

# SAS<sup>®</sup> Viya<sup>®</sup> 3.5 Administration: Using the Command-Line Interfaces

|   |           |
|---|-----------|
| <b>Command-Line Interface: Overview</b> .....                 | <b>2</b>  |
| Introduction .....  | 2         |
| How to Use This Document .....                                | 2         |
| Administrative CLI Plug-Ins .....                             | 2         |
| Key Points .....  | 5         |
| About the Examples .....                                      | 6         |
| <b>Command-Line Interface: Preliminary Instructions</b> ..... | <b>7</b>  |
| Set the SSL_CERT_FILE Environment Variable .....              | 7         |
| Download the CLI Executable .....                             | 8         |
| Create at Least One Profile .....                             | 10        |
| Use a Profile to Sign In .....                                | 11        |
| Batch Script Examples .....                                   | 13        |
| <b>Command-Line Interface: Syntax</b> .....                   | <b>14</b> |
| Structure .....   | 14        |
| Integrated Help .....   | 15        |
| <b>Command-Line Interfaces: Concepts</b> .....                | <b>18</b> |
| Output Type .....   | 18        |
| Global Command: Output .....                                  | 18        |
| Global Command: Profile .....                                 | 18        |
| Global Command: Authenticate .....                            | 21        |
| Source Files: JSON Templates .....                            | 22        |
| Source Files: Data .....                                      | 27        |
| <b>Command-Line Interface: Examples</b> .....                 | <b>29</b> |
| CLI Examples: CAS Administration .....                        | 29        |
| CLI Examples: Configuration .....                             | 32        |
| CLI Examples: Fonts .....                                     | 33        |
| CLI Examples: Inventory .....                                 | 34        |

|  |           |
|--|-----------|
| CLI Examples: Reports .....                          | 44        |
| <b>Command-Line Interface: Reference</b> .....       | <b>47</b> |
| Inventory CLI .....                                  | 47        |
| <b>Command-Line Interface: Troubleshooting</b> ..... | <b>50</b> |

---

# Command-Line Interface: Overview

---

## Introduction

SAS Viya contains administrative command-line interfaces (CLIs). In SAS Viya, a CLI is a user interface to the SAS Viya REST services where you enter commands on a command line and receive responses from the system. You can use a CLI to interact directly with SAS Viya programmatically without a GUI.

---

## How to Use This Document

This document describes administrative tasks that enable you to run the CLIs, provides information about how to use the integrated Help, and includes examples of how to use each CLI. Here is a strategy for getting started:

- [Set up](#) your environment to run the CLIs.
- Familiarize yourself with [CLI syntax](#) and the [integrated help](#) within the CLIs.
- See [“Administrative CLI Plug-Ins” on page 2](#) for information about each CLI plug-in and links to examples.

---

## Administrative CLI Plug-Ins

**IMPORTANT** Do not run the `sas-admin` commands as the root user on a host where SAS Viya 3.X is installed.

The following administrative CLIs are available in SAS Viya:

| Name                   | Scope and Examples  |
|------------------------|---|
| <code>sas-admin</code> | Top-level administrative command-line interface that is used to initialize, authenticate, and execute other plug-ins. |
| <code>audit</code>     | Gets SAS audit information.   |

| Name          | Scope and Examples  |  |
|---------------|---|--|
| authorization | See <a href="#">“Auditing: How To (CLI)”</a> in <i>SAS Viya Administration: Auditing</i> .  |  |
| authorization | Gets general authorization information and manages rules.<br>See <a href="#">“General Authorization: How To (CLI)”</a> in <i>SAS Viya Administration: General Authorization</i> .   |  |
| backup        | Manages backups.<br>See <a href="#">“CLI Examples: Backup”</a> in <i>SAS Viya Administration: Backup and Restore</i> .  |  |
| cas           | Manages CAS administration.   | See <a href="#">“CLI Examples: CAS Administration”</a> .   |
| cas           | Manages CAS authorization.  | See <a href="#">“CAS Authorization: How To (CLI)”</a> in <i>SAS® Viya Administration: CAS Authorization</i> .                    |
| cas           | Manages CAS users, sessions, and servers.   | See <a href="#">“SAS Cloud Analytic Services: How To (CLI)”</a> in <i>SAS Viya Administration: SAS Cloud Analytic Services</i> . |
| cas           | Manages CAS formats.  | See <a href="#">“CLI Examples: Formats”</a> in <i>SAS Viya Administration: Data</i> .  |
| cas           | Manages guest access for predefined caslibs.  | See <a href="#">“Enable Guest Access”</a> in <i>SAS Viya Administration: Authentication</i> .                                    |
| compute       | Manages the operations of the compute service.<br>See <a href="#">“CLI Examples”</a> in <i>SAS Viya Administration: Programming Run-Time Servers</i> .  |  |
| configuration | Manages the operations of the configuration service.<br>See <a href="#">“CLI Examples: Configuration”</a> .   |  |
| dagentsrv     | Manages interactions with the SAS Data Agent server. Usage of this plug-in is license restricted.<br>See <a href="#">“CLI ‘dagentsrv’”</a> in <i>Cloud Data Exchange for SAS Viya: Administrator’s Guide</i> .                                  |  |
| dataexplorer  | The dataexplorer plug-in is no longer available as of February 7, 2022.   |  |
| dcmtransfer   | Enables you to transfer rule sets, rule flows, lookup tables, and decisions from a SAS 9.4 environment to a SAS Viya environment.<br>See <a href="#">“dcmtransfer Plug-in”</a> in <i>SAS Intelligent Decisioning: Command-Line Interfaces</i> . |  |
| devices       | Manages mobile device blacklist and whitelist actions and information.<br>See <a href="#">“Mobile: How To (CLI)”</a> in <i>SAS Viya Administration: Mobile</i> .  |  |

| Name           | Scope and Examples   |
|----------------|--|
| folders        | <p>Manages SAS folders.</p> <p>See <a href="#">“Folders: CLI Examples”</a> in <i>SAS Viya Administration: Folders</i>.</p>   |
| fonts          | <p>Manages fonts that are provided by SAS as well as custom fonts that are registered in SAS Visual Analytics 8.3 and later.</p> <p>See <a href="#">“CLI Examples: Fonts”</a>.</p>                     |
| healthcheck    | <p>Queries the system and services and returns status results.</p> <p>See <a href="#">“Monitoring: How To (CLI)”</a> in <i>SAS Viya Administration: Monitoring</i>.</p>                                |
| identities     | <p>Retrieves identity information and manages custom groups.</p> <p>See <a href="#">“Identity Management: How To (CLI)”</a> in <i>SAS Viya Administration: Identity Management</i>.</p>                |
| inventory      | <p>Scans the contents of a SAS Viya deployment and generates output tables that can be consumed by the SAS Viya Inventory report.</p> <p>See <a href="#">“CLI Examples: Inventory”</a> on page 34.</p> |
| job            | <p>Manages the operations of the job flow scheduling service.</p> <p>See <a href="#">“Jobs and Flows: How to (Command Line Interface)”</a> in <i>SAS Viya Administration: Jobs and Flows</i>.</p>      |
| launcher       | <p>Manages the operations of the launcher service.</p> <p>See <a href="#">“CLI Examples”</a> in <i>SAS Viya Administration: Programming Run-Time Servers</i>.</p>                                      |
| licenses       | <p>Manages SAS product license status and information.</p> <p>See <a href="#">“Licensing: How To (CLI)”</a> in <i>SAS Viya Administration: Licensing</i>.</p>  |
| qkbs           | <p>Manages the SAS Quality Knowledge Bases (QKBs) on a CAS server.</p> <p>See <a href="#">“How To (CLI)”</a> in <i>SAS Viya Administration: QKB Management</i>.</p>                                    |
| reports        | <p>Manages reports in SAS Visual Analytics 8.2 and later.</p> <p>See <a href="#">“CLI Examples: Reports”</a>.</p>  |
| restore        | <p>Manages restore operations.</p> <p>See <a href="#">“CLI Examples: Restore”</a> in <i>SAS Viya Administration: Backup and Restore</i>.</p>   |
| scoreexecution | <p>Manages resources that are no longer used by the Score Execution service.</p> <p>See <a href="#">“scoreexecution Plug-in”</a> in <i>SAS Intelligent Decisioning: Command-Line Interfaces</i>.</p>   |
| tenant         | <p>Manages tenants in a multi-tenant deployment.</p> <p>See <a href="#">“How-To (CLI)”</a> in <i>SAS Viya Administration: Multi-tenancy</i>.</p>   |

| Name     | Scope and Examples   |
|----------|--|
| transfer | Promotes SAS content.<br>See <a href="#">“Promotion: How To (CLI)” in SAS Viya Administration: Promotion (Import and Export)</a> . |

## Key Points

Here are key points for using the CLIs:

- SAS recommends that users with administrative privileges run the sas-admin CLI in order to ensure optimal results. Users without administrative privileges are able to run the sas-admin CLI, but the results might not always be as expected.

**IMPORTANT** Do not run the sas-admin commands as the root user on a host where SAS Viya 3.X is installed.

- To prepare to use the sas-admin CLI plug-ins, see [“Command-Line Interface: Preliminary Instructions” on page 7](#).
- Commands and subcommands are case sensitive.
- You must precede an option of a command with two hyphens (--) if the option is a word. If the option is a single letter, you can precede the option with two hyphens (--) or one hyphen (-). For example, you can use --help or -h for the Help global option.
- Linux special characters that are specified on the command line must be preceded (or escaped) with a backslash (\) so that they are not interpreted by the Linux shell. Linux special characters that are included in JSON template files do not need to be escaped.
- If a single parameter contains spaces (such as an object’s name), you must enclose it in quotation marks in order to pass it as one item. Depending on what can be parsed by your operating system, other parameter values might need to be enclosed in quotation marks or escaped accordingly.
- Use the Help from within each CLI for information about the available commands, subcommands, and options. For the sas-admin CLI plug-ins, see [“Command-Line Interface: Syntax” on page 14](#).
- Timestamps that are returned from the sas-admin CLIs are in Universal Time Coordinated (UTC) format rather than in local time. By contrast, timestamps returned from SAS Environment Manager are in local time.

For example, in SAS Environment Manager, you might notice that a SAS Visual Analytics report has a modified date of April 26, 2018 09:23:47. This is the local time. However, the reports CLI shows that the modified date for the same report is 2018-04-26T13:23:47.314Z. This is UTC time.

**Note:** The integrated Help supersedes this documentation and provides the most current information about any expanded or enhanced functionality.

---

## About the Examples

The following points apply to all of the examples in this document:

- The examples assume that you have signed in to SAS Viya using the command line. See [“Command-Line Interface: Preliminary Instructions” on page 7](#).
- The examples explicitly specify all necessary options. In practice, you might find it more efficient and concise to use environment variables where available. Remember to clear values for any environment variables when appropriate.
- The examples generally were run in a Linux environment.
- The examples include line breaks within commands to enhance readability. Do not include line breaks when you submit a command.
- The examples generally include single quotation marks when quotation marks are required. Use the quotation marks that are appropriate for your platform.
- The examples generally assume that you are using the default profile rather than a named profile.

---

**Note:** If you log in using a named profile, you can specify the PROFILE option in the CLI command or you can set it in an environment variable:

- Specify the PROFILE option in the CLI command.

Here is the command format:

```
sas-admin --profile profile-name CLI-name CLI-commands
```

In the following example, **Target1** is the profile name:

```
sas-admin --profile Target1 auth login
```

In the following example, **Target1** is the profile name in an environment that has been configured for Kerberos:

```
sas-admin --profile Target1 auth kerberos
```

- Set the SAS\_CLI\_PROFILE environment variable to the name of the profile. The environment variable remains in effect until you log off.

For example, if you set the SAS\_CLI\_PROFILE environment variable to **Target1**, when you sign in to SAS Viya, the environment variable is applied automatically.

In the following example, **Target1** is not specified explicitly because it has been set via the environment variable:

```
sas-admin auth login
```

If you are logging in to an environment that has been configured for Kerberos, the environment variable is applied automatically and does not have to be specified in the CLI. You specify Kerberos.

Here is an example:

```
sas-admin auth kerberos
```

---

---

# Command-Line Interface: Preliminary Instructions

Complete the following required preliminary tasks before you use a CLI.

---

## Set the SSL\_CERT\_FILE Environment Variable

If your environment is enabled for Transport Layer Security (TLS), you must set the SSL\_CERT\_FILE environment variable to the path location of the trustedcerts.pem file (if using the SAS default truststore) or the path location of your site-signed certificate (if using an internal truststore).

**Note:** For CLI users on Linux who are running the CLIs directly on the SAS machine, you might be able to [source the consul.conf file](#) rather than set the SSL\_CERT\_FILE environment variable.

Set the SSL\_CERT\_FILE environment variable to one of the following values:

| Truststore             | Operating System | Path  |
|------------------------|------------------|---|
| SAS default truststore | Linux            | <code>/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.pem</code>    |
|                        | Windows          | <code>C:\ProgramData\SAS\Viya\etc\SASSecurityCertificateFramework\cacerts\trustedcerts.pem</code> |
| internal truststore    | Linux or Windows | <code>path_to_truststore_including_name</code>  |

Here is an example of how to set the environment variable in a Linux environment:

```
export SSL_CERT_FILE=/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.pem
```

---

## Download the CLI Executable

### Overview

You can download the sas-admin CLI directly from the SAS Support website and install the plug-ins that you need. This is the recommended approach for running the sas-admin CLI from client machines. The most current plug-ins are available with the downloaded sas-admin CLI.

All plug-ins are available for installation with the downloaded sas-admin CLI with the exception of the following:

|         |                   |
|---------|-------------------|
| restore | Is not available. |
| tenant  | Is not available. |

---

**Note:** Alternatively, the sas-admin CLI and its plug-ins is available on every SAS Viya server in the deployment. If you are not downloading the sas-admin CLI to run on a client machine or if you want to run a plug-in that is not available with the downloaded sas-admin CLI, you can run the sas-admin CLI directly from one of the following locations on the SAS Viya server:

- Linux: `/opt/sas/viya/home/bin`
  - Windows: `C:\Program Files\SAS\Viya\bin`
- 

### How To

Here are the steps to download the sas-admin CLI and install the plug-ins.

---

**Note:** You must have a valid SAS profile on [support.sas.com](https://support.sas.com) or on [the SAS website](#) to download the sas-admin CLI.

---

- 1 Go to the [Support / Downloads and Hot Fixes page](#) and download the appropriate file to your machine.
- 2 Prepare the files for your environment.

---

UNIX

Expand the file to a directory from which you plan to run the CLI:

```
tar -xvzf /CLI-directory/sas-admin-cli-1.1.2-download-linux.tgz
```

From the directory, make sure that the Execute permission is set:

```
chmod +x sas-admin
```

---



---

Windows or Macintosh    Unzip the file and place it in a directory from which you plan to run the CLI.

---

### CAUTION

Do not place the sas-admin CLI download file in either of the following directories:

---

**Table 1** CLI Default Directory Prohibited from Download

|         |                               |
|---------|-------------------------------|
| Linux   | /opt/sas/viya/home/bin        |
| Windows | C:\Program Files\SAS\Viya\bin |

- 3 Navigate to the directory location where you saved the CLI file. You must run the CLI commands directly from this directory, or you can add this directory to your system path to make the sas-admin CLI available to all CLI users.
- 4 [Create a profile and sign in.](#)
- 5 Install the CLI plug-ins by running these commands:

```
1 sas-admin plugins
2 sas-admin plugins list
3 sas-admin plugins list-repo-plugins
4 sas-admin plugins install --repo SAS plugin-name
```

- 1 Display the commands that are available to the sas-admin CLI plug-ins command.
- 2 Display the sas-admin CLI plug-ins that are currently installed.
- 3 List the plug-ins in the SAS repository that are available for installing.
- 4 Install a plug-in from the SAS repository.

---

**Note:** The plug-ins are installed in the home directory of the user who ran the `sas-admin plugins` command. For example, `home-directory/.sas/admin-plugins`.

---

## Run the sas-admin CLI Plug-in from a Shared Location

**IMPORTANT** SAS recommends that each sas-admin CLI user should download a unique instance of the sas-admin CLI and install a unique instance of a plug-in from the SAS repository. If your organization has a policy that enforces the download of only a single instance, it is still possible to make the CLI plug-ins available from a shared location as follows:

- 1 [Create a profile and sign in.](#)
- 2 Navigate to the config.json file according to the operating system being used.

**Table 2** CLI Directory for config.json File

|           |  |
|-----------|--|
| Linux     | <code>home-directory/.sas/admin-plugins</code> |
| Windows   | <code>home-directory\.sas\admin-plugins</code> |
| Macintosh | <code>home-directory/.sas/admin-plugins</code> |

- 3 Open the config.json file. For each CLI plug-in, find the line that contains the location, and modify the path to point to a shared location that contains the sas-admin CLI plug-ins directory. Here is an example of this section in the config.json file for the transfer CLI plug-in:

```
"transfer": {
  "location": "\shared-location\admin-plugins\sas-transfer-cli-1.3.5-20180330.1522434840.exe",
  "description": "tool for promoting contents across environments",
  "version": "1.3.5"
```

- 4 Propagate this change to the config.json file for all users.

## Create at Least One Profile

**IMPORTANT** Do not run the sas-admin commands as the root user on a host where SAS Viya 3.X is installed.

If you have not already created a profile for the environment that you want to use, complete the following steps.

- 1 If you downloaded the CLI executables separately, navigate to the directory on the machine that contains the CLIs. If you run CLIs directly from the SAS Viya server and you have not downloaded the CLI executables separately, from a command prompt on the SAS Viya server, navigate to the appropriate directory.

**Table 3** CLI Default Directory

|         |  |
|---------|--|
| Linux   | <code>/opt/sas/viya/home/bin</code>      |
| Windows | <code>\Program Files\SAS\Viya\bin</code> |

- 2 At the command prompt, enter a command to initialize a new profile. Here are examples:

To create a default profile, enter: `sas-admin profile init`

To create a profile called `prod`, enter: `sas-admin --profile prod profile init`

You can use a named profile to access different environments using the same set of CLIs. See [“Default Profile and Named Profiles” on page 19](#) for information about why you might want to use a named profile.

**Note:** Running the PROFILE global command creates a config.json file in this directory: *home-directory/.sas*. For more information, see [“Overview” on page 18](#).

**3** Respond to the subsequent prompts as follows:

|                            |  |
|----------------------------|--|
| Service Endpoint           | Specify the URL for the SAS Viya environment. Use the following format:<br><br><i>communications-protocol://web-server-host-name:web-server-port</i><br><br><b>Note:</b> The port is required only if it is not the default port of 443.<br><br>For example: <code>https://host.example.com</code> |
| Output type                | Specify your preferred format for CLI output (text, json, or fulljson).<br><br><b>Note:</b> For more information about the output types, see <a href="#">“Output Type” on page 18</a> .  |
| Enable ANSI colored output | Specify whether to enable colored output (y or n).   |

**4** Repeat steps 2 and 3 for any additional profiles that you want to create.

## Use a Profile to Sign In

**IMPORTANT** Do not run the sas-admin commands as the root user on a host where SAS Viya 3.X is installed.

- 1 If you downloaded the CLI executables separately, navigate to the directory on the machine that contains the CLIs. If you run CLIs directly from the SAS Viya server and you have not downloaded the CLI executables separately, from a command prompt on the SAS Viya server, navigate to the appropriate directory.

**Table 4** CLI Default Directory

|          |  |
|----------|--|
| Linux:   | <code>/opt/sas/viya/home/bin</code>      |
| Windows: | <code>\Program Files\SAS\Viya\bin</code> |

- 2 At the command prompt, enter a command to initiate the sign-in process. Here are examples:

---

To use your default profile (assuming that the SAS\_CLI\_PROFILE environment variable is not set), enter:

```
sas-admin auth login
```

---

To use a profile called `prod`, enter:

```
sas-admin --profile prod auth login
```

---

**Note:** Kerberos is supported for Linux and Windows environments. If you are signing in to an environment that has been configured for Kerberos, replace `auth login` with `auth kerberos` in the preceding examples.

---

- 3 At the subsequent prompts, enter your user ID and password. No prompts appear in environments that have been configured for Kerberos.

The authentication has a default timespan before it expires. See “[sas.logon.jwt](#)” in *SAS Viya Administration: Configuration Properties* for more information about the default expiry of an access token in SAS.

You can use the AUTH LOGOUT command to sign out.

---

**Note:** When you run the AUTH LOGOUT global command, a bearer token is written to the `credentials.json` file in this directory: `home-directory/.sas`. For more information, see “[Overview](#)” on [page 18](#).

---

**Note:** If you log in using a named profile, you can specify the PROFILE option in the CLI command or you can set it in an environment variable:

- Specify the PROFILE option in the CLI command.

Here is the command format:

```
sas-admin --profile profile-name CLI-name CLI-commands
```

In the following example, `Target1` is the profile name:

```
sas-admin --profile Target1 auth login
```

In the following example, `Target1` is the profile name in an environment that has been configured for Kerberos:

```
sas-admin --profile Target1 auth kerberos
```

- Set the SAS\_CLI\_PROFILE environment variable to the name of the profile. The environment variable remains in effect until you log off.

For example, if you set the SAS\_CLI\_PROFILE environment variable to `Target1`, when you sign in to SAS Viya, the environment variable is applied automatically.

In the following example, `Target1` is not specified explicitly because it has been set via the environment variable:

```
sas-admin auth login
```

If you are logging in to an environment that has been configured for Kerberos, the environment variable is applied automatically and does not have to be specified in the CLI. You specify Kerberos.

Here is an example:

```
sas-admin auth kerberos
```

---

## Batch Script Examples

You can create a profile and sign in using a batch script. The following examples are from a Linux environment.

- Here is an example of a script that includes the password directly in the script:

---

### CAUTION

If you include a password in the script, you must secure it so that it can be read only by the administrator and the execution tools that use it.

---

```
#!/bin/sh

clidir=/opt/sas/viya/home/bin

$clidir/sas-admin profile set-endpoint "https://host.example.com"
$clidir/sas-admin profile toggle-color off
$clidir/sas-admin profile set-output fulljson

$clidir/sas-admin auth login --user userID --password password
```

---

**Note:** If your password includes special characters, then you must enclose the password in single quotation marks.

---

- Here is an example of a script that uses the `pyviyatools` command-line tool named `loginviaauthinfo.py` to authenticate using an `authinfo` file:

```
#!/bin/sh

clidir=/opt/sas/viya/home/bin
pyviyadir=/pyviyatools

$clidir/sas-admin profile set-endpoint "https://host.example.com"
$clidir/sas-admin profile toggle-color off
$clidir/sas-admin profile set-output fulljson

$pyviyadir/loginviaauthinfo.py
```

---

**Note:** See the [pyviyatools GitHub repository](#) for more information about `pyviyatools`. See [Client Authentication Using an Authinfo File](#) for more information about the `authinfo` file.

---

**TIP** Please report any problems or questions regarding pyviyatools through the [Issues](#) page in GitHub. Do not contact SAS Technical Support for assistance with pyviyatools.

---

**Note:**

- If you can securely obtain an access token from SASLogon, then you can assign the access token to the OAUTH\_TOKEN environment variable to eliminate the need to hardcode the password in the script. See [“Obtain an Access Token Using Password Credentials”](#) in *SAS Viya Administration: Authentication* for more information.
  - If your system is configured for Kerberos, then a user is authenticated with their available Kerberos ticket. No user name and password are necessary.
- 

## See Also

[“Command-Line Interface: Overview”](#) on page 2

---

# Command-Line Interface: Syntax

---

## Structure

The basic structure of a command-line interface (CLI) command is:

```
sas-admin [global options] plug-in-name [command] [command options] [subcommand]
[subcommand options] [arguments]
```

---

**Note:** Order matters. The global options are not recognized unless they appear after sas-admin.

---

**sas-admin**

specifies the sas-admin CLI.

**[global options]**

specifies options that are applicable to all CLIs.

**plug-in name**

specifies the CLI plug-in.

**[command]**

specifies a command that is specific for the plug-in that you are using.

**[command options]**

specifies options for the plug-in specific command that you are using.

**[subcommand]**

specifies a subcommand for the plug-in command that you are using.

**[subcommand options]**

specifies options for the subcommand.

**[arguments]**

specifies arguments for options.

Here are some basic examples of issuing CLI commands:

**Example:** List the records of audit entries, and specify to use the JSON output type.

```
sas-admin --output json audit list
```

**Example:** Show more detailed information for the devices plug-in BLACKLIST LIST command.

```
sas-admin --verbose devices blacklist list
```

For information about the CLIs that are available, see [“Administrative CLI Plug-Ins” on page 2](#).

## Integrated Help

### Global Commands

Use the integrated Help within the CLI to learn about the available global commands, plug-ins, and global options. The global options apply to each CLI plug-in.

**Example:** List all the global commands, plug-ins, and global options for the sas-admin CLI.

```
sas-admin help
```

Here is the output from this command in a Linux environment:

```
NAME:
  sas-admin - SAS Administrative Command Line Interface

USAGE:
  sas-admin [global options] command [command options] [arguments...]

VERSION:
  1.1.26

COMMANDS:
  authenticate, auth, authn  Handles authentication to the target environment.
  help, h                    Shows a list of commands or help for one command.
  plugins                    Manages plugins.
  profile, prof              Shows and updates options.

PLUGINS:
  audit
  authorization
  backup
  cas
  compute
  configuration
  devices
  folders
  fonts
  healthcheck
```

```

identities
inventory
job
launcher
licenses
qkbs
reports
restore
tenant
transfer

```

GLOBAL OPTIONS:

```

--colors-enabled      Enables or disables ANSI colored output. [$SAS_CLI_COLOR]
--help, -h           Shows help.
--insecure, -k       Allows connections to TLS sites without validating the
server certificates.
--locale "en"        Specifies a locale to use. [$LC_ALL, $LANG]
--log-file            Specifies the file to write log events to. [$SAS_LOG_FILE]
--output              Specifies output format - text, json, fulljson. [$SAS_OUTPUT]
--profile, -p "Default" Specifies a named profile to use. [$SAS_CLI_PROFILE]
--quiet, -q          Quiets spurious output, data only.
--sas-endpoint        Sets the URL to the SAS services. [$SAS_SERVICES_ENDPOINT]
--verbose            Shows detailed processing information and output.
--version, -v        Prints the version.

```

COPYRIGHT:

(c) 2016-2019 SAS Institute Inc. All Rights Reserved.

**Example:** Show the version of the CLI.

```
sas-admin --version
```

Here is the output from this command in a Linux environment:

```
sas-admin version 1.1.26
```

## CLI Plug-in

Use the integrated Help within the CLI to learn about the available commands, subcommands, and options for each CLI plug-in. Use the same syntax for each CLI, substituting the CLI plug-in name or command that you are getting help for.

**Example:** List all the commands for the devices plug-in to the sas-admin CLI.

```
sas-admin devices --help
```

Here is the output from this command in a Linux environment:

NAME:

```
sas-devices
```

USAGE:

```
sas-admin devices command [command options] [arguments...]
```

COMMANDS:

```

authorized-devices  Manages the authorization of devices.
blacklist           Manages the list of blacklisted devices.
enforcement         Manages the policy enforcement of mobile devices.

```



|             |  |
|-------------|--|
| help, h     | Shows a list of commands or help for one command.                                  |
| last-access | Manages the set of history records of the devices that use the mobile application. |
| whitelist   | Manages the list of whitelisted devices.   |

**Example:** List the subcommands of the `blacklist` command that is a part of the devices plug-in to the `sas-admin` CLI.

```
sas-admin devices help blacklist
```

Here is the output from this command in a Linux environment:

NAME:

```
sas-devices blacklist - Manages the list of blacklisted devices.
```

USAGE:

```
sas-devices blacklist command [command options] [arguments...]
```

COMMANDS:

|         |  |
|---------|--|
| add     | Adds a device to the blacklist.                          |
| clear   | Clears the blacklist of all device IDs that are present. |
| delete  | Deletes a device from the blacklist.                     |
| help, h | Shows a list of commands or help for one command.        |
| list    | Lists the devices in the blacklist.                      |

OPTIONS:

```
--help, -h Shows help.
```

**Example:** List the options of the `list` subcommand of the `blacklist` command that is a part of the devices plug-in to the `sas-admin` CLI.

```
sas-admin devices blacklist help list
```

Here is the output from this command in a Linux environment:

NAME:

```
sas-devices blacklist list - Lists the devices in the blacklist.
```

USAGE:

```
sas-devices blacklist list [command options] [arguments...]
```

OPTIONS:

|              |   |
|--------------|---|
| --all        | Returns all of the devices in the blacklist.  |
| --limit "10" | Specifies the maximum number of devices to return. The default value is 10.         |
| --start "0"  | Specifies the 0-based offset of the first device to return. The default value is 0. |

---

# Command-Line Interfaces: Concepts

---

## Output Type

You must specify an output type for your CLI when you create your profile. The output types for CLIs are as follows:

- text  
Specifies that the output from the CLI is in text format. This is the default format.
- json  
Specifies that the output from the CLI is in JSON format.
- fulljson  
Specifies that the output from the CLI is the entire JSON response. This option is useful when writing scripts in which you need access to the entire response in order to complete a task.

---

## Global Command: Output

Use the OUTPUT global command to specify the output format for a CLI command. Doing so overrides any output type that was specified when you created your profile or that was set in the SAS\_OUTPUT environment variable.

---

## Global Command: Profile

### Overview

Use the PROFILE global command to create the connection profile that defines your SAS Viya deployment. This process creates the following two files in the directory *home-directory/.sas*:

- config.json  
Contains information about your SAS Viya deployment, including the name of the connection profile, the service endpoint, and the output type (TEXT, JSON, FULLJSON).

.....  
**Note:** If you downloaded the CLI executables, the config.json file is created in this directory:  
.....

**Table 5** CLI Directory for config.json File

|           |  |
|-----------|--|
| Linux     | <i>home-directory/.sas/admin-plugins</i> |
| Windows   | <i>home-directory\.sas\admin-plugins</i> |
| Macintosh | <i>home-directory/.sas/admin-plugins</i> |

- credentials.json

Contains the authentication tokens for your session that are created after you issue the AUTH LOGIN global command.

## Default Profile and Named Profiles

You can create a default profile or a named profile. If you do not specify a named profile in the SAS-ADMIN AUTH LOGIN command, and the SAS\_CLI\_PROFILE environment variable is not set, then the default profile is used.

You can use a named profile if you need to work with two or more environments simultaneously from the same machine using the same set of CLIs. When you log on to an environment with a named profile, the associated token is stored for that specific profile. You can work in different environments at the same time by specifying different profile names in the commands. For example, suppose that you have different development, test, and production environments. You can create a separate, named profile for each environment to help distinguish the environment that you are connecting to.

You can use a named profile to eliminate the need to create a profile with the correct settings every time you log on. Suppose that you know that you want to use the JSON output type and a certain endpoint. You can create a named profile with these settings. Then, when you want to log on, you can specify this named profile.

---

**Note:** If you log in using a named profile, you can specify the PROFILE option in the CLI command or you can set it in an environment variable:

- Specify the PROFILE option in the CLI command.

Here is the command format:

```
sas-admin --profile profile-name CLI-name CLI-commands
```

In the following example, **Target1** is the profile name:

```
sas-admin --profile Target1 auth login
```

In the following example, **Target1** is the profile name in an environment that has been configured for Kerberos:

```
sas-admin --profile Target1 auth kerberos
```

- Set the SAS\_CLI\_PROFILE environment variable to the name of the profile. The environment variable remains in effect until you log off.

For example, if you set the SAS\_CLI\_PROFILE environment variable to **Target1**, when you sign in to SAS Viya, the environment variable is applied automatically.

In the following example, **Target1** is not specified explicitly because it has been set via the environment variable:

```
sas-admin auth login
```

If you are logging in to an environment that has been configured for Kerberos, the environment variable is applied automatically and does not have to be specified in the CLI. You specify Kerberos.

Here is an example:

```
sas-admin auth kerberos
```

## Examples

Here are typical examples of creating default and named profiles in an environment that has not been configured to use Kerberos:

**Example:** Create a default connection profile, specify the TEXT output type, and specify the path to your SAS Viya deployment as the service endpoint.

**Note:** The SAS\_CLI\_PROFILE environment variable must not be set in order for this example to work.

```
sas-admin profile init
```

- At the prompt for service endpoint, enter the URL for the SAS Viya environment as follows:  
`https://endpoint URL`
- At the prompt for output type, enter `text`.
- At the prompt for "whether to enable ANSI colored output", enter `y` or `n`.

Here is the config.json file that was created. "Default" indicates that a Default profile was created.

```
{
  "Default": {
    "ansi-colors-enabled": "false",
    "output": "text",
    "sas-endpoint": "https://endpoint URL"
  }
}
```

**Example:** In the same environment, create a connection profile that is named Target1, specify the JSON output type, and specify the path to your SAS Viya deployment as the service endpoint.

```
sas-admin --profile Target1 profile init
```

Here is the config.json file that was created. Notice that there are now two profiles: Default and Target1. Notice that the output type for the Target1 profile is JSON.

```
{
  "Default": {
    "ansi-colors-enabled": "false",
    "output": "text",
    "sas-endpoint": "https://endpoint URL"
  },
  "Target1": {
    "ansi-colors-enabled": "false",
    "output": "json",
    "sas-endpoint": "https://endpoint URL"
  }
}
```

See “[Command-Line Interface: Preliminary Instructions](#)” on page 7 for more information about creating profiles.

**Example:** In the same environment, log on using the Target1 profile that you created in the previous example.

```
sas-admin --profile Target1 auth login
```

---

## Global Command: Authenticate

Use the AUTHENTICATE global command to log on or log off from the environment. This process stores a token in the credentials.json file.

---

### Note:

The authentication has a default timespan before it expires, so you might need to re-execute the command at a later time to reconnect to the environment. See “[sas.logon.jwt](#)” in *SAS Viya Administration: Configuration Properties* for more information about the default expiry of an access token in SAS.

---

To log on, run the AUTH LOGIN command. For syntax information, see “[Structure](#)” on page 14.

---

**Note:** Kerberos is supported for Windows and Linux environments. Sign in to environments that have been configured for Kerberos with this command: AUTH KERBEROS

---

Here are typical examples from a Linux environment that has not been configured for Kerberos:

- Use this command to log on with a default profile: `sas-admin auth login`. The `SAS_CLI_PROFILE` environment variable must not be set in order for this example to work.
- Use this command to log on with the profile named Target1: `sas-admin --profile Target1 auth login`

If the `SAS_CLI_PROFILE` environment variable is set to `Target1`, then you can log on to the Target1 environment as follows: `sas-admin auth login`

---

To log off, issue the AUTH LOGOUT command. For syntax information, see “[Structure](#)” on page 14. Here are typical examples:

- Use this command to log off from the SAS Viya environment that is specified in your default profile: `sas-admin auth logout`. The `SAS_CLI_PROFILE` environment variable must not be set in order for this example to work.
- Use this command to log off from a SAS Viya environment that is specified in the profile named Target1: `sas-admin --profile Target1 auth logout`.

If the `SAS_CLI_PROFILE` environment variable is set to `Target1`, then you can log off from the Target environment as follows: `sas-admin auth logout`.

---

## Source Files: JSON Templates

### Overview

For some CLI commands, you can include the command options in a separate JSON file by using the `SOURCE-FILE` option. This file is referred to as a template or source file. You might choose this option for the following reasons:

- to accelerate the process for entering CLI commands by including common options in a template
- to automatically create a template by piping output to a file

### Details

- If a duplicate CLI option is specified on the command line and in the source file, the command-line option takes precedence over the option in the source file.
- Although some CLI options can be specified in the source file and on the command line, other CLI options can be specified only on the command line.

## Create Caslibs from JSON Templates

Templates that are used for creating caslibs must be JSON files. You can generate sample JSON templates for creating caslibs with the `cas CLI GENERATE-CAS-SAMPLES` command.

**Example:** Generate a template for creating a caslib. In this example, a template is generated for creating a path caslib. You will generate the `path.json` sample template and use it to create the new template.

```
1 sas-admin cas generate-cas-samples --output-location sample-template-path
2 sas-admin cas caslibs create --help
3 sas-admin cas caslibs create path --server serverA --path path --name caslibA
  --source-file /sample-template-path/path.json
4 sas-admin --output json cas caslibs show-info --name caslibA > pathA.json
```

- 1 Generate sample templates. The `path.json` template is one of the files that are created. Specify the folder location for the templates in the `output-location` option.
- 2 List the types of caslibs that can be specified. You will use this value with the `create` command in the next step.
- 3 Create the new path caslib with the name `caslibA`. In the `source-file` option, specify the `path.json` template that you just created. In the next step, you use the information about the new caslib to create a JSON template.

On the command line, specify any options for overriding the values in the sample template. In this example, the server and path were specified.

- 4 Redirect the JSON output for `caslibA` to a file named `pathA.json`, which you can then use as a template for creating new caslibs. Specify the use of JSON output with the `OUTPUT JSON` global option.

**Example:** Create a caslib from a template. This example uses the template that you created in the previous example.

```
1 sas-admin cas caslibs create --help
2 sas-admin cas caslibs create path --server serverB --path pathB --name caslibB
  --source-file pathA.json
3 sas-admin --output json cas caslibs show-info --name caslibB --server serverB
```

- 1 List the types of caslibs that can be specified. You use this information in the `create` command in the next step.
- 2 Create a new caslib named `caslibB` by specifying the `pathA.json` file that you created in the previous example in the `source-file` option. On the command line, specify any options for overriding the values in the template. In this example, the server and path were specified.
- 3 Show information about `caslibB` in JSON format. Note that the server and path that were specified on the command line overwrote the server and path values that were in the template.

## Create User-Defined Formats from JSON Templates

In templates that are used for adding a user-defined format, templates that include command options as well as the format ranges must be JSON files. You can generate JSON templates for creating user-defined formats with the cas CLI `generate-cas-samples` command.

**Example:** Create a JSON template for adding a user-defined format.

```
1 sas-admin cas generate-cas-samples --output-location sample-template-path
2 sas-admin cas format-libraries add-format json --format-library format_libraryA --
  server serverA
  --source-file /sample-template-path/gender.json
3 sas-admin cas --output json format-libraries show-format-ranges --format 'en_us-
  $gender'
  --format-library format_libraryA --server serverA > formatA_template.json
4 sas-admin cas format-libraries add-format json --format-library format_libraryB
  --server serverA --source-file formatA_template.json
```

- 1 Generate sample templates. The `gender.json` template is one of the files that are created. Specify the folder location for the templates in the `output-location` option.
- 2 Add a new format to a format library. In the `source-file` option, specify the `gender.json` template that you just created. In the next step, you will use the information about the new format to create a JSON template.

On the command line, specify any options for overriding the values in the template. In this example, the format library was specified. If the command is successful, the server returns a message that identifies the name of the format that you added. In this example, the `en_us-$gender` file is created. The `en_us-` prefix identifies the English locale.

- 3 Redirect the JSON output for the new format to a JSON file using the `format-libraries show-format-ranges` command. Specify the use of JSON output with the `output json` global option. The format name to use in the `format` option was returned from the server when you created the format. Because the `gender` format is a character format, you must precede it with a dollar sign (\$) and enclose it in quotation marks.

You can use the JSON file as a template for creating new caslibs. Note that the format library that you specified when you created the new format is saved in the new JSON file.

- 4 Create a new format using the new JSON file as the `source-file` template. On the command line, specify any options for overriding the values in the template. In this example, a new format library was specified.

## Create Policies from JSON Templates

Templates that are used for creating CAS server policies must be JSON files.

**IMPORTANT** For CAS to use resource management policies, the environment variable `CAS_ENABLE_CONSUL_RESOURCE_MANAGEMENT` must be set on the CAS server. For more information, see [Environment Variables](#).

### Global Caslib Policies

**Example:** Create a policy from a sample JSON template.

```
1 sas-admin cas generate-cas-samples --output-location sample-template-path
2 sas-admin cas servers policies define global-caslibs
  --server serverA --source-file /sample-template-path/policies-examples/
  globalCaslibs.json
```

- 1 Generate templates. Specify the folder location for the templates in the `output-location` option. The `policies-examples` folder is created, and one of the files in the folder is `globalCaslibs.json`.

Edit the `globalCaslibs.json` file for your environment. You can [add new policy options](#) or change values of existing policy options. Save the file.

---

**Note:** As an alternative to manually editing the JSON file, you can add additional options by using the `ATTRIBUTES` option.

---

- 2 Define a global caslib policy by specifying the edited `globalCaslibs.json` template file in the `source-file` option. The values in the `globalCaslibs.json` file are used to define the new policy. Information about the new policy is returned from the server after the policy is created. Make sure that the policy is correct.

**Example:** Create a JSON template for creating a global caslib policy.

```
1 sas-admin cas generate-cas-samples --output-location sample-template-path
2 sas-admin cas servers policies define global-caslibs
  --server serverA --source-file /sample-template-path/policies-examples/
  globalCaslibs.json
3 sas-admin --output json cas servers policies show-info --policy globalCaslibs
  --server serverA > /path/global-caslib-template.json
4 sas-admin cas servers policies define global-caslibs
  --attributes sessionTables:1000000000 --server serverA
  --source-file /path/global-caslib-template.json
5 sas-admin cas servers policies show-info --policy globalCaslibs --server serverA
```

- 1 Generate templates. Specify the folder location for the templates in the `output-location` option. The `policies-examples` folder is created, and one of the files in this folder is `globalCaslibs.json`.

Edit the `globalCaslibs.json` file for your environment. You can [add new policy options](#) or change values of existing policy options. Save the file.

- 2 Define a global caslib policy by specifying the edited `globalCaslibs.json` template file in the `source-file` option. The values in the `globalCaslibs.json` file are used to define the new policy. Information about the new policy is returned from the server after the policy is created. You will use this new policy to create a JSON template.



- 3 To create the template, redirect the JSON output for the new policy to a JSON file using the `servers policies show-info` command. Specify the use of JSON output with the `output json` global option. The policy name to use in the `policy` option was returned from the server when you created the policy in the previous step. Open the new JSON file to verify that it is correct.
- 4 Create a new global caslib policy using the new JSON template in the `source-file` option. On the command line, specify any new attributes or attributes to override the values in the template. In this example, a `sessionTables` value was specified. The value that you assign to the `sessionTables` attribute must be specified in bytes.
- 5 Display information about the new global caslib policy to verify the presence of the desired attributes and the additional `sessionTables` attribute.

## Priority Assignments Policies

---

**Note:** The groups to which you are assigning a priority level must exist in LDAP or as a custom group in order for the policy to work.

---

**Example:** Create a policy from a sample JSON template.

```
1 sas-admin cas generate-cas-samples --output-location sample-template-path
2 sas-admin cas servers policies define priority-assignments
  --server serverA --source-file /sample-template-path/policies-examples/
  priorityAssignments.json
```

- 1 Generate templates. Specify the folder location for the templates in the `output-location` option. The `policies-examples` folder is created, and one of the files inside this folder is `priorityAssignments.json`.

Edit the `priorityAssignments.json` file for your environment. You can [add new policy options](#) or change values of existing policy options. Save the file.

---

**Note:** As an alternative to manually editing the JSON file, you can add additional options by using the `ATTRIBUTES` option in the next step.

---

- 2 Define a priority assignments policy by specifying the edited `priorityAssignments.json` template file in the `source-file` option. The values in the `priorityAssignments.json` file are used to define the new policy. Information about the new policy is returned from the server after the policy is created. Make sure that the policy is correct.

**Example:** Create a JSON template for creating a priority assignment policy.

```
1 sas-admin cas generate-cas-samples --output-location sample-template-path
2 sas-admin cas servers policies define priority-assignments
  --server serverA --source-file /sample-template-path/policies-examples/
  priorityAssignments.json
3 sas-admin --output json cas servers policies show-info --policy priorityAssignments
  --server serverA > /path/priority-assignments-template.json
4 sas-admin cas servers policies define priority-assignments
  --attributes userM:1 --server serverA --source-file /path/priority-assignments-
  template.json
5 sas-admin cas servers policies show-info --policy priorityAssignments --server serverA
```

- 1 Generate templates. Specify the folder location for the templates in the `output-location` option. The `policies-examples` folder is created, and one of the files in the folder is `priorityAssignments.json`.

Edit the `priorityAssignments.json` file for your environment. You can [add new policy options](#) or change values of existing policy options. Save the file.

- 2 Define a priority assignments policy by specifying the edited `priorityAssignments.json` template file in the `source-file` option. The values in the `priorityAssignments.json` file are used to define the new policy. Information about the new policy is returned from the server after the policy is created. You use this new policy to create a JSON template.
- 3 To create the template, redirect the JSON output for the new policy to a JSON file using the `servers policies show-info` command. Specify that JSON output is used with the `output json` global option. The policy name to use in the `policy` option was returned from the server when you created the policy in the previous step. Open the new JSON file to verify that it is correct.
- 4 Create a new priority assignments policy using the new JSON template in the `source-file` option. On the command line, specify the users and their priority levels with the `attributes` option. In this example, `userM` is added to the policy with priority 1. The server returns information about the new policy after it is defined. Verify that `userM` is now included in the policy as well as the groups that were already in the source file.
- 5 Display information about the new priority assignments policy to verify the presence of the desired attributes and the additional user.

## Priority-Level Policies

---

### Note:

Priority-level policies can be used to assign resources based on a user's group membership or by explicit assignment with the priority assignments policy. To facilitate assigning resources by group membership, define custom groups with the same names as the priority-level policy and assign users to the groups.

---

**Example:** Create a policy from a sample JSON template.

```
1 sas-admin cas generate-cas-samples --output-location sample-template-path
2 sas-admin cas servers policies define priority-levels --server serverA --priority 3
--source-file /sample-template-path/policies-examples/cas-shared-default-priority-2.json
```

- 1 Generate templates. Specify the folder location for the templates in the `output-location` option. The `policies-examples` folder is created, and one of the files in the folder is `cas-shared-default-priority-2.json`.

Edit the `cas-shared-default-priority-2.json` file with values appropriate for your environment. You can [add new policy options](#) or change values of existing policy options. Save the file.

- 2 Define a priority-level policy by specifying the edited `cas-shared-default-priority-2.json` template file in the `source-file` option. Use the `priority` option to specify the priority for the policy. In this example, the new policy is priority 3. Information about the new policy is returned from the server after the policy is created. Make sure that the policy is correct.

**Example:** Create a JSON template for creating a priority-level policy.

```
1 sas-admin cas generate-cas-samples --output-location sample-template-path
2 sas-admin cas servers policies define priority-levels --priority 3
--server serverA
--source-file /sample-template-path/policies-examples/cas-shared-default-priority-2.json
3 sas-admin --output json cas servers policies show-info --policy serverA-priority-3
--server serverA > /path/priority-levels-template.json
4 sas-admin cas servers policies define priority-levels --priority 4 --cpu 30
--session-tables 2000000000 --server serverA
```

```
--source-file /path/priority-levels-template.json
```

- 1 Generate templates. Specify the folder location for the templates in the `output-location` option. The `policies-examples` folder is created, and one of the files in the folder is `cas-shared-default-priority-2.json`.

Edit the `cas-shared-default-priority-2.json` file with values appropriate for your environment. You can [add new policy options](#) or change values of existing policy options. Save the file.

- 2 Use the edited `cas-shared-default-priority-2.json` template file in the `source-file` option to define a new policy. You use this new policy to create a JSON template.

In this example, a policy for priority level 3 is specified with the `priority` option. Information about the new policy is returned by the server after the policy is created. The name of priority-level policies is generated as follows: `server-priority-priority`. Therefore, in this example, the policy is named `serverA-priority-3`.

- 3 To create the template, redirect the JSON output for the new policy to a JSON file using the `servers policies show-info` command. Specify the use of JSON output with the `output json` global option. Open the new JSON file to verify that the contents are correct.
- 4 Create a new policy using the new JSON template in the `source-file` option. This example shows a policy for priority level 4.

Specify the appropriate values for priority level 4. This example specifies different values for the CPU and `sessionTables` attributes with the `cpu` and `session-tables` options. The value that you assign to the `session-tables` option must be in bytes. The server returns information about the new policy after it is defined. Verify that the `serverA-priority-4` policy was created with the correct values for CPU and `sessionTables`.

---

**Note:** To enable the policy resource settings by group membership, create a custom group that has the same name as the priority-level policy. For example, if the policy is named `serverA-priority-4`, you must create a custom group with the same name. Then, add the users to the group who will be granted this priority level. These users will automatically be granted the resource settings that are specified in the priority-level policy by being a member of this custom group.

---

## See Also

[“CAS Resource Management Policies” in SAS Viya Administration: SAS Cloud Analytic Services](#)

---

## Source Files: Data

### Overview

For some CLI commands, you can pass data to the CLI by including it in a separate file using the `SOURCE-FILE` option. This file is referred to as a template or source file. You might want to use this option to enter multiple data items at the same time, rather than entering multiple CLI commands to enter data items.

## Examples

### Format Ranges

Templates that are used for adding ranges for new user-defined formats must be CSV files or JSON files. You can generate sample templates for adding ranges to user-defined formats with the cas CLI GENERATE-CAS-SAMPLES command.

Here is an example of a formats source file in CSV format:

```
F, female
M, male
f, female
m, male
```

### See Also

[“CLI Examples: Formats” in SAS Viya Administration: Data](#)

### CAS Server Paths Lists

Files that are used for adding paths to the paths list must be text files.

Here is an example of a paths list source file:

```
/opt/sas/viya/config/folderA
/opt/sas/viya/config/folderB
/opt/sas/viya/config/folderC
```

**Example:** Add multiple paths to the paths list for the specified CAS server using a template. This example assumes that you have a template text file that is formatted so that each path is on a separate line, as shown in the preceding source file.

```
1 sas-admin cas servers paths-list list --server serverA
2 sas-admin cas servers paths-list add-paths --server serverA
  --source-file /path_to_template/templateA.txt
3 sas-admin cas servers paths-list list --server serverA
```

- 1 Display the current paths list for serverA.
- 2 Add the paths in the template text file to the paths list for serverA.
- 3 Display the current paths list for serverA to confirm that the paths were added.

### Devices Blacklist

Files that are used for adding mobile devices to the blacklist must be text files.

Here is an example of a mobile devices source file:

```
deviceID4
deviceID5
deviceID6
```

**Example:** Add multiple devices to the blacklist using a template. This example assumes that you have a template text file that is formatted with each device on a separate line, as shown in the preceding source file.

```
sas-admin devices blacklist add --source-file /path_to_template/templateA .txt
```

---

# Command-Line Interface: Examples

---

## CLI Examples: CAS Administration

The following examples assume that you have already signed in to SAS Viya at the command line. See [“Command-Line Interface: Preliminary Instructions” on page 7](#).

**IMPORTANT** For Windows users of the CAS CLI who are using it for CAS administration and who are CAS administrators, CAS sessions are launched under the CAS service account. Also, any user who has credentials that are stored can use those credentials to run the CAS CLI for any purpose. See [“External Credentials: How To” in SAS Viya Administration: External Credentials](#).

## Generate CAS Samples

**Example:** Generate sample template files for creating caslibs, user-defined formats, and resource management policies.

```
sas-admin cas generate-cas-samples --output-location sample-location
```

**Note:** The `cas generate-cas-samples` command also generates an `md5.txt` file that contains checksums for each sample file. SAS Technical Support can use this file to validate the integrity of the sample files.

## See Also

[“Source Files: JSON Templates ” on page 22](#)

## Manage Tables

**Note:** These examples explicitly specify the `server` and `caslib` required options. However, using environment variables might be more efficient for these options. For more information about the environment variables, see [“Details” on page 31](#).

**Example:** For the specified caslib and CAS server, list all tables with names that contain the string `visual`, sort by `state`, and return a maximum number of 50 tables.

```
1 sas-admin cas help tables
2 sas-admin cas tables help list
3 sas-admin cas tables list --caslib caslibA --server serverA --name-contains visual
```

```
--sort-by state --limit 50
```

- 1 Review the Help for the `cas tables` command.
- 2 Review the Help for the `list` subcommand of the `cas tables` command.
- 3 Issue the command to list the tables for the specified caslib and CAS server with the appropriate subcommand and options.

**Example:** Load the given table for the specified caslib and CAS server.

```
sas-admin cas tables load --table airlines --server serverA --caslib caslibA
```

**Example:** Unload the given table for the specified caslib and CAS server.

```
sas-admin cas tables unload --table airlines --server serverA --caslib caslibA
```

**Example:** Show information about the given table for the specified caslib and CAS server.

```
sas-admin cas tables show-info --table airlines --server serverA --caslib caslibA
```

**Example:** Import and load a Microsoft Excel Open XML Spreadsheet as a table into the specified caslib and CAS server. The first row of the spreadsheet is a header row.

```
1 sas-admin cas tables import xlsx --caslib caslibA --server serverA --contains-header-
row
--source-file /path-to-file --table airlines
2 sas-admin cas list --caslib caslibA --server serverA
```

- 1 Import and load the spreadsheet as a table into the specified caslib and CAS server. Use the `source-file` option to specify the path to the spreadsheet file. Use the `contains-header-row` option to indicate that the first row of the spreadsheet is a header row.
- 2 Verify that the new table was added to the specified caslib and CAS server.

**Note:** The CAS plug-in to the `sas-admin` CLI does not verify that the type of table to import (specified on the command line following the `import` command) matches the file type of the file that is specified with the `source-file` option. The CAS plug-in attempts to process the file and later produces an unknown encoding error.

## Manage Caslibs

**IMPORTANT** The `caslibs` command prompts for the required options for the different types of caslibs that you can create. For information about the available options for each type of caslib, see the `dataSource` option for the `addCaslib` action here: [addCaslib Action](#). Be aware that there are more required options for the `cas caslibs` CLI command than there are for the `addCaslib` action set.

**Note:** These examples explicitly specify the `server` required option. However, using an environment variable might be more efficient for this option. For more information about the environment variables, see [“Details” on page 31](#).

**Example:** Generate sample template files for creating caslibs.

```
sas-admin cas generate-cas-samples --output-location /sample-template-path
```

**Example:** On the specified server, create a caslib that is based on a file path.

```
sas-admin cas caslibs create path --name caslibA --path /tmp/dept --server serverA
```

**Example:** On the specified server, create a caslib that is based on an instance of an Impala server.

```
sas-admin cas caslibs create impala --authentication-domain domain --server serverA
--name caslibA --schema schema --impala-server Impala-server
```

**Example:** On the specified server, create a caslib that is based on an instance of a Postgres server.

```
sas-admin cas caslibs create postgres --authentication-domain domain --server serverA
--name caslibA --postgres-server Postgres-server --postgres-database Postgres-
database
```

**Example:** On the specified server, list the first 20 global caslibs. You must be able to assume the Superuser role to run the command with elevated privileges.

```
sas-admin cas caslibs list --server serverA --scope global --limit 20 --superuser
```

**Example:** On the specified server, list the global caslibs starting at caslib number 21. You must be able to assume the Superuser role to run the command with elevated privileges.

```
sas-admin cas caslibs list --server serverA --scope global --start 21 --superuser
```

**Example:** Delete the given caslib from the specified server.

```
sas-admin cas caslibs delete --server serverA --name caslibA
```

## See Also

[“Create Caslibs from JSON Templates” on page 22](#)

## Details

- When running the CAS CLI on a UNIX machine or a Macintosh machine, with the CAS server running on a Windows machine, a Windows pathname that contains backslashes must be enclosed in single quotation marks. Here is an example: `'\\tmp\sas'`.
- The CAS CLI supports the following environment variables:
  - SAS\_CLI\_DEFAULT\_CAS\_SERVER
  - SAS\_CLI\_DEFAULT\_CASLIB
  - SAS\_CLI\_DEFAULT\_CAS\_SESSION

You can assign values to the environment variables that you want to remain in effect throughout your session. If the CAS CLI command requires the `server`, `caslib`, or `session-id` options, and the environment variables are set, then you can omit the required options from the CAS CLI command.

For example, suppose the following:

- **SAS\_CLI\_DEFAULT\_CAS\_SERVER** is set to `serverA`.
- **SAS\_CLI\_DEFAULT\_CASLIB** is set to `caslibA`.

You can then run this command without specifying the required `server` and `caslib` options: `./sas-admin cas tables show-info --table airlines`.

---

**Note:** Some commands do not support the use of some of the environment variables. For example, the CAS CLI ignores the CAS environment variables for the following commands:

- `caslib remove-control`

- `caslib delete`
- `tables remove-control`
- `sessions delete`

You must explicitly specify all required options when using these commands.

---

## See Also

- [“Command-Line Interface: Overview” on page 2](#)
- [SAS Viya Administration: Data](#)

---

## CLI Examples: Configuration

The following examples assume that you have already signed in to SAS Viya at the command line. See [“Command-Line Interface: Preliminary Instructions” on page 7](#).

### Examples

**Example:** List all the configurations that exist in the Configuration service.

```
sas-admin configuration configurations list
```

**Example:** Download the configurations with the specified definition name (`spring` in this example) from the SASLogon service, and write the output to a file.

```
sas-admin configuration configurations download --definition-name spring --service
SASLogon
--target path-to-output-file
```

**Example:** List the expectation objects in the Configuration service.

```
sas-admin configuration expectations list
```

**Example:** Show the expectation with the specified ID.

```
sas-admin configuration expectations show --id 185a046e-4e88-4e29-86cf-61d04b9abd07
```

### Details

- You can delete a configuration with the configuration CLI.

---

#### **CAUTION**

Do not use the `delete` subcommand of the `configurations` command unless you are sure that you want to delete your configuration.

---

## See Also

[“Command-Line Interface: Overview” on page 2](#)



## CLI Examples: Fonts

The following examples assume that you have already signed in to SAS Viya at the command line. See [“Command-Line Interface: Preliminary Instructions” on page 7](#).

### Examples

**Example:** Add and register the open-source OpenSans-Bold.ttf Google font using the `--uri` option. This option specifies the uniform resource indicator that indicates the fully qualified location of the font that is being added, including the file name. The `--uri` option path should point to the web server that hosts the font files.

```
sas-admin fonts add --uri https://server:port/directory/OpenSans-Bold.ttf
```

**Note:** In a Windows environment, to add a font with a file uniform resource indicator (URI), copy the file to the Windows server and then add the font to the environment with this command: `sas-admin fonts add --uri file://hostname/path-to-font`

**Example:** Add and register the Arial Narrow font using the `--source-file` option. This option specifies the fully qualified path of the font and adds the font directly.

```
sas-admin fonts add --source-file /path-to-font/ARIALN.ttf
```

**Example:** List the currently registered fonts in the system.

```
sas-admin fonts list
```

**Example:** Show information about the Arial Symbol font.

```
1 sas-admin fonts list --name "Arial Symbol"
2 sas-admin fonts show-info --id font-ID-of-Arial-Symbol-font
```

- 1 Identify the ID of the Arial Symbol font.
- 2 Show information about the Arial Symbol font using the specified font ID.

**Example:** Delete fontA from the system.

```
1 sas-admin fonts list --name fontA
2 sas-admin fonts delete --file-id file-id of fontA
```

- 1 Identify the file ID of fontA.
- 2 Delete fontA from the system using the specified file ID.

### Details

- The fonts CLI can be used to add web open font format (WOFF), TrueType (TTF), and TrueType Collection (TTC) fonts to the Fonts service. Once added, the fonts are available for the following purposes:
  - To render content in web browsers with web open font format (WOFF) and TrueType (TTF) fonts.

TrueType Collection (TTC) fonts are not displayed in web browsers. Therefore, users cannot select text that uses TrueType Collection (TTC) fonts if they are included in a SAS web application such as SAS Visual Analytics.

- To render contents for printing with TrueType (TTF) and TrueType Collection (TTC) fonts.  
Web open font format (WOFF) fonts are not supported for printing PDF or SVG output. Therefore, WOFF fonts must be paired with a matching TTF or TTC font for print support.

---

**Note:** The following SAS system fonts are WOFF only, and are not supported for printing:

- Noto Sans
- Noto Sans JP
- Noto Sans KR
- Noto Sans SC
- Noto Sans TC
- Noto Sans Thai

---

Given the TTC and WOFF font limitations, you might need to upload a combination of font formats in order to satisfy both of the preceding purposes.

- A font might not be available from a web server that is accessible from the middle tier without authentication. If so, you can upload the font to the SAS Viya server and reference it with the `file:///` URI scheme.
- In multi-tenant environments, the fonts are shared across all the tenants. Maintenance is supported only on the provider tenant.
- You are responsible for obtaining licensing of any fonts that are registered with the system. In a multi-tenancy configuration, this includes licensing the fonts for use by all tenants.

## See Also

[“Command-Line Interface: Overview” on page 2](#)

---

## CLI Examples: Inventory

The following examples assume that you have already signed in to SAS Viya at the command line. See [“Command-Line Interface: Preliminary Instructions” on page 7](#).

### Scan Content

**Example:** Scan and inventory the artifacts created by CAS. Specify a customer name and identifier for the deployment that is being inventoried. Specify a specific directory to write the results of the inventory. Display the name of each caslib and CAS table as it is processed.

```
sas-admin inventory scan cas --customer-id "customerName" --deployment-label "prod"
--output-location sample-location --show-details
```

**Example:** Scan and inventory the SAS Viya file system content that is used by SAS and the customer. Specify a customer name and identifier for the deployment that is being inventoried. Specify a specific directory to write the results of the inventory.

```
sas-admin inventory scan filesystem --customer-id "customerName" --deployment-label "prod"
--output-location sample-location
```

**Example:** Scan and inventory the infrastructure elements of your deployment. Specify a customer name and identifier for the deployment that is being inventoried.

```
sas-admin inventory scan infrastructure --customer-id "customerName" --deployment-label "prod"
--output-location sample-location
```

**Example:** Scan and inventory the artifacts created by SAS Viya services. Specify a customer name and identifier for the deployment that is being inventoried. Specify a specific directory to write the results of the inventory.

```
sas-admin inventory scan services --customer-id "customerName" --deployment-label "prod"
--output-location sample-location
```

---

**Note:** You provide the value for the deployment label to identify the deployment where scans are run. You also provide the value for the customer ID to identify the customer who is running scans. These are user-provided values.

---

**IMPORTANT** In order to merge the results of the scans with the PUBLISH command, each scan in a deployment must have the same customer ID and deployment label. To compare the results of multiple scans in the inventory report, each scan must use a different deployment label.

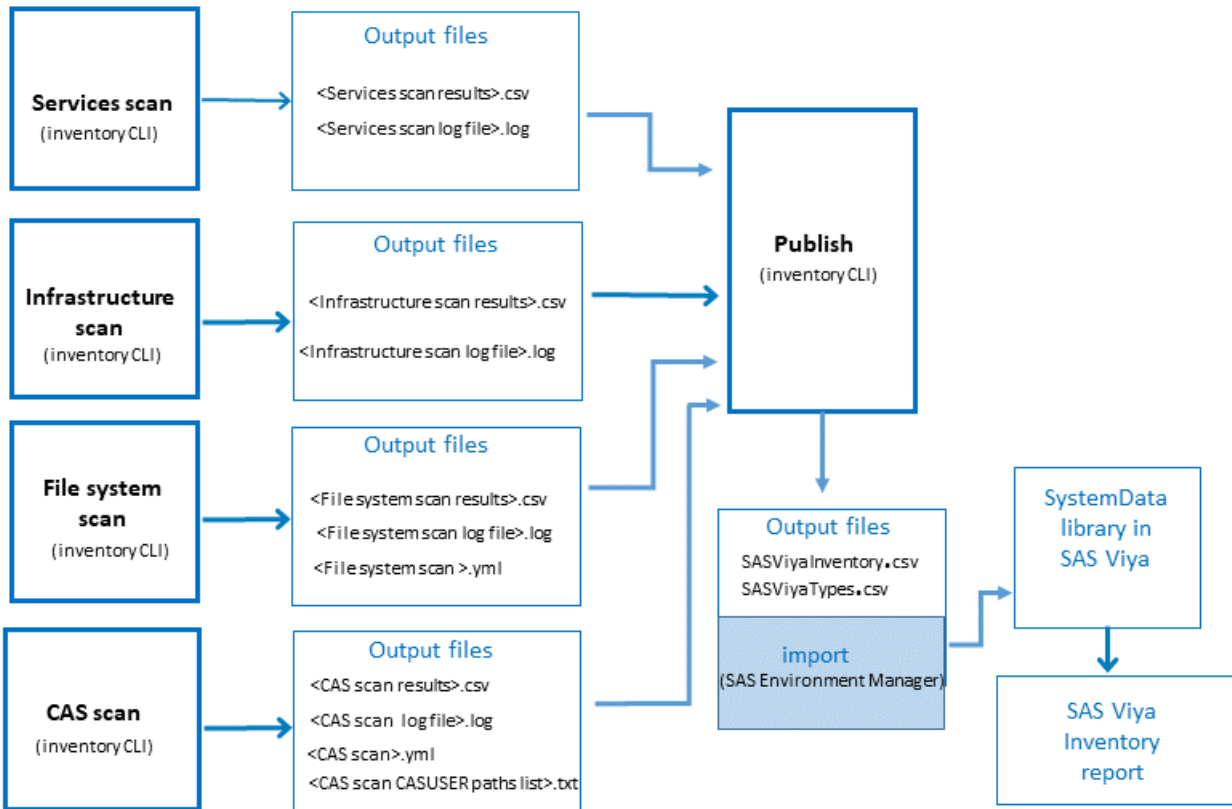
## Publish Results

**Example:** Merge the results of the scans into one CSV file. Specify a specific directory to write the CSV file. Specify the directory that contains the results of the various scans that will be merged.

```
sas-admin inventory publish --output-location sample-location
--source-location sample-location
```

## Details

Here is a workflow for the scanning and publishing process in SAS Viya:



## Scan

- To prepare for scanning, identify and note the machines that house the following:

---

Analytics projects<sup>1</sup>

---

CAS controller

---

Operations<sup>2</sup>

---

ModelServices<sup>2</sup>

---

User modified files<sup>3</sup>

---

SAS Configuration Server (Consul)  
leader

---

Primary PostgreSQL instance

---

Common Planning Service (CPS)  
primary PostgreSQL instance (if you  
have CPS PostgreSQL)

---

<sup>1</sup>Analytics projects are the artifacts created by the various SAS analytic solutions and can be located on any CAS controller in the environment.

<sup>2</sup>This is a host group in the inventory file.

<sup>3</sup>Any machine where a user modified (USERMODS) file was modified to affect the behavior of a SAS product should be scanned.

- The file system scan searches the following:

| Files Searched for by File System Scan | Description                 |
|--|-----------------------------|
| .sashdat                               | SASHDAT file                |
| .sas7bdat                              | SAS data set                |
| .sas7bcatalog                          | SAS catalog                 |
| .sas                                   | SAS program file            |
| .csv                                   | Comma-Separated Values File |
| USERMODS                               | SAS USERMODS file           |
| .astore                                | Analytic store file         |
| .p                                     | Scoring resource file       |
| .pkl                                   | Scoring resource file       |
| .pickle                                | Scoring resource file       |
| Audit.csv                              | Genaudit timeslice file     |
| AuditLastRecordFile                    | Genaudit record file        |

- Use the following as a guideline for running scans, considering your specific topology.

**Note:** In the scenarios below, the host account is either the cas user host account or the sas user host account. For the cas user account, the default user ID is cas. For the sas user account, the default user ID is sas. The SAS administrative user is an account that is defined in SAS Environment Manager as a member of the SAS Administrators users group who has superuser privileges.

□ Services scan

- Log on to any machine as the sas user host account.
- Authenticate to the sas-admin CLI as a SAS administrative user.
- Run only once per deployment.

□ CAS scan

- Log on to each CAS controller as the cas user host account.
- Authenticate to the sas-admin CLI as the SAS administrative user.

- File system scan
  - Log on to the machine that houses the analytics projects as the cas user host account. The default location for these projects is located on each CAS controller. Authenticate to the sas-admin CLI as the SAS administrative user.
  - Log on to the machine specified in the [Operations] host group of the inventory.ini file - as the sas user host account. If the cas user host account is a member of the same group as the sas user host account, then you can log on to the machine as the cas user host account. Authenticate to the sas-admin CLI as the SAS administrative user.
  - Log on to the machine specified in the [ModelService] host group of the inventory.ini file - as the sas user host account.. If the cas user host account is a member of the same group as the sas user host account, then you can log on to the machine as the cas user host account. Authenticate to the sas-admin CLI as the SAS administrative user.
  - Log on to each machine that houses a user modified file as the sas user host account. If the cas user host account is a member of the same group as the sas user host account, then you can log on to the machine as the cas user host account. Authenticate to the sas-admin CLI as the SAS administrative user.

**Note:** The file system scan in the table below will search each subdirectory under the path that is specified.

**Table 6** File System Scan Path Information for Linux

| Path being scanned  | Host account owner | Location                 | Description   |
|---|--------------------|--------------------------|---|
| <code>/opt/sas/viya/config/data/cas/default/projects</code> | cas                | CAS controller (default) | The location on disk for the transient analytics caslibs.                                   |
| <code>/opt/sas/viya/config/data/modelsvr</code>             | sas                | Services layer machine   | The default location on disk for the ASTOREs that are written by SAS Model Manager / TKMAS. |
| <code>/opt/sas/viya/config/data/mllearning</code>           | sas                | Services layer machine   | The default location on disk for the PKL files that are generated by Python models.         |
| <code>/opt/sas/viya/config/var/cache/auditcli</code>        |                    |                          | The default location on disk for the genaudit content.                                      |
| <code>/opt/sas/viya/config/etc</code>                       | sas                | Services layer machine   | The default configuration location. The scan  |

will search  
subfolders of this  
location for  
USERMODS files.

□ Infrastructure scan

- Log on to the machine that houses the SAS Configuration Server (Consul) leader as the sas user host account. Authenticate to the sas-admin CLI as the SAS administrative user. If only one machine is specified, run the scan on the machine specified in the [Consul] host group of the inventory.ini file. If multiple machines are specified, determine which machine houses the leader. See “[Consul Status Leader](#)” on page 49 for more information.
- Log on to the machine that houses the primary PostgreSQL instance as the sas user host account. Authenticate to the sas-admin CLI as the SAS administrative user.
- If you have CPS PostgreSQL, log on to the machine that houses the primary CPS PostgreSQL instance as the sas user host account. Authenticate to the sas-admin CLI as the SAS administrative user.

**Note:** To determine the primary PostgreSQL node, see “[Determine the Primary Data Node](#)” in *SAS Viya Administration: Infrastructure Servers*.

**Note:** The infrastructure scan in the table below will search each subdirectory under the path that is specified.

**Table 7** Infrastructure Scan Path Information for Linux

| Path being scanned   | Host account owner | Location   | Description                          |
|--|--------------------|--|--------------------------------------|
| <code>/opt/sas/viya/config/data/sasdatasvrc/postgres</code>    | sas                | PostgreSQL node  | The default PostgreSQL location.     |
| <code>/opt/sas/viya/config/data/sasdatasvrc/cpspostgres</code> | sas                | CPS PostgreSQL node                                    | The default CPS PostgreSQL location. |
| <code>/opt/sas/viya/config/data/consul</code>                  | sas                | SAS Configuration Service (Consul) leader <sup>1</sup> | The default Consul location.         |

<sup>1</sup>See “[Consul Status Leader](#)” on page 49 for instructions about how to determine the SAS Consul leader.

- Consider the scan scenario in the following sample topologies.

*Windows Topology*

When scanning a Windows machine, log on to the machine as a system administrative user. Log in to the sas-admin CLI as the SAS administrative user to run all scans.

### Linux Topology

**Note:** In the following scenarios, the host account is either the cas user host account or the sas user host account. For the cas user account, the default user ID is cas. For the sas user account, the default user ID is sas. The SAS administrative user is an account that is defined in SAS Environment Manager as a member of the SAS Administrators users group who has superuser privileges.

**Table 8** Scans on Multiple CAS Server Systems (Linux)

| Scan Type      | Machine on Which to Run Commands   | Account to Log On To the Machine As            | Account to Log On To the sas-admin CLI As |
|----------------|--|--|---|
| Services       | Any host   | sas user host account                          | SAS administrative user                   |
| Infrastructure | Consul leader node<br>PostgreSQL primary node <sup>1</sup>   | sas user host account                          | SAS administrative user                   |
| File system    | Operations host<br>ModelServices host<br>Each CAS controller <sup>2,3,4</sup><br>Additional machines as necessary <sup>5</sup> | sas user host account<br>cas user host account | SAS administrative user                   |
| CAS            | CAS controller <sup>2,3,4</sup>  | cas user host account                          | SAS administrative user                   |

<sup>1</sup>If the SAS Configuration Server (Consul) and SAS Infrastructure Data Server (PostgreSQL) are on the same machine then the infrastructure scan can be run on the service layer machine. In a multiple machine deployment, there can be more than one machine where you would need to run the infrastructure scan.

<sup>2</sup>If the CASUSER caslibs are not owned by the cas user host account, then you might need to run the command from an account with higher privileges.

<sup>3</sup>If there are multiple CAS controllers, you must run the file system and CAS scans on each one.

<sup>4</sup>In the case of a single machine deployment, if the sas user host account and cas user host account are in the same group, then either of them can be used to run the CLI for the file system and cas scans.



<sup>5</sup>If you specify a text file with additional paths to search, you should perform the scan on the machines that host those paths.

### Multi-Tenant Topology

See “Multi-Tenancy Deployment Scans” on page 47.

- Each type of scan produces the following sets of output files:

---

**Note:** It is recommended that you use the same output location for all of the scans.

---

**Table 9** Files Produced by Scans

| Type of Scanned Content | Files Produced  | Description   |
|-------------------------|---|---|
| CAS                     | <code>SASViyaInventory_CAS_deployment-label_short-host-name.csv</code>                    | CSV file that contains results of CAS scan.         |
|                         | <code>SASViyaInventory_CAS_deployment-label_short-host-name_time-stamp.log</code>         | Log file of CAS scan.                               |
|                         | <code>SASViyaInventory_CAS_deployment-label_short-host-name_CASUSER_Paths.txt</code>      | Text file that contains paths of CASUSER libraries. |
|                         | <code>SASViyaInventory_CAS_deployment-label_short-host-name_Caslibs.yml</code>            | YML output file.                                    |
| File system             | <code>SASViyaInventory_File_system_deployment-label_short-host-name.csv</code>            | CSV file that contains results of file system scan. |
|                         | <code>SASViyaInventory_File_system_deployment-label_short-host-name_time-stamp.log</code> | Log file of file system scan.                       |
|                         | <code>SASViyaInventory_File_system_deployment-label_short-host-name_Paths.yml</code>      | YML output file                                     |
| Services                | <code>SASViyaInventory_Services_deployment-label.csv</code>                               | CSV file that contains results of services scan.    |

| Type of Scanned Content | Files Produced   | Description   |
|-------------------------|--|---|
|                         | <code>SASViyaInventory_Services_deployment-label_time-stamp.log</code>                       | Log file of services scan.                            |
| Infrastructure          | <code>SASViyaInventory_Infrastructure_deployment-label_short-host-name.csv</code>            | CSV file that contains results of infrastructure scan |
|                         | <code>SASViyaInventory_Infrastructure_deployment-label_short-host-name_time-stamp.log</code> | Log file of infrastructure scan.                      |

See “[Workflow for a Multiple CAS Server Scan](#)” on page 50 for an example of the output files created.

## Publish

- The publish command of the inventory plug-in to the sas-admin CLI produces two CSV files. Prior to using the SAS Viya Inventory report in SAS Environment Manager, you must import these files into the SystemData library. See “[Importing Data from the Available Tab or Data Sources Tab](#)” in *SAS Data Explorer: User’s Guide* for more information.

The files produced by the publish command are as follows:

---

SASViyaInventory.csv

---

SASViyaTypes.csv

---

## Report

The SAS Viya Inventory report contains merged information from the various scans that are run by the inventory plug-in to the sas-admin CLI. The report is only available to SAS administrators. See “[Working with System Reports](#)” in *SAS Viya Administration: Monitoring* for more information about working with predefined reports from SAS Environment Manager.


When you open the report, the deployments for which an inventory was run are listed at the top. Select a deployment to display a report.

Select a report page to view. The report pages are represented by these tabs:

### Status


Displays the number of content items and metadata objects found during scans.

### What’s Being Used

Displays an overview of the software usage by either SAS product totals (Content Items Found) or more specific totals of SAS objects (**Types of Items Found**), SAS servers (**Server, Providers and Contexts**), and SAS storage (**Data Item Storage**). From each chart, select  for additional


options. From the **Content Items Found** chart, click on a bar to drill down and populate the other charts with information specific to that item.

### CAS Inventory

Explore the types of CAS libraries found and the tables within them. From each chart, select  for additional options. For **Library type**, select the caslib type for which you want to display information. From the **Largest CAS Libraries** chart, click on a bar to drill down and populate the other charts with information specific to that item.


Place your cursor over a line in a graph to view detailed values.

### CAS Controls Inventory

Drill into the CAS access controls associated with the CAS libraries and tables that were found. For **Choose an Identity**, select the identity for which you want to view the access controls. From each chart, select  for additional options. For **Choose a Library**, select the caslib for which you want to view the access controls. From the **Top Library Controls** chart, click on a bar to drill down and populate the other charts with information specific to that item.


Place your cursor over a line in a graph to view detailed values.

### Filesystem Inventory

Explore the file types that exist on your system and their respective file sizes to determine file system consumption. From the bar below the **Filesystem Inventory** tab, select the area of the file system for which you want to display the details. From the bar below the file system area bar, select the file extension (within the file system area) for which you want to display the details. From each chart, select  for additional options. From the **Largest filesystem areas** chart, click on a bar to drill down and populate the other charts with information specific to that item.


Place your cursor over a line in a graph to view detailed values.

### Analytics Projects Inventory

Drill into the Analytics Suites' project files and their file sizes to discover file system consumption. For **Analytics Project Type**, select the type of analytics project for which you want to view details. From each chart, select  for additional options. From the **Largest filesystem areas** and **Largest projects** charts, click on a bar to drill down and populate the other charts with information specific to that item.


Place your cursor over a line in a graph to view detailed values.

### Object Inventory

Explore the product usage through the count of each object type created and the services that created them. From the bar below the **Object Inventory** tab, select the product portfolio for which you want to display the details. From each chart, select  for additional options. From the **Top Service Counts** and **Top Item Counts** charts, click on a bar to drill down and populate the other charts with information specific to that item.

Place your cursor over a line in a graph to view detailed values.

### Type Navigator

Offers a quick visual of which SAS products are being used and the object types created by those products. From each chart, select  for additional options. From the **Frequency of Portfolio** and **Frequency of Sellable Offering** charts, click on a bar to drill down and populate the other charts with information specific to that item. Clear filters from the **Filters** tab.

Place your cursor over a line in a graph to view detailed values.

### New System Provisioning

Determine the amount of file system space needed to back up your SAS content.

### Compare Two Environments

Compare the difference between two different deployments or two different executions of the same deployment. From the bar below the **Compare Two Environments** tab, select the

deployment labels of the two deployments that you want to compare. From the top left of the window below the deployment label bar, select the type of scan information that you want to compare. From the resulting chart (**CAS Items**, **Filesystem Items**, **Content Objects**, or **Infrastructure**), click on a bar to drill down and populate the other charts with information specific to that item.

Place your cursor over a line in a graph to view detailed values.

---

## CLI Examples: Reports

The following examples assume that you have already signed in to SAS Viya at the command line. See [“Command-Line Interface: Preliminary Instructions” on page 7](#).

### Single Command Examples

**Example:** Show information about the report that has the ID a85235e7-fad1-4f8a-9ad9-ea0d576619e1.

```
sas-admin reports show-info --id a85235e7-fad1-4f8a-9ad9-ea0d576619e1
```

**Example:** List the detailed output of all reports that were created after 2017-05-23.

```
sas-admin reports list --created-after 2017-05-23 --details
```

**Example:** List the reports in the SAS Viya system that were modified by user1.

```
sas-admin reports list --modified-by user1
```

**Example:** List a maximum of 50 reports that are sorted by ID and in descending order.

```
sas-admin reports list --details --sort-by ~id --limit 50
```

**Example:** Delete the report that has the ID a85235e7-fad1-4f8a-9ad9-ea0d576619e1.

```
sas-admin reports delete --id a85235e7-fad1-4f8a-9ad9-ea0d576619e1
```

**Example:** Delete the report that has the ID a85235e7-fad1-4f8a-9ad9-ea0d576619e1 without prompting the user for confirmation.

```
sas-admin reports delete --id a85235e7-fad1-4f8a-9ad9-ea0d576619e1 --force
```

### Multiple Command Examples

---

**Note:** Some of these tasks must be performed by a user with administrative privileges.

---

**Example:** Update the theme that is used by a SAS Visual Analytics report. This example assumes that you already have a SAS Visual Analytics report that is using the Marine theme (the default).

```
1 sas-admin reports themes list
2 sas-admin reports themes show-info --theme-id theme_ID
3 sas-admin reports list --name reportA
4 sas-admin reports themes update --report-id report_ID --theme-id theme_ID
```

- 1 List the themes that are available for SAS Visual Analytics reports, and record the ID of the theme that you want to use.

- 2 Show detailed information about the identified theme for updating the report. You use the ID of the theme that was identified in the previous step.
- 3 List information about reportA that you want to update, and record the ID of the report.
- 4 Update the theme that is used in reportA to the theme that you identified in step 1. You use the ID of the report that you identified in the previous step.

---

**Note:** You can update a report's theme only to a theme of type `system` or `custom`. Themes of types `legacy` and `retired` are allowed in existing reports, but cannot be used to update a theme.

---

**Example:** Export the translation worksheets for reports that need to be translated, translate the worksheets, and then import the translated worksheets into the report. Some of these tasks must be performed by a user with administrative privileges whereas the actual translation might be performed by another user.

You need to determine what reports need to be translated and what languages that the reports need to be translated into. You also need to identify the base language that the reports were created in. The translation worksheets that are exported will contain strings to translate, and these strings are written in the base language that the report was created in. You must save the translation worksheets using the UTF-8 character encoding.

For this example, you learn that you need to translate the reports that were created after 01MAY2018 and whose names contain the text "VAN". You also learn that the reports need to be translated into French, Spanish, and Japanese. You determine that the base language that the reports were written in is English. See ["Details" on page 46](#) for additional information about translation worksheets.

```

1 sas-admin reports list --created-after 2018-05-01 --limit "100" --name-contains VAN
2 sas-admin reports translations export --report-id report_ID
  --output-location /output-path-for-worksheet --report-locale fr-FR
3 sas-admin reports translations export --report-id report_ID
  --output-location /output-path-for-worksheet --report-locale es-ES
4 sas-admin reports translations export --report-id report_ID
  --output-location /output-path-for-worksheet --report-locale ja-JP

5 ##### LOCALE FOR THIS TRANSLATION WORKSHEET #####
  fr-FR
##### BEGIN TRANSLATABLE STRINGS #####
property.label = translated value
property1.label = translated value
property2.label = translated value

6 sas-admin reports translations import
  --source-file /output-path-for-worksheet/report-name_fr-FR.reports
7 sas-admin reports translations import
  --source-file /output-path-for-worksheet/report-name_es-ES.reports
8 sas-admin reports translations import
  --source-file /output-path-for-worksheet/report-name_ja-JP.reports

```

- 1 List the reports that were created after 01MAY2018 and whose names contain the text "VAN". Use the `limit` option to specify the maximum number of reports to list. The default value is 20. To make sure that you list all the reports that were created after 01MAY2018, specify a value of 100. If you receive a message at the end of the report list that indicates that more reports are available, list the reports again with a higher value for the `limit` option.

After running the code to list the reports, you find that there are four reports that you need to translate. In other words, there are four reports that were created after 01MAY2018 and whose

names contain the text “VAN”. Note the report ID of each report. You will need the report ID to export the translation worksheets.

If you do not have administrative privileges, you might be able to obtain the report ID from SAS Environment Manager.

- 2 Export the translation worksheet for the first of the four reports in French (fr-FR),
- 3 Export the translation worksheet for the first of the four reports in Spanish (es-ES).
- 4 Export the translation worksheet for the first of the four reports in Japanese (ja-JP).

Repeats steps 2, 3, and 4 for the second, third, and fourth reports. Be sure to update the report ID of the report in the command when you move to the second, third, and fourth report. You will issue the command a total of 12 times in this example. (There are four reports, and each report is translated into three languages). Afterward, you should have 12 translation worksheets in the location that is specified in the output location option.

---

**Note:** If another user is performing the translation, then provide them with the translation worksheets. They can proceed with step 5.

---

- 5 Here is an example of the lines that exist in each translation worksheet. The line following `LOCALE FOR THIS TRANSLATION WORKSHEET` indicates the language of the translation worksheet.

In the French translation worksheets (*report\_name\_fr-FR*), locate the line that contains `BEGIN TRANSLATABLE STRINGS`. In the following lines, edit the values to the right of the equal sign (=) and provide the appropriate values, in French. Save the file.

In the Spanish translation worksheets (*report\_name\_es-ES*), locate the line that contains `BEGIN TRANSLATABLE STRINGS`. In the following lines, edit the values to the right of the equal sign (=) and provide the appropriate values, in Spanish. Save the file.

In the Japanese translation worksheets (*report\_name\_ja-JP*), locate the line that contains `BEGIN TRANSLATABLE STRINGS`. In the following lines, edit the values to the right of the equal sign (=) and provide the appropriate values, in Japanese. Save the file.

---

**Note:** If the translation was performed by another user, then this user must provide the translated worksheets to the user who is performing the import.

---

- 6 Import the French translation worksheet (*report\_name\_fr-FR*).
- 7 Import the Spanish translation worksheet (*report\_name\_es-ES*).
- 8 Import the Japanese translation worksheet (*report\_name\_ja-JP*).

## Details

- The `translations export` command appends the locale value that is specified as the report locale to the name of the translation report. For example, if you export a translation worksheet for reportA and specify the US English locale as the value of the `report-locale` option, the translation worksheet will be saved in a file with the name:

```
reportA_en-US.reports
```

The `translations export` command includes a locale section in the translation worksheet that includes the locale name that you specified as the report locale. Here is an example of the locale section for the locale en-US:

```
##### LOCALE FOR THIS TRANSLATION WORKSHEET #####
```

en-US

---

**Note:** Before importing a translation worksheet, you must make sure that the language in the locale section of the worksheet matches the language in the name of the translation worksheet. The name of the file that you are importing must be appended with a locale value.

---

- The character encoding of the files that contain the translation worksheets must be UTF-8.
- The `clear-results-cache` option clears the results cache of report data for SAS Visual Analytics reports in SAS Viya. In multi-tenancy environments, this command can be run only on the provider tenant, but it clears the results cache across all tenants.
- The reports CLI lists only 20 reports at a time by default. To list more than 20 reports, you can use the `--limit` option. To list 50 reports, enter `--limit 50`.
- To specify what report number to start the list with, you can use the `--start` option. Suppose that you have listed the first 20 reports, and you want to list the next 20 reports, starting with report number 21, enter `--start 20`.

## See Also

[“Command-Line Interface: Overview” on page 2](#)

---

# Command-Line Interface: Reference

---

## Inventory CLI

### Multi-Tenancy Deployment Scans

Consider the following scenario when using the inventory plug-in to the sas-admin CLI to scan a multi-tenancy deployment:

**Table 10** Scans in Multi-Tenant Deployments

| Scan Type | Machine                    | Provider   | Tenant  |
|-----------|----------------------------|--|---|
| Services  | Main service layer machine | <ol style="list-style-type: none"> <li>1 Log on to the provider service layer machine as the provider host account administrative user.</li> <li>2 Switch to the sas user host account.</li> </ol> | <ol style="list-style-type: none"> <li>1 Log on to the tenant service layer machine as the tenant host account administrative user.</li> <li>2 On the tenant service layer machine, list the CAS servers to determine the CAS controller.</li> <li>3 Log on to the tenant controller as the tenant host account system administrator.</li> <li>4 Log in to the sas-admin CLI as the SAS administrator (sasprovider), using the tenant service layer machine as the endpoint.</li> </ol> |

| Scan Type      | Machine                    | Provider   | Tenant   |
|----------------|----------------------------|--|--|
|                |                            | <ol style="list-style-type: none"> <li>3 Log in to the sas-admin CLI as the provider SAS administrator (sasboot).</li> <li>4 Run the services scan with the inventory plug-in to the sas-admin CLI.</li> </ol>   | <ol style="list-style-type: none"> <li>5 Run the services scan with the inventory plug-in to the sas-admin CLI.</li> </ol>   |
| Infrastructure | Main service layer machine | <ol style="list-style-type: none"> <li>1 Log on to the provider service layer machine as the provider host account administrative user.</li> <li>2 Switch to the sas user host account.</li> <li>3 Log in to the sas-admin CLI as the provider SAS administrator (sasboot).</li> <li>4 Run the infrastructure scan with the inventory plug-in to the sas-admin CLI.</li> </ol>   | Not applicable   |
| CAS            | CAS controller             | <ol style="list-style-type: none"> <li>1 Log on to the provider service layer machine as the provider host account administrative user.</li> <li>2 On the provider service layer machine, list the CAS servers to determine the CAS controller.</li> <li>3 Log on to the provider's CAS controller as the owner of the files. The provider needs to be able to see files in this location: /opt/sas/viya/home/config.</li> </ol> | <ol style="list-style-type: none"> <li>1 Log on to the tenant service layer machine as the tenant host account administrative user.</li> <li>2 On the tenant service layer machine, list the CAS servers for the tenant to determine the CAS controller.</li> <li>3 Log on to the tenant controller as the tenant host account administrator.</li> <li>4 Log in to the sas-admin CLI as the SAS administrator (sasprovider), using the tenant service layer machine as the endpoint.</li> <li>5 Run the CAS scan with the inventory plug-in to the sas-admin CLI with the TENANT-ID option.</li> </ol> |



| Scan Type   | Machine                    | Provider  | Tenant   |
|---|----------------------------|---|--|
|   |                            | <ol style="list-style-type: none"> <li>Run the CAS scan with the inventory plug-in to the sas-admin CLI with the TENANT-ID option.</li> </ol>   |  |
| <p>File system</p> <p><b>Note:</b> In multi-tenant deployments, run the services and CAS scans with the inventory plug-in to the sas-admin CLI before running the file system scan.</p> | Main service layer machine | <ol style="list-style-type: none"> <li>Log on to the provider service layer machine as the provider host account administrative user.</li> <li>Switch to the sas user host account.</li> <li>Log in to the sas-admin CLI as the provider SAS administrator (sasboot).</li> <li>Run the file system scan with the inventory plug-in to the sas-admin CLI with the TENANT-ID option.</li> </ol> | <p><b>Note:</b> File paths in multi-tenant environments are in <code>/opt/sas/&lt;tenantname&gt;</code><sup>1</sup></p> <ol style="list-style-type: none"> <li>Create a paths list text file that includes the paths in the <code>SASViyaInventory_CAS_deployment-label_CASUSER_Paths.txt</code> file (that resulted from the CAS scan), as well as the tenant-specific paths.</li> <li>Log on to the tenant service layer machine as the tenant host account administrative user.</li> <li>Log in to the sas-admin CLI as the SAS administrator (sasprovider), using the tenant service layer machine as the endpoint.</li> <li>Run the file system scan with the inventory plug-in to the sas-admin CLI, and provide the paths list text file on the command-line with the <code>PATHS-LIST-FILE</code> option.</li> </ol> |

<sup>1</sup>File paths specific to tenants include:

---

```
/opt/sas/tenantname/config/data/cas/default/projects
```

---

```
/opt/sas/tenantname/config/data/cas/default/casuserlibraries
```

---

```
/opt/sas/tenantname/config/data/modelsvr
```

---

```
/opt/sas/tenantname/config/etc
```

---

```
/opt/sas/tenantname/config/var/cache/auditcli
```

---

## Consul Status Leader

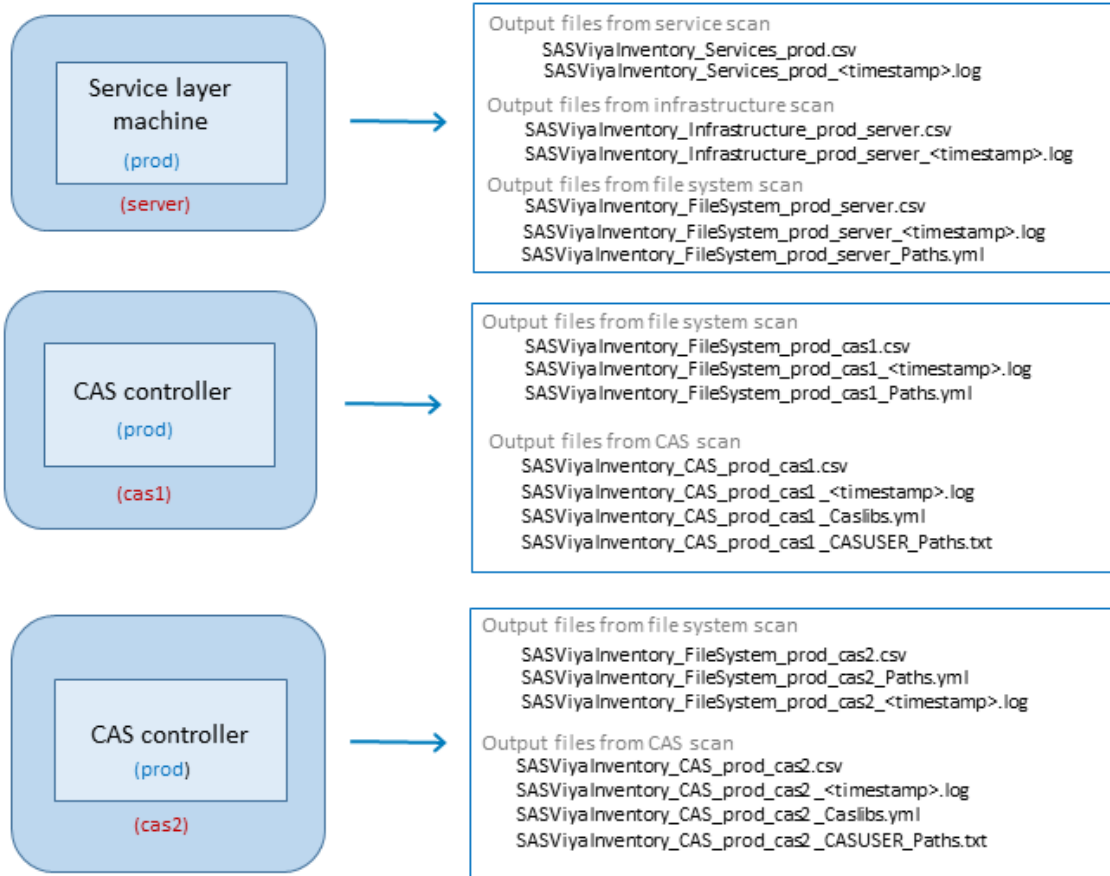
When using the inventory plug-in to the sas-admin CLI to perform an infrastructure scan, run the following commands to determine the Consul status leader:

```
sudo su sas
source /opt/sas/viya/config/consul.conf
export CONSUL_HTTP_TOKEN=$(cat
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token)
```

```
/opt/sas/viya/home/bin/sas-bootstrap-config status leader
```

## Workflow for a Multiple CAS Server Scan

Here is a workflow for using the inventory plug-in to the sas-admin CLI to scan a multiple CAS server environment:



Here are details about the preceding figure:

- The blue text denotes a deployment label.
- The red text denotes the short host name.

## Command-Line Interface: Troubleshooting

Message: token expired and refresh token is not set

Explanation: You are not currently authenticated to SAS Viya at the command line. See [“Command-Line Interface: Preliminary Instructions”](#) on page 7.

**Message:** flag provided but not defined

**Explanation:** You might have specified a global option in the wrong location. See [“Command-Line Interface: Syntax” on page 14](#).

