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 a response back from the system. You can use a CLI to interact directly with SAS Viya programmatically without a GUI.
## Inventory

The following administrative CLIs are available in SAS Viya:

<table>
<thead>
<tr>
<th>Name</th>
<th>Scope and Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>Hosts other CLIs that run as plug-ins to this one. The top-level administrative command-line interface that is used to initialize, authenticate, and execute other plug-ins.</td>
</tr>
<tr>
<td>audit</td>
<td>Gets SAS audit information. See &quot;CLI Examples: Audit&quot;.</td>
</tr>
<tr>
<td>backup</td>
<td>Manages backups. See &quot;CLI Examples: Backup&quot; on page 19.</td>
</tr>
<tr>
<td>restore</td>
<td>Manages restore operations. See &quot;CLI Examples: Restore&quot;.</td>
</tr>
<tr>
<td></td>
<td>For information about cas environment variables, see CAS Administration: Details on page 22.</td>
</tr>
<tr>
<td>configuration</td>
<td>Manages the operations of the configuration service. See &quot;CLI Examples: Configuration&quot; on page 23.</td>
</tr>
<tr>
<td>compute</td>
<td>Manages the operations of the compute service. See &quot;CLI Examples: Compute&quot; on page 23.</td>
</tr>
<tr>
<td>fonts</td>
<td>Manages fonts that are provided by SAS as well as custom fonts that are registered in SAS Visual Analytics 8.2. See &quot;CLI Examples: Fonts&quot; on page 25.</td>
</tr>
<tr>
<td>devices</td>
<td>Manages mobile device blacklist and whitelist actions and information. See &quot;CLI Examples: Device Management&quot;.</td>
</tr>
<tr>
<td>identities</td>
<td>Gets identity information, and manages custom groups. See &quot;CLI Examples: Identities&quot; on page 27.</td>
</tr>
<tr>
<td>licenses</td>
<td>Manages SAS product license status and information. See &quot;CLI Examples: Licensing&quot;.</td>
</tr>
</tbody>
</table>
## Key Points

Here are key points for using the CLIs:

- To prepare to use the admin CLI plug-ins, see “Command-Line Interface: Preliminary Instructions” on page 5.

- Commands and subcommands are case-sensitive.

- You must precede the options of the commands with `- -`.  
  
  **Note:** Some command options support a shortcut notation in which a single hyphen precedes the option. For example, you can use `--help` or `-help` for the help global option.

- Use the Help from within each CLI for information about the available commands, subcommands, and options. For the admin CLI plug-ins, see “Command-Line Interface: Syntax” on page 7.

**Note:** This document reflects sas-admin CLI functionality in the initial release of SAS Viya 3.3. 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 5.

- 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 include line breaks within commands for presentation purposes only. 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 logged in using a named profile, you must do one of the following to use that profile:

  - set the `SAS_CLI_PROFILE` environment variable to the name of the profile. This will remain in effect until you log off from the environment.
For example, suppose that you have set the `SAS_CLI_PROFILE` environment variable to `Target1` as follows: `export SAS_CLI_PROFILE=Target1`

Then, you can log on to the environment with the "Target1" profile with this command: `sas-admin auth login`

- include the `profile` global option on each CLI command as follows: `sas-admin --profile profile-name CLI-name CLI-commands`

Then, you can log on to the environment with the "Target1" profile with this command: `sas-admin --profile Target1 auth login`

---

**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), then you must set the `SSL_CERT_FILE` environment variable to the path location of the trustedcerts.pem file.

If the `SSL_CERT_FILE` environment variable is not already set, complete these steps:

1. In a command window on the SAS Viya machine, navigate to the following directory: `/opt/sas/viya/home/bin`.
2. Set the environment variable as follows: `export SSL_CERT_FILE=/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.pem`.

### Create at Least One Profile

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

1. In a command window on the SAS Viya machine, navigate to the following directory: `/opt/sas/viya/home/bin`.
2. At the command prompt, enter a command to initialize a new profile. Here are examples:

   - To create a default (unnamed) 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 Profile” on page 10 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 10.

3. Respond to the subsequent prompts as follows:

   **Service Endpoint**
   Specify the URL for the SAS Viya environment. Use the following format: `communications-protocol://web-server-host-name:web-server-port`
   For example: `https://host.example.com:443`
Output type Specify your preferred format for CLI output (text, json, or fulljson).

Note: For more information about the output types, see “Output Type” on page 10.

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

1 In a command window on the SAS Viya machine, navigate to the following directory: /opt/sas/viya/home/bin.

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

To use your default (unnamed) 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
```

3 At the subsequent prompts, enter your user ID and password.

By default, your authentication remains active for 12 hours. You can use the auth logout command to sign out.

Note: When you run the auth login 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 10.

Note: If you logged in using a named profile, you must do one of the following to use that profile:

- set the SAS_CLI_PROFILE environment variable to the name of the profile. This will remain in affect until you log off from the environment.

  For example, suppose that you have set the SAS_CLI_PROFILE environment variable to Target1 as follows: export SAS_CLI_PROFILE=Target1

  Then, you can log on to the environment with the “Target1” profile with this command: sas-admin auth login

- include the profile global option on each CLI command as follows: sas-admin --profile profile-name CLI-name CLI-commands

  Then, you can log on to the environment with the “Target1” profile with this command: sas-admin --profile Target1 auth login

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 interface-name [global options] command [command options] [subcommand] [subcommand options] [arguments]
```

- `sas-admin` specifies the sas-admin CLI.
- `interface name` specifies the CLI plug-in.
- `[global options]` specifies options that are applicable to all CLIs.
- `command` specifies a command that is specific for the CLI that you are using.
- `[command options]` specifies options for the CLI-specific command that you are using.
- `[subcommand]` specifies a subcommand for the CLI 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:** Change the output type that is used with the audit CLI to JSON.

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

**Example:** Show more detailed information for the mobile device CLI blacklist list command.

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

For information about the CLIs that are available, see “Inventory” on page 3.

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 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:
   1.0.9

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
   identities
   job
   licenses
   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, $LANG]
   --output                     Specifies output format - text, json, fulljson. [$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) 2016-2017 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 1.0.9
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 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 admin CLI.

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

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

**NAME:**

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

**USAGE:**

```
sas-admin devices blacklist [arguments...]
```

**COMMANDS:**

- `add` Adds a device to the blacklist.
- `delete` Deletes a device from the blacklist.
- `list` Lists the devices in the blacklist.

Example: List the options of the `list` subcommand of the `blacklist` command that is a part of the devices plug-in to the 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.
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: 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, or fulljson).

- **credentials.json**
  Will contain the authentication tokens for your session that are created after you issue the `auth login` global command.

Default Profile and Named Profile
You can create a default (unnamed) 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 (unnamed) 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 also use a named profile to eliminate the requirement 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 profile name and eliminate the need to create a new profile with these settings.

Note: If you logged in using a named profile, you must do one of the following to use that profile:

- set the `SAS_CLI_PROFILE` environment variable to the name of the profile. This will remain in effect until you log off from the environment.
For example, suppose that you have set the `SAS_CLI_PROFILE` environment variable to `Target1` as follows:

```
export SAS_CLI_PROFILE=Target1
```

Then, you can log on to the environment with the "Target1" profile with this command:

```
sas-admin auth login
```

- include the `profile` global option on each CLI command as follows: `sas-admin --profile profile-name CLI-name CLI-commands`

Then, you can log on to the environment with the "Target1" profile with this command:

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

### Examples

Here are typical examples of creating default (unnamed) and named profiles:

**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 **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.

```json
{
    "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.

```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 5 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 token expires after 12 hours, so you might need to re-execute the command at a later time to reconnect to the environment.

To log on, issue the auth login command. For syntax information, see “Structure” on page 7. Here are typical examples:

- Use this command to log on with a default (unnamed) 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 7. Here are typical examples:

- Use this command to log off from the SAS Viya environment that is specified in your default (unnamed) 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 “Target1” environment as follows: sas-admin auth logout.

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 5.

**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 7.

CLI Examples: Audit

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 5.

**Examples**

**Example:** List the records of audit entries for reports.

```
sas-admin audit list --application reports
```
Example: List the records of audit entries of type security and sort by user.

```
sas-admin audit list --sort-by user --type security
```

Example: List the records of audit entries of type security and state of success, and then write the results as CSV to an output file:

```
sas-admin audit list --state success --type security --csv /tmp/outputfile.text
```

See Also

- “Command-Line Interface: Overview” on page 2
- SAS Administration: Auditing

---

**Command-Line Examples: CAS Authorization**

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 5.

---

**Getting Access Information**

Example: List tableA's direct access controls.

```
sas-admin cas tables list-controls --server serverA --caslib caslibA --table tableA
```

Example: List tableA's direct access controls and inherited settings.

```
sas-admin cas tables list-controls --server serverA --caslib caslibA --table tableA --list-type all
```

Example: Show effective (net) access to tableA for userA.

```
sas-admin cas tables list-controls --server serverA --caslib caslibA --table tableA
--control-type effective --user userA
```

Example: Show effective access to tableA for groupA and groupB.

```
sas-admin cas tables list-controls --server serverA --caslib caslibA --table tableA
--control-type effective --group 'groupA|groupB'
```

Example: Show the source of userA's access to tableA.

```
sas-admin cas tables list-controls --server serverA --caslib caslibA --table tableA
--control-type origin --user userA
```

---

**Managing Access Controls**

Example: Remove all direct access controls from tableA.

```
sas-admin cas tables clear-controls --server serverA --caslib caslibA --table tableA
```

Example: Set a simple row-level access control on the CARS table so that members of groupA can see only those rows where the value in the Make column is Ford.

```
sas-admin cas tables add-control --server serverA --caslib caslibA --table CARS --group groupA
--grant Select --where "make='Ford'"
```

Example: Set an identity-based, row-level access control on the Salary table. The reason is so that each authenticated user can see only those rows where the value in the User column is his or her own user ID.

```
sas-admin cas tables add-control --server serverA --caslib caslibA --table salary --group "User::"--grant Select --where "User='SUB::SAS.Userid'"
```
Example: Enable guests to read data in the Public caslib.

```bash
1 sas-admin cas caslibs add-control --server serverA --caslib Public --guest --grant ReadInfo --superuser
2 sas-admin cas caslibs add-control --server serverA --caslib Public --guest --grant Select --superuser
```

1 This example is applicable to only a deployment where guest access is enabled.

In the standard configuration, because only a privileged user can modify access to the Public caslib, the superuser option is specified here. Only a member of the Superuser role for the specified CAS server can obtain elevated privileges by specifying the superuser option.

2 The grants in this example support reading of data, but do not support just-in-time loading of data. Instead of granting LimitedPromote to guest, consider using a different technique for ensuring that data is loaded.

Because this example does not create and reuse a dedicated superuser session, you must specify the superuser option in each command where elevated privileges are needed.

Example: Enable groupA to read data (and perform just-in-time data loading) in a new caslib.

```bash
1 sas-admin cas caslibs add-control --server serverA --caslib caslibA --group groupA --grant ReadInfo
2 sas-admin cas caslibs add-control --server serverA --caslib caslibA --group groupA --grant Select
3 sas-admin cas caslibs add-control --server serverA --caslib caslibA --group groupA --grant LimitedPromote
```

Example: Make the same changes as in the preceding example, but use an access control transaction so that you can review your changes before you commit them to the server.

```bash
1 sas-admin cas sessions create --name mysess --server serverA --superuser
2 sas-admin cas transactions checkout --session-id XYZ --server serverA --caslib caslibA
3 sas-admin cas caslibs add-control --session-id XYZ --server serverA --caslib caslibA --group groupA --grant ReadInfo
4 sas-admin cas caslibs add-control --session-id XYZ --server serverA --caslib caslibA --group groupA --grant Select
5 sas-admin cas caslibs add-control --session-id XYZ --server serverA --caslib caslibA --group groupA --grant LimitedPromote
6 sas-admin cas transactions list-controls --session-id XYZ --server serverA --caslib caslibA
7 sas-admin cas transactions commit --session-id XYZ --server serverA
8 sas-admin cas sessions delete --session-id XYZ --server serverA
```

1 Start a session. If you are a member of the Superuser role for the associated CAS server, give the session Superuser status.

2 Check out the caslib into the session that you just started. Use the session ID that is returned from the preceding command. The session-id value XYZ is used here for simplicity. Checking out an object automatically starts a transaction.

3 Grant access within the transaction.

4 Grant access within the transaction.

5 Grant access within the transaction.

6 Review the results. Because you supply the session ID, the output reflects the uncommitted changes in your session.

7 After you review the output from the list-controls command, commit your changes.
If you are finished, it is a good practice to delete your session.

**Example:** Replace any direct access controls on tableA with access controls from an external JSON file. In this example, the replacement access controls are derived from tableB and are then applied to tableA.

```
sas-admin cas tables list-controls --server serverA --caslib caslibA --table tableB > ac.json
sas-admin cas tables replace-controls --server serverA --caslib caslibA --table tableA
    --source-file ac.json
sas-admin cas tables list-controls --server serverA --caslib caslibA --table tableA
```

This example writes the direct access controls for tableB to the file `ac.json` in the directory from which you are running your CLI.

**Note:** This example assumes that the profile that you are using specifies `json` as your default output type. Otherwise, you must use the global option `--output` to specify `json` as the output type for this command. That option must immediately follow `sas-admin`.

**Note:** You can reference an absolute path or a relative path.

1. Delete any direct access controls on tableA, and replace them with the access controls that you wrote to the ac.json file.
2. Review the new set of direct access controls on tableA.

**Note:** You can modify the output from tableB before you use it to replace direct access controls on tableA.

### Details and Tips

#### Basics

- The term *access control* refers to an access control in the CAS authorization system. To manage access to content objects and functionality, see “Command-Line Examples: General Authorization” on page 16.
- You can add, delete, and replace only direct access controls.
- A request to delete a direct access control that does not exist does not generate an error.
- To modify access that a table inherits, set direct access controls on the parent caslib.
- You cannot modify access that a caslib inherits.
- Use of *access control transactions* is optional. You do not have to check out an object in order to modify its access controls.
- In the list-controls command, use the control-type and list-type options as follows:
  - Use the control-type option only if you want to obtain net access information (*effective*) or source information (*origins*).
  - The list-type options are not relevant if the control-type is *effective* or *origin*.
  - Use the list-type option only if you want to obtain inherited settings, in addition to direct access controls (*all*) or instead of direct access controls (*inherited*).
- You can obtain *origins* information for only one identity at a time.
- The value `serverA` is used in the examples for simplicity. A more typical server name is `cas-shared-default`.
- See “Details” on page 22 for information about using environment variables with the CAS commands.
Principals

- To specify a particular identity (where supported), you must provide a user ID or a group ID rather than a name.
  
  **CAUTION! The user ID and the group ID that you provide are not validated.** Make sure the IDs that you provide are accurate.

- To specify multiple users or multiple groups (where supported), use the pipe character (|) as a delimiter and enclose the string in single quotation marks (for example: `--user 'userA|userB'`). You cannot specify both users and groups in a single request.

- The group * corresponds to Authenticated Users. To specify that principal, enter `--group '*'`.

- The Guest principal represents all users who connect as guests. To specify that principal, enter the option `--guest` and do not specify a value.

Permissions

- To specify a particular permission (where supported), use one of the following case-insensitive values: ReadInfo, Select, LimitedPromote, Promote, CreateTable, DropTable, DeleteSource, Insert, Update, Delete, AlterTable, AlterCaslib, or ManageAccess. You cannot specify multiple permissions in a single request.

- For information about the scope and purpose of each permission, see “CAS Authorization: Concepts” in SAS Viya Administration: Cloud Analytic Services Authorization.

Fine-Grained Controls

- Row-level grants are always for the Select permission on a table. The syntax for row-level permission filters is the same as in other CAS authorization interfaces.

- You cannot set column-level permissions using this interface.

See Also

- “Command-Line Interface: Overview” on page 2

Command-Line Examples: General Authorization

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 5.

Getting Access Information

**Example:** Show detailed properties of a specified rule.

```
sas-admin authorization show-rule --id d85144aa-79dc-4852-b949-645cc5ff8ffc --details
```

**Example:** Show effective access for a specified object URI. To return information about contributing rules, you must specify fulljson output in your profile. Specifying the output global option in the CLI command inline is insufficient.

```
sas-admin authorization explain --target-uri /SASHome/**
```

**Example:** List all the rules in the deployment.
Managing Rules

Example: Give groupA Read access to reportA.

```bash
sas-admin authorization authorize --permissions Read --group groupA
--object-uri /reports/reports/33db163a-716e-4980-a5bc-6c42a0278c40
```

Example: Provide guest access to reportA.

```bash
sas-admin authorization authorize --permissions Read --guest
--object-uri /reports/reports/33db163a-716e-4980-a5bc-6c42a0278c40
```

Example: Grant Authenticated Users Read access to folderA and its child members.

```bash
sas-admin authorization authorize --permissions Read --authenticated-users
--object-uri /folders/folders/2414f911-d276-4357-8550-fcf03753c9e7/**
--container-uri /folders/folders/2414f911-d276-4357-8550-fcf03753c9e7
```

Example: Delete a rule.

1. ```bash
sas-admin authorization show-rule --id d85144aa-79dc-4852-b949-645cc5ff8fffc --details
```
2. ```bash
sas-admin authorization remove-rule --id d85144aa-79dc-4852-b949-645cc5ff8fffc
```

1. Review the rule’s properties so that you are certain you are deleting the correct rule.
2. Delete the rule.

Example: Change the principal in an existing rule so that the rule is assigned to Group B, which has groupB as its ID.

1. ```bash
sas-admin authorization show-rule --id cd75a376-c5d4-4951-9e57-cf441610628c --details
```
2. ```bash
sas-admin authorization update-rule --id cd75a376-c5d4-4951-9e57-cf441610628c
--grant --permissions Read,Update,Delete
```

1. Review the rule’s properties so that you are certain you are modifying the correct rule.
2. Modify the rule.

Example: Include the Update and Delete permissions in an existing rule that already grants the Read permission.

1. ```bash
sas-admin authorization show-rule --id cd75a376-c5d4-4951-9e57-cf441610628c --details
```
2. ```bash
sas-admin authorization update-rule --id cd75a376-c5d4-4951-9e57-cf441610628c
--grant --permissions Read,Update,Delete
```

1. Review the rule’s properties so that you are certain you are modifying the correct rule.
2. Modify the rule.

Example: Edit the description in an existing rule.

1. ```bash
sas-admin authorization show-rule --id 0e8a6ce7-e51a-40cc-aeda-e2a5efb53ca --details
```
2. ```bash
sas-admin authorization update-rule --id 0e8a6ce7-e51a-40cc-aeda-e2a5efb53ca
--description 'This is a revised description.'
```

1. Review the rule’s properties so that you are certain you are modifying the correct rule.
2. Modify the rule.
Details and Tips

Throughout this topic, the term rule refers to an authorization rule in the general authorization system. To manage access to CAS objects (such as caslibs and tables), see “Command-Line Examples: CAS Authorization” on page 13.

To assign a rule to a principal type, use one of the following options:

<table>
<thead>
<tr>
<th>Principal Type</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guest</td>
<td>--guest</td>
</tr>
<tr>
<td>Authenticated Users</td>
<td>--authenticated-users</td>
</tr>
<tr>
<td>Everyone</td>
<td>--everyone</td>
</tr>
</tbody>
</table>

To assign a rule to a particular identity, you must provide a user ID or a group ID, not a name. For example, to assign a rule to the SAS Administrators custom group, specify: --group SASAdministrators

**CAUTION!** The user ID and the group ID that you provide are not validated. Make sure the IDs that you provide are accurate.

You can obtain the ID for a user or group from the Users page in SAS Environment Manager.

You can obtain the objectURI for a content object (such as a report) from the Content page in SAS Environment Manager. Select the object in the navigation pane. On the right, the URI field in the Basic Properties section contains the object URI. To target the object URI for a content object (such as a report) or a container (such as a folder), append a suffix. See “Rule Targets” in SAS Viya Administration: General Authorization.

You can obtain the ID for a rule from the Rules page in SAS Environment Manager. Right-click a rule and select Properties. The last field in the Properties window contains the rule’s ID.

When you use the show-rule command, always specify that you want details to be returned. Some of the fields that can be essential to interpreting a rule are excluded from the default response. For example, a condition is not included in the default response.

When you use the update-rule command, specify only the options for the rule properties that you want to modify. For any option that you specify in the update-rule command, provide the complete replacement value or values.

When you use the explain command, the returned information indicates the effective (net) access of each relevant principal for all permissions.

**Note:** In the output from the explain command, the grant and prohibit values indicate effective (net) access, not direct settings. For example, a prohibit value in the output from the explain command is usually caused by the lack of any relevant grant, rather than by the existence of a relevant Prohibit rule. See “Authorization Decisions” in SAS Viya Administration: Cloud Analytic Services Authorization.

Enabling or disabling guest access involves more than running the enable-guest-access or disable-guest-access command. See the guest access documentation.

See Also

- “Command-Line Interface: Overview” on page 2
- SAS Viya Administration: General Authorization
CLI Examples: Backup

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 5.

Examples

Example: List the first 50 backup jobs.

sas-admin backup list --limit 50

Example: Start a binary backup named backupA.

sas-admin backup start --slug backupA

See Also
- “Backup and Restore: Overview” in SAS Viya Administration: Backup and Restore
- “Command-Line Interface: Overview” on page 2

CLI Examples: Restore

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 5.

Examples

Example: Show the history of restore operations.

sas-admin restore list

Example: Start a restore operation of a specified backup, and specify that the name of the restore operation is restoreA.

sas-admin restore start --backup-name backup-ID --slug restoreA

See Also
- “Backup and Restore: Overview” in SAS Viya Administration: Backup and Restore
- “Command-Line Interface: Overview” on page 2

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 5.
Facilitate Guest Access

Example: To facilitate guest access on the specified server, modify the direct access controls for the predefined caslibs on the server. To do this, use the controls that are defined in the specified source file. Perform this action with elevated privileges if the user is a member of the Superuser role. This is one of several required steps to enable guest access. For a complete set of instructions on how to enable guest access, see “Authentication: Guest Access” in SAS Viya Administration: Authentication.

sas-admin cas facilitate-guest --source-file path-to-controls-file --server serverA --superuser

The default access control file is located on the SAS Viya machine at this location: /opt/sas/viya/home/share/guest/facilitate-guest-controls.txt.

Manage CAS Role Memberships

Example: List the administrative users on the specified CAS server.

sas-admin cas admin-users list --server serverA

Example: Add the user user1 to the Superuser role on the specified CAS server.

sas-admin cas admin-users add --user user1 --server serverA

Example: Delete the group with the ID group1 and name group1_name from the Superuser role on the specified CAS server. The action is performed without prompting the user for confirmation since the force option is used.

sas-admin cas admin-users delete --group group1 --server serverA --name group1_name --force

TIP To delete a user or a group from the administrative users, you must specify the name of the user or the group as well as the identity of the user or the group. Use the name option to specify the name. Use the user option to specify the ID for a user. Use the group option to specify the ID for a group. In order to obtain the ID, use the admin-users list command.

Manage SAS Sessions.

Example: List the sessions of which you are the owner on the specified CAS server.

sas-admin cas sessions list --server serverA

Example: Using elevated privileges, list 50 sessions for which the owner is user1 on the specified CAS server, and sort by state. You must be a member of the Superuser role to run the command with elevated privileges.

sas-admin cas sessions list --server serverA --superuser --limit 50 --owner user1 --sort-by state

Example: Using elevated privileges, list all sessions for which the name contains the string dataExplorer on the specified CAS server. You must be a member of the Superuser role to run the command with elevated privileges.

sas-admin cas sessions list --server serverA --superuser --all --name-contains dataExplorer

Example: Using elevated privileges, show additional information about the session with ID 12345 on the specified CAS server. You must be a member of the Superuser role to run the command with elevated privileges.

sas-admin cas sessions show-info --superuser --session-id 12345 --server serverA

Manage SAS Formats

Example: Display the values for the specified SAS format (in this example, $fruit).
Example: Display all SAS format libraries for serverA.

```
sas-admin cas formats list --format-library=* --server serverA
```

Example: Display the search order for SAS format libraries for serverA.

```
sas-admin cas formats search-order --server serverA
```

Note: Format names that are used in the commands must be enclosed in single quotation marks.

If you run the show-info command for a format, the results are displayed according to these rules:

- If you request information about a non-locale format (such as `$charfmt`) and you do not specify the `ignore-locale` option, the default locale format will be returned. If the default system locale is en_US and the `ignore-locale` option was not specified, then the returned format for `$charfmt` is en_US-$charfmt.

- If you request information about a non-locale format (such as `$charfmt`), you must specify the `ignore-locale` option in order to return the actual format with no locale. If the `ignore-locale` option was specified, the returned format for `$charfmt` is $charfmt.

- If you request information about a format that has a non-existent locale, the format for the default system locale is returned. Suppose that you request information about en_bogus-$charfmt, which is a format that does not exist. If the default system locale is en_US, the returned format is en_US-$charfmt.

- If you request information about an invalid format, an error is returned. Suppose that you request information about en-US-$charfmt. The hyphen in the locale name (en-US) should have been an underscore (en_US). No format is returned, and an error is displayed.

See Also
See “Data Administration: Reference” in SAS Viya Administration: Data for more information about user-defined formats.

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 22.

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.
Manage Caslibs

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 22.

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, list the first 20 global caslibs. You must be a member of 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 a member of 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
```

Details

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: SAS Cloud Analytic Services*
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 5.

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: Compute

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 5.

Examples

Example: List the compute contexts.

sas-admin compute contexts list

Example: Validate the compute context session with the specified ID.

sas-admin compute contexts validate --id 389fee7a-e164-4e45-b836-a301638e9945

Example: Delete the compute context session with the specified name.
sas-admin compute contexts delete --name "SAS Job Execution compute context"

**Example:** List the launcher contexts.

sas-admin compute launchers list

**Example:** Delete the launcher context with the specified ID.

sas-admin compute launchers delete --id 8fbdd5f8-a2ee-42a5-a228-8737a0cf778f

**Example:** List the compute sessions.

sas-admin compute sessions list

**See Also**

“Command-Line Interface: Overview” on page 2

---

**CLI Examples: Folders**

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 5.

**Examples**

**Example:** Create folderA. Add subfolderA as a child folder to folderA.

1. sas-admin folders create --name folderA
2. sas-admin folders create --name subfolderA --parent-id parent-folder-ID

1. Create folderA and note the ID.
2. Issue the command to create subfolderA and specify the ID of folderA as the parent folder.

**Example:** List the members of folderA.

sas-admin folders list-members --id folderA-ID

**Example:** Update the name of folderA to departmentA.

sas-admin folders update --id folderA-ID --name departmentA

**Example:** Delete departmentA.

sas-admin folders delete --id departmentA-ID

**Details**

Many of the folders commands require the ID of a folder as an argument. The ID of a folder is displayed when you create the folder. The ID of folders is also displayed when you list folders.

**See Also**

- “Command-Line Interface: Overview” on page 2
- “Content Management: Overview” in SAS Viya Administration: Content Management
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 5.

Examples

Example: Add and register the open-source OpenSans-Bold.ttf Google font.

```
sas-admin fonts add --uri https://server:port/fonts/OpenSans-Bold.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: Device Management

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 5.

Examples

Example: Determine whether the device with ID device1 is authorized for use in the environment.

sas-admin devices authorized-devices validate --device-id device1

Example: Show device enforcement status.

sas-admin devices enforcement status

Example: Add the device with ID device1 to the whitelist.

sas-admin devices whitelist add --device-id device1

Example: List the devices that are enabled in the whitelist.

sas-admin devices whitelist list

Example: Add the device with ID device1 to the blacklist.

sas-admin devices blacklist add --device-id device1

Example: Remove the device with ID device1 from the blacklist.

sas-admin devices blacklist delete --device-id device1

Example: From a list of the devices of type iPhone that have connected or attempted to connect to the server, add a specific iPhone to the blacklist.

1. sas-admin devices last-access list --device-type iPhone
2. sas-admin devices blacklist add --device-id device-id

1 List the last-access attempts of all devices of type iPhone.
2 Add a device to the blacklist using the device ID that was identified in the previous step.

Example: List in fulljson output the last access attempts to the server for all devices.

sas-admin --output fulljson devices last-access list
CLI Examples: Identities

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 5.

Examples

Example: Add user1 to the group that has the ID 4444.

```bash
sas-admin identities add-member --user-member-id user1 --group-id 4444
```

Example: Create a group with the name Salesgroup, the group ID 8888, and the description of “Custom sales group”.

```bash
sas-admin identities create-group --id 8888 --name Salesgroup --description "Custom sales group"
```

Example: Remove user1 from the sales and marketing groups.

```
1  sas-admin identities list-memberships --user-id user1
2  sas-admin identities remove-member --group salesgroup --user-member-id user1
3  sas-admin identities remove-member --group marketinggroup --user-member-id user1
```

1 Verify the groups that user1 belongs to.
2 Remove user1 from the group that has the group ID sales-group.
3 Remove user1 from the group that has the group ID marketing-group.

Example: Show details about the group that has the group ID ABC.

```bash
sas-admin identities show-group --id ABC
```

Details

- If a group is created with no name, the specified ID will be used for the name.
- The following identities commands list only 50 items at a time by default:
  - list-groups
  - list-members
  - list-memberships
- To list more than 50 items, you can use the --limit option.

Example:

```bash
sas-admin identities list-members --group-id ABCD --limit 100
```

See Also

- “Command-Line Interface: Overview” on page 2
- “Identity Management Overview” in SAS Viya Administration: Identity Management
CLI Examples: Licensing

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 5.

Examples

Example: List site information, which is the site name, the site number, the operating system name, the release number, the server date, the grace period, and the warning period.

```
sas-admin licenses site-info list
```

Example: List all products in the system whose name contains “Visual Analytics”.

```
sas-admin licenses products list --name-contains "Visual Analytics"
```

Example: List all products in the system that are expired.

```
sas-admin licenses products list --expired
```

Example: List the products in the system that have these identifiers: 827, 921, and 985.

```
sas-admin licenses products list --product-ids 827,921,985
```

Example: List the number of products with a current license that are deployed.

```
sas-admin licenses count --current
```

Example: List the Data-Connector products whose licenses are covered within the grace period.

```
sas-admin licenses data-connectors list --grace
```

See Also

- “Command-Line Interface: Overview” on page 2
- SAS Viya Administration: Licensing

CLI Examples: Job

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 5.

Examples

Example: List the job flows.

```
sas-admin job flows list
```

Example: Generate a template for a flow and write the output to a file.

```
sas-admin job flows generate-template --file-out /tmp/output.txt
```

Example: List job flow scheduling service objects.

```
sas-admin job schedulers list
```
Details

Here is an example of a template file that is generated from a flow that you created:

```
{
  "name": "Replace with name of the flow",
  "description": "(Optional) Replace with description of the flow",
  "triggerType": "Replace with trigger type, select one of: runnow, manual, event",
  "triggerCondition": "Replace with either any or all",
  "flowProperties": {},
  "defaultJobProperties": {}
}
```

See Also

“Command-Line Interface: Overview” on page 2

---

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 5.

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 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
```

Details

- 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 21`.

See Also

“Command-Line Interface: Overview” on page 2
CLI Examples: Tenant 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 5.

Examples

Example: Create the tenant with the ID companya and assign a description.

```
sas-admin tenant create --id companya --description "description of companya"
```

Example: Delete the tenant that has the ID companya.

```
sas-admin tenant delete --id companya
```

Example: Offboard the tenant that has the ID companya.

```
sas-admin tenant offboard --id tenanta
```

Example: Enable the tenant that has the ID companya.

```
sas-admin tenant enable --id companya
```

Example: Disable the tenant that has the ID companya.

```
sas-admin tenant disable --id companya
```

Details

- The Help for the ID of a tenant states that the string must match this pattern: ^[a-z]+[a-z0-9]*
  
  This pattern means that the ID of a tenant must start with a lowercase letter, followed by any number of lowercase letters. Use of numbers is optional.

- When you enable a tenant so that users can sign in to it, the access policy of the tenant is changed from `providerTenantUsersOnly` to `allUsers`.

  When you disable a tenant so that users can no longer sign in to it, the access policy of the tenant is changed from `allUsers` to `providerTenantUsersOnly`.

  For more information about the access policy of a tenant, see “Edit the Properties of a Tenant” in SAS Viya Administration: Multi-tenancy.

See Also

- “Command-Line Interface: Overview” on page 2
- SAS Viya Administration: Multi-tenancy

CLI Examples: Transfer

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 5.
Examples

Example: See the commands and subcommands that are available for the transfer import command.

sas-admin transfer import --help

Example: Export a new transfer package from an object that has the resource URI "/reports/reports/\nfaa7f5f2-0822-4ca0-9f92-23bda3e02738", and name the package "Export Report".

sas-admin transfer export --name "Export Report"
--resource-uri "/reports/reports/faa7f5f2-0822-4ca0-9f92-23bda3e02738"

Example: Get the mapping information for the transfer package that is named "Export Report" from the previous example.

1 sas-admin transfer list --name "Export Report"
2 sas-admin transfer get-mapping --id transfer-package-ID

1 Locate the ID for the transfer package that you want to export.
2 Issue the command to retrieve the mapping information for the "Export Report" transfer package that has the
specified ID.

Example: Upload a package source environment to the target environment, and write the mapping file to the
specified location.

1 sas-admin --profile source transfer download --id transfer-package-ID --file /tmp/MyPackage.json
2 sas-admin --profile target transfer upload --file /tmp/MyPackage.json --mapping /tmp/map.txt

1 Download the package to your local machine and store it in a package file that is named MyPackage.json.
2 Upload the MyPackage.json file to the target environment, and write the mapping file to the file map.txt.

Details

You can specify information about the export or import operation that you want to perform using the Transfer
service REST API standards, as follows:

- export

  You can specify information about the export operation that you want to perform using the request option of
  the transfer export command. The option accepts JSON input of type ExportRequest. The content of the
  input can be contained in a quoted string or in a file. The filename must begin with the at sign (@). The
  filename can be specified in either of two forms:

  @filename.txt

  @/path/filename.txt

  The content of this option makes up the POST request through which the export package is sent.

  Table A.1 HTTP POST Export Request Members

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>integer</td>
<td>The schema version number of the JSON media type. This is version 1.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the export job that is used to export objects from a source system to a transfer package. This is also the name of the transfer package that is being created.</td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>A short description of the export job.</td>
</tr>
<tr>
<td>items</td>
<td>list</td>
<td>The list of URIs to include in the transfer package.</td>
</tr>
</tbody>
</table>

Here is a sample JSON file of type ExportRequest. The name of the file is export.json.

```json
{
    "version": 1,
    "name": "My reports",
    "description": "Export of all my reports",
    "items": [
        "/reports/reports/4d083692-3c9a-4f2c-945f-e96fad972036",
        "/folders/folders/d4f1533a-229d-4d45-a5c9-b8a21fbc1e39"
    ]
}
```

You can use either of the following syntax options of the command to export the information:

- `sas-admin transfer export --request @/path-to-file/export.json`

  Note: When you include the information in a file, the first character following the request option must be the at sign (@). Therefore, if a pathname is used, it must start with the at sign (@).

- `sas-admin transfer export --request '{ "version": 1, "name": "My reports", "description": "Export of all my reports", "items": [ 
    "/reports/reports/4d083692-3c9a-4f2c-945f-e96fad972036",
    "/folders/folders/d4f1533a-229d-4d45-a5c9-b8a21fbc1e39"
    ] }'`

## import

You can specify information about the import operation that you want to perform using the request option of the transfer import command. The option accepts JSON input of type ImportRequest. The content can be contained in a quoted string or in a file. The filename must begin with an at sign (@). The filename can be specified in either of two forms:

- `@filename.txt`
- `@/path/.filename.txt`

The content of this option makes up the POST request through which the import package is sent.

**Table A.2 HTTP POST Import Request Members**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>integer</td>
<td>The schema version number of the JSON media type. This is version 1.</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the import job that is used to import objects from a transfer package to a target system.</td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>A short description of the import job.</td>
</tr>
<tr>
<td>packageUri</td>
<td>string</td>
<td>The package to import.</td>
</tr>
</tbody>
</table>
Here is an example of a JSON file of type ImportRequest that is named import.json:

```json
{
    "version": 1,
    "name": "My reports",
    "description": "import all my reports",
    "packageUri": "/transfer/packages/a2ef940e-14ac-4960-9a9a-7689702b06f0"
}
```

You can use either of the following syntax options of the command to import the information:

- `sas-admin transfer import --request @/path-to-file/import.json`
  
  **Note:** When you include the information in a file, the first character following the request option must be the at sign (@). Therefore, if a pathname is used, it must start with the at sign (@).

- `sas-admin transfer import --request '{ "version": 1, "name": "My reports", "description": "import all my reports ", "packageUri": "/transfer/packages/a2ef940e-14ac-4960-9a9a-7689702b06f0" }'`

**See Also**

- "How To" in SAS Viya Administration: Promotion (Import and Export)
- “Command-Line Interface: Overview” on page 2