping files can occur concurrently if multiple disks are used. Hadoop also uses this method. Therefore, there is an advantage to using a set of disks that map to both hadoop_data and CAS_DISK_CACHE directories.
Example | `env.CAS_DISK_CACHE = '/data/disk1:/data/disk2'`

### `env.CAS_INSTALL='install-path'`
Specifies the installation directory for CAS.

- **Valid in**: server configuration file, cas.settings file, and operating system command line
- **Category**: Environment
- **Example**: `env.CAS_INSTALL='/opt/sas/viya/home/SASFoundation'`

### `env.CAS_LICENSE='path/license-file'`
Specifies the path and filename that contains the CAS license.

- **Valid in**: server configuration file, cas.settings file, and operating system command line
- **Category**: Administration
- **Example**: `env.CAS_LICENSE='/opt/sas/viya/config/etc/cas/default/SASViyaV0300_09JB84_Linux_x86-64.txt'`

### `env.CAS_VIRTUAL_HOST = 'host-name'`
The external host or machine name for the controller.

- **Use** `env.CAS_VIRTUAL_HOST` when an external HTTP client needs to use an external address that differs from the actual host name known by the operating system. A common use is when the controller machine is behind a proxy server.
- **Valid in**: server configuration file, cas.settings file, and operating system command line
- **Category**: Network
- **Example**: `env.CAS_VIRTUAL_HOST='my_machine'`

### `env.CAS_VIRTUAL_PORT=port`
The external port number for the controller.

- **Use** `env.CAS_VIRTUAL_PORT` when an external HTTP client needs to use a port that differs from the actual port that is local to the controller machine. A common use is when the controller machine is behind a proxy server.
- **Valid in**: server configuration file, cas.settings file, and operating system command line
- **Category**: Network
- **Example**: `env.CAS_VIRTUAL_PORT=5580`

### `env.CASCLIENTDEBUG=true | false`
Enables client-side debug messages to diagnose connection problems to the CAS server.
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Valid in</th>
<th>Category</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>env.HADOOP_HOME='path'</td>
<td>Specifies the standard HADOOP_HOME variable used by Hadoop.</td>
<td>server configuration file, cas.settings file, and operating system command line</td>
<td>Data</td>
<td>env.HADOOP_HOME='/opt/hadoop'</td>
</tr>
<tr>
<td>env.HADOOP_NAMENODE='machine-name : [machine-name]'</td>
<td>Identifies which machines in the Hadoop cluster are NameNodes. There can be up to two Hadoop NameNodes. Separate machine names with a colon (:). Machine-name can be a name, fully qualified domain name, or an IP address for a machine.</td>
<td>server configuration file, cas.settings file, and operating system command line</td>
<td>Data</td>
<td>env.HADOOP_NAMENODE='my_namenode1:my_namenode2'</td>
</tr>
<tr>
<td>env.TKTXTANIO_BINDAT_DIR='install-path'</td>
<td>Specifies the installation directory for SAS linguistic binary files required to perform text analysis. Note: TKTGDat.sh contains the SAS linguistic binary files required to perform text analysis in SAS LASR Analytic Server with SAS Visual Analytics and to run PROC HPTMINE and HPTMSCORE with SAS Text Miner.</td>
<td>cas.settings file, and operating system command line</td>
<td>Data</td>
<td>env.TKTXTANIO_BINDAT_DIR='/opt/sas/viya/home/SASFoundation/utilities/TKTGDat'</td>
</tr>
<tr>
<td>LD_LIBRARY_PATH= [path] : [path] ...</td>
<td>Specifies the path to search for additional shared libraries.</td>
<td>cas.settings file</td>
<td>Data</td>
<td>export LD_LIBRARY_PATH=/var/my_libs:/share/groups_libs:$LD_LIBRARY_PATH</td>
</tr>
</tbody>
</table>
Command: casadmin

Purpose

The **casadmin** command is used to perform limited server administration from a command line. The command is available from `/opt/sas/viya/home/SASFoundation/utilities/bin/casadmin` on a machine that has SAS Cloud Analytic Services installed.

Syntax

The syntax is as follows:

```
Usage: casadmin [options] command
Options:
   -host <hostname>     Specifies the CAS controller node name or IP address (CASHOST).
   -port <integer>      Specifies the CAS port number (CASPORT).
   -user <userid>       Specifies the administrative user (CASUSER).
   -verbose             Print the command and full response.
Commands:
   about                Get information about the server.
   ping                 Test whether the server is alive.
   shutdown             Stop the server.
```

You can use the CASHOST, CASPORT, and CASUSER environment variables instead of the related command-line options.
about
The `about` command returns information about the running CAS server.

ping
The `ping` command tests whether the server is running.

shutdown
The `shutdown` command stops the server if you are authorized to do so.

Example: Return Status Information for a Server
>>> ./casadmin -host cloud.example.com -port 5570 about
NOTE: Grid node action status report: 8 nodes, 1 total actions executed.

Example: Check That a Server Is Running
>>> ./casadmin -host cloud.example.com -port 5570 ping
>>> echo $?
0
A return code of zero indicates that the command succeeded. If the server is not running, you receive a message that is similar to the following:
ERROR: The TCP/IP tcpSockConnect support routine failed with error 111 (The connection was refused.).

Example: Shut Down a Server
>>> ./casadmin -host cloud.example.com -port 5570 shutdown
>>> echo $?
0
A return code of zero indicates that the command succeeded.

Authentication
The `casadmin` command searches for a `~/.authinfo` file. See the "How To" section of the Authentication information for details.

Troubleshooting
The following error response indicates that the `casadmin` command cannot access libraries that are required for encrypted communications:

```
ERROR: The encryption provider libraries cannot be found.
ERROR: Unable to load extension: (tkersa2)
```

To avoid the error message, include `SASFoundation/sasexe` in the `LD_LIBRARY_PATH` environment variable before you run the `casadmin` command.

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/sas/viya/home/SASFoundation/sasexe
```