

Command-Line Administration in SAS® Viya®

Danny Hamrick, SAS

ABSTRACT

Administration of the SAS Viya environment gives users more flexibility with the `sas-admin` command-line utility. You can manage custom identity groups, create folder structures, or update the configuration, all from the command line. It is well-suited for scripting common changes in this complex environment. There are even capabilities that enable content promotion for building development and test environments. The command-line utility has the following benefits:

- It is automatically plannable as part of the server installation process.
- You can download it from the web on Red Hat Linux, Microsoft Windows, and even Apple Mac.
- It is deployed as a stand-alone executable—there's no require installation run-time environment as for Java.
- There is profile support, so you can manage multiple environments from one location.

INTRODUCTION

SAS has used several methods of administering its software over the years. At first, there was little need for application administration. Base SAS® was administered manually by editing configuration files. With the introduction of the SAS® Metadata Server, the information in the configuration files was integrated with metadata. Manually editing configuration files was no longer an option. To manage the configuration information in metadata, we created SAS® Management Console. It was robust, flexible, and worked well. However, because it was a user interface (UI), there was no way to automate configuration changes in batch or at a command line. We satisfied that need by creating the SAS Platform Object Framework batch utilities. These tools worked pretty well for consumers, but caused quite a few problems for SAS development. Each of the tools had its own custom code to handle similar requirements. There was little code reuse, which made feature enhancements more time consuming and testing more arduous.

For SAS Viya, one of the major architecture decisions was to use RESTful interfaces for all operations. The capabilities of each component are made available through these interfaces as a matter of design. We also made the decision to put the functionality of each interface in individual services, so that client applications would all have access to the same feature set. With this approach, we are now able to provide a more robust set of clients exclusively for the command line.

The first implementation available to customers is the SAS administration command line interface (CLI) or **sas-admin**. This command supports well over a dozen plug-ins for performing different administrative tasks such as security, configuration, and interacting with CAS management facilities. This paper discusses the capabilities and usage of the `sas-admin` command.

COMMAND LINE DESIGN FROM THE GROUND UP

The SAS Viya CLIs were designed to provide both a common feel and to be very flexible. A new framework was developed so that each CLI would have the same basic functionality. Some of those common features are:

- The framework is written in a language called GoLang. A major reason for this choice is that GoLang allows SAS to create a single executable that does not have a prerequisite of a run-time library. This is a big advantage over languages such as Java or Python. GoLang also has a portable compiler, which allows SAS to produce CLIs for Windows, Linux, and OSX (yes, SAS CLIs support the Apple Mac).
- The framework provides the CLIs with a common profile mechanism that is used for connecting to the SAS Viya system. You provide the URL of the web server for the specific environment that you want to access, along with some additional information to save about the interaction. There is also a common way to authenticate. The authentication will expire based on your configuration and will have to be renewed, but the profile information does not have to be resupplied.
- The framework provides flexibility. SAS provides multiple installation techniques so that you can install the CLI wherever you need it. We also provide a plug-in based architecture, which provides the ability to enhance the functionality of the CLI by simply installing more plug-ins. There are two types of installation techniques: the out-of-the-box server installation and the more ad hoc client installation.

MULTIPLE DEPLOYMENT STRATEGIES

The **server installation** is RPM-based, and is part of the default installation scripts. In the default Ansible playbook file called `inventory.ini`, you will see a group called `[CommandLine]`. This group defines the machines where the administration CLI and its plug-ins will be installed. This group does not need to be installed on all machines. Installing it on one machine that has access to a web server host and port is sufficient. In fact, you can even install this group on its own machine if you choose. Updates for this installation are provided as RPM updates.

To use the CLI in a server installation, you must SSH onto the installed box. On a Linux system, the command is installed in `/opt/sas/viya/home/bin`. You can either change directory to this path or add this to your `PATH` environment variable. All the plug-ins are automatically installed.

Note: The manual plug-in install described below is disabled for server installations.

The **client installation** is quite different, because it is more like a download model. This approach allows you to install the administration CLI on one or more administration machines. This also works well when your SAS Viya installation is not RPM-based, such as in Cloud Foundry installations. For this installation, you would start by downloading the `sas-admin` CLI from the SAS Viya download site, located at <http://supportprod.unx.sas.com/downloads/package.htm?pid=2133>. See Figure 1.

SAS Administration CLI

The SAS Administration Command Line Interface (CLI) allows you to perform administrative tasks like changing the configuration or updating security rules at the command line. **It can only be used with SAS Viya.** The SAS Administration CLI can run on Linux, Windows or OSX.

Requirements

- The CLI requires access to your SAS Viya system.
- The CLI requires that the machine running the tool has access to the internet.

SAS Administration CLI					
Platform	Description	Request Download	Size	Release Date	ReadME
Linux 64-bit	sas-admin for Linux	sas-admin-cli-1.1.3-download-linux.tgz	2.5 MB	2018-02	txt
Mac	sas-admin for OSX	sas-admin-cli-1.1.3-download-osx.zip	2.7 MB	2018-02	txt
Windows	sas-admin for Windows	sas-admin-cli-1.1.3-download-windows.zip	2.5 MB	2018-02	txt

Figure 1: SAS Administration CLI Download Site

Download and unzip the archive that matches your operating system. Follow the instructions provided in the ReadME link. After you download and unzip the CLI, you can manually search for and install the plug-ins that you need. The ReadME shows examples of the plug-in install commands. Figure 2 illustrates the output of a plug-in install command.

```
$ ./sas-admin plugins install --repo SAS authorization
Installing the authorization plugin...
Retrieving source file from the SAS repository...
2285660 bytes downloaded.
Saving authorization plugin to /Users/me/.sas/admin-plugins/sas-authorization-cli-1.2.0-
20180131.1517421124...
Plugin authorization successfully installed.

$ ./sas-admin plugins list
Name          Version      Description
authorization 1.2.0       Plugin that creates and manages SAS authorization rules
```

Figure 2: Plug-in Install Command

USAGE IS ALWAYS EASY... WITH ENOUGH HELP

After you've installed your CLI and plug-ins, you should investigate the syntax. You can find the documentation for all the commands and options in the online SAS Viya Administration guide, located at

<http://documentation.sas.com/?docsetId=calcli&docsetTarget=titlepage.htm&docsetVersion=3.3>

However, all commands have built-in Help documentation. After you are familiar with the CLI, the Help makes it easy to figure out the capabilities and required options of each command. Figure 3 is an example of one of the Help screens.

```
$ ./sas-admin --help
NAME:
  sas-admin - SAS Administrative Command Line Interface

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

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:
  authorization
  folders
  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, $LANG]
  --output           Specifies output format - text, json, fulljson. [$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) 2016-2018 SAS Institute Inc. All Rights Reserved.
```

Figure 3: Command Help

This Help screen provides a syntax for the command line, a list of commands and options, and a list of installed plug-ins. In the syntax, the plug-ins are used as additional commands. Figure 4 shows the Help for the authorization plug-in.

```
$ ./sas-admin authorization --help
NAME:
  sas-authorization

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

COMMANDS:
  authorize, grant, create-rule    Creates an authorization rule to grant privileges to
the specified principal.
  create-rules                    Create a set of authorization rules.
  explain                         Shows the explanations of a target object URI. For example:
--target-uri /SASHome/**
  help, h                         Shows a list of commands or help for one command.
  list-rules                      Lists the authorization rules that are defined for the SAS
environment.
  prohibit                        Creates an authorization rule to prohibit privileges for the
specified principal.
  remove-rule, revoke            Removes a specific authorization rule.
  remove-rules                  Removes a set of authorization rules.
  show-rule                     Shows a specific authorization rule.
  update-rule                    Modifies a specified authorization rule.
```

Figure 4: Authorization Plug-in Help

BUT REALLY, EXAMPLES ARE WHAT YOU WANT

Here are several examples of the different use cases that the SAS Administration CLI is intended to address. Note that these examples assume you have either the server or the client installation and that you are in the directory for sas-admin.

STANDARD PROFILE CREATION

Figure 5 shows the command for updating the default profile. You can have multiple profiles using the `--profile` global option. See the documentation for additional details.

```
$ ./sas-admin prof init
Enter configuration options:

Service Endpoint> http://mymachine:7980

Output type (text|json|fulljson)> text

Enable ANSI colored output (y/n)?> y
Saved 'Default' profile to /Users/me/.sas/config.json.

$ ./sas-admin auth login
Enter credentials for http://mymachine:7980:

Userid> sasboot

Password>
Login succeeded. Token saved.
```

Figure 5: Command for Updating the Default Profile

All the following examples assume a default profile and a valid authentication.

NAVIGATING FOLDERS

The folders plug-in interacts with the logical folder mechanism in SAS Viya. This is the same information provided in the Content view of SAS® Environment Manager. Figure 6 shows abbreviated output from the list subcommand.

```
./sas-admin folders list
Id          Name                Type                ParentFolderUri
bc2<snip>700f  sas.dataMiningModels userFolder          /folders/folders/21<snip>d3cb
868<snip>db8b  Public              folder              <nil>
070<snip>ed3f  Images              hiddenFolder       /folders/folders/b32<snip>490
751<snip>b293  My Folder           myFolder           /folders/folders/03f<snip>174b
b32<snip>c490  Custom Themes       hiddenFolder       /folders/folders/0f9<snip>14e7
f9e<snip>9213  My History          historyFolder      /folders/folders/03f<snip>174b
```

Figure 6: List Folders

The folder plug-in can show you members of a folder or output the entire hierarchy of a specified parent folder, as shown in Figure 7.

```
$ sas-admin folders list-members --id 3e2d548d-fce5-4374-862c-920686e8f26a --tree
|— Products
|  |— SAS Environment Manager

$ sas-admin folders list-members --id 3e2d548d-fce5-4374-862c-920686e8f26a --tree --recursive
|— Products
|  |— SAS Environment Manager
|     |— Dashboard Items
|        |— Application Activity (report)
|        |— CAS Activity (report)
|        |— Disk Space (report)
|        |— Infrastructure Data Server Tables (report)
|        |— Message Queue Activity (report)
|        |— System Activity (report)
|        |— User Activity (report)
```

Figure 7: Listing Folder Members and Hierarchy

SHOWING RULES

The authorization plug-in enables you to interact with the SAS Viya authorization rules. A common problem is trying to figure out why someone doesn't have permission to access a particular asset. With the authorization plug-in, that investigation might look like Figure 8.

```
$ ./sas-admin folders list | grep DemoFolder
e66<snip>0cf7 DemoFolder folder

$ ./sas-admin folders list-members --id e66<snip>0cf7 --details
Id Name Type Uri
f59<snip>4192 FancyReport child /reports/reports/a22<snip>f8c2

$ ./sas-admin authorization explain --target-uri /folders/folders/e66<snip>0cf7
Resource: /folders/folders/e66cc029-fcb2-4e74-b4d5-748337bc0cf7
Principal Read Update Delete Secure Add Remove Create
authenticatedUsers prohibit prohibit prohibit prohibit prohibit prohibit prohibit
SASAdministrators (group) grant grant grant grant grant grant grant grant
sasboot (user) grant grant grant grant grant grant grant prohibit
ApplicationAdmins (group) grant prohibit prohibit prohibit prohibit prohibit prohibit prohibit

$ ./sas-admin authorization explain --target-uri /reports/reports/a22<snip>f8c2
Resource: /reports/reports/a2241fff-b708-4d5e-b3f4-582b77fef8c2
Principal Read Update Delete Secure Add Remove Create
authenticatedUsers prohibit prohibit prohibit prohibit prohibit prohibit prohibit
SASAdministrators (group) grant grant grant grant grant grant grant grant
User1 (user) grant grant grant grant grant grant grant prohibit
```

Figure 8: Investigating permissions

The explain subcommand will show all the rules that apply to a specified resource. In this case, any authenticated user is denied access to FancyReport, while User1 has rights to interact with it. SASAdministrators group always has full rights on any resource.

CHANGING THE CONFIGURATION

One of the most powerful plug-ins is configuration. With this plug-in, you can interact directly with the environment configuration as you would in SAS Environment Manager. The advantage of using the CLI is the ability to create easily reproducible tasks. A perfect example is changing the logging levels. Consider a situation where you've contacted SAS Technical Support with a problem. They've asked you to increase the logging levels for a service. You can perform this action in SAS Environment Manager, but it takes many clicks to get the settings correct. With the configuration plug-in, SAS Technical Support can send you a file that contains the exact settings that they want you to apply. Figure 10 illustrates the command to apply the change

```
{
  "name": "loggingChange",
  "items": [
    {
      "metadata": {
        "mediaType": "application/vnd.sas.configuration.config.logging.level+json;version=1",
        "services": ["authorization"]
      },
      "name": "com.sas.authorization",
      "level": "DEBUG"
    }
  ]
}
```

Figure 9: logging.json

```
$ ./sas-admin configuration configurations create --file logging.json
"POST" "/configuration/configurations" complete from logging.json".
```

Figure 10: Applying configuration change

The beauty of this particular change is that it does not require a restart. It usually takes less than a minute for an instance to react to a logging change. After you have the information you need from using DEBUG-level logging, you can easily change back to the default logging settings by simply deleting the configuration that you added. Figure 11 illustrates two commands - a query of the service configuration followed by a delete of the configuration that you added.

```

$ ./sas-admin configuration configurations list --service authorization
Id      DefinitionName      Name      Services      IsDefault
d2b<snip>d579      jvm      com.sas.authorization      authorization      true
412<snip>93fb      logging.level      com.sas.authorization      authorization      false
2f7<snip>9c46      sas.abacus      GLOBAL      true
019<snip>4c27      sas.authorization      authorization      true

$ ./sas-admin configuration configurations delete --id 4127c641-5af3-4b87-92a8-b210440993fb
The deletion of configuration "4127c641-5af3-4b87-92a8-b210440993fb" is complete.

```

Figure 11: Querying and Changing the Configuration File

Some common changes do require a restart. Figure 12 illustrates a configuration file that requires a restart. The file is applied like the configuration in Figure 10. In this case, you must restart the authorization service before the memory change is picked up. Memory configurations such as this one are not normally deleted. To change it again, you use the same process to update the configuration with a different value.

```

{
  "name": "java heap",
  "items": [
    {
      "metadata": {
        "mediaType": "application/vnd.sas.configuration.config.jvm+json;version=1",
        "services": ["authorization"]
      },
      "java_option_xmx": "-Xmx512M"
    }
  ]
}

```

Figure 12: Changing the Memory Configuration

CREATING A CAS LIBRARY DEFINITION

One of the required tasks for any system is to have a set of CAS libraries defined for access to data. Figure 13 shows a very basic way to accomplish this using the cas plug-in.

```
$ sas-admin cas caslibs create dnfs --caslib HPS --server cas-shared-default --path
/CASData/sashdat/nfs/
The requested caslib "HPS" has been added successfully.
```

Figure 13: Creating a CAS Library Definition

SCRIPTING AROUND CLI COMMANDS

Another purpose of the sas-admin command is to enable customers to write scripts that can be used to automate repetitive tasks. Figure 14 is an example of a set of tasks that could be used to automate the first steps performed on a new installation. This command uses the server installation and does not require any human interaction.

```
/opt/sas/viya/home/bin/sas-admin profile set-endpoint http://mymachine
/opt/sas/viya/home/bin/sas-admin profile set-output text
/opt/sas/viya/home/bin/sas-admin auth login -u sasboot -p pw1
/opt/sas/viya/home/bin/sas-admin configuration configurations \
create --file ldapSettings.json
/opt/sas/viya/home/bin/sas-admin folders create \
--name "Customer Reports" --description "Root folder for customer reports"
```

Figure 14: Automating Installation Setup Tasks

This example contains commands that use global options to specify the profile information, which keeps you from having to set up a profile manually. The commands also provide authentication so that there are valid tokens. The command then updates the LDAP settings and creates a new root folder.

CONCLUSION

The SAS CLI sas-admin command is a very powerful tool to add to your administrative toolbox. You can automate common tasks, promote content, and figure out authorization problems quickly, without taking your fingers off the keyboard to find that pesky mouse.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Danny Hamrick
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Danny.Hamrick@sas.com
<http://www.sas.com>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.