## Contents

*Chapter 1 • Command-Line Interfaces*  
Using SAS Decision Manager CLIs .................................................. 1  
Global Commands ................................................................. 1  
Global Options ................................................................. 2  
Dictionary ................................................................. 4
Chapter 1
Command-Line Interfaces

Using SAS Decision Manager CLIs

In SAS Viya, a command-line interface (CLI) is a user interface to the SAS Viya REST services. In this interface, 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. For information about the CLIs that are provided with SAS Viya, see “Command-Line Interface: Overview” in SAS Viya Administration: Using the Command-Line Interfaces.

Two administrative CLIs are available with SAS Decision Manager:

sas-dcmtransfer-cli
   enables you to transfer rule sets, rule flows, lookup tables, and decisions from a SAS 9.4 environment to a SAS Viya environment.

sas-scoreexecution-cli
   lists or deletes resources such as log files, code files, jobs, and SAS Cloud Analytic Services (CAS) tables that were not deleted when the associated rule set, decision, or model test was deleted.

To use these CLIs, you must create a profile and sign in to SAS Viya. For instructions, see “Command-Line Interface: Preliminary Instructions” in SAS Viya Administration: Using the Command-Line Interfaces.

Global Commands

The following commands are available for all CLIs:
authenticate
   enables you to log on to and log off from the environment. When you log on, this
   command prompts you for your SAS Viya user ID and password.

   Alias auth
   See “Global Command: Authenticate” in SAS Viya Administration: Using the
   Command-Line Interfaces
   “Command-Line Interface: Preliminary Instructions” in SAS Viya
   Administration: Using the Command-Line Interfaces

help
   displays the list of commands that are available for a CLI.

   Alias h
   Example sas-scoreexecution-cli help

plugins list
   displays the list of command line plugins for a CLI.

   Note No plugins are available for the SAS Decision Manager CLIs.
   Example sas-admin plugins list

profile
   creates or displays the connection profile that defines your SAS Viya deployment.
   This command asks you to enter the SAS Viya service endpoint, your preferred
   output type, and whether you want to enable colored output.

   Alias prof
   See “Global Command: Profile” in SAS Viya Administration: Using the
   Command-Line Interfaces
   “Command-Line Interface: Preliminary Instructions” in SAS Viya
   Administration: Using the Command-Line Interfaces

Global Options

The following options apply to all CLIs.

--colors-enabled true|false
   enables or disables ANSI-colored output.

   Note This option does not work in all environments.
   Tip You can set the environment variable $SAS_CLI_COLOR instead of
   specifying this option.

--help
   displays the list of commands and options that are available for the CLI. If this
   option is specified after a command, it displays the Help for that command.
**Global Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--insecure</td>
<td>allows connections to TLS sites without validating the server certificates.</td>
</tr>
<tr>
<td>--locale locale</td>
<td>specifies the locale to use, such as en (English) or de (German).</td>
</tr>
<tr>
<td>--output format</td>
<td>specifies the format for the output from the CLI. Specify text, json, or fulljson.</td>
</tr>
<tr>
<td>--profile profile_name</td>
<td>specifies the name of the profile to use.</td>
</tr>
<tr>
<td>--quiet</td>
<td>suppresses all output from the CLI except the data.</td>
</tr>
<tr>
<td>--sas-endpoint URL</td>
<td>specifies the URL to the SAS services.</td>
</tr>
<tr>
<td>--verbose</td>
<td>displays additional details about the commands that are processed, in addition to the output data.</td>
</tr>
<tr>
<td>--version</td>
<td>prints the version of the CLI.</td>
</tr>
</tbody>
</table>

**Example**

```bash
dsas-dcmtransfer-cli authenticate9x --help
```
Dictionary

sas-dcmtransfer-cli

Enables you to transfer rule sets, rule flows, lookup tables, and decisions from a SAS 9.4 environment to a SAS Viya environment.

Notes: Data tables, tests, comments, attachments, and version information are not transferred. Vocabularies are not transferred, but the terms used in rule sets and decisions are transferred within those objects. Lookup tables must be activated in the target environment. See “Activate a Lookup Table” in SAS Decision Manager: User’s Guide for more information.


See: “Global Options” on page 2
“Global Commands” on page 1

Syntax

sas-dcmtransfer-cli <global-options> command <command-options>

Commands

In addition to the global commands in “Global Commands” on page 1, you can specify the following commands:

authenticate9x subcommand <options>

logs you in to and out of the SAS 9.4 environment. You can specify the following subcommands:

login

logs you in to the SAS 9.4 environment. The login subcommand accepts these options:

--password password
specifies your password.

--service-endpoint protocol://host_name:port
specifies the URI for the SAS 9.4 environment.

Note Do not specify localhost. You must provide a fully qualified host name.

Example http://mySAS94server:7980

--user user-id
specifies your user ID.
Tip  If you do not specify any options, the login subcommand prompts you for the SAS 9.4 service endpoint, user ID, and password.

logout

logs you out of the SAS 9.4 environment.

Alias  auth9x

**export9x object-type <export-options>**

exports the specified object types from SAS 9.4. For the object type, specify one of the following: decisions, lookups, ruleflows, or rulesets. If you do not specify any options, the CLI prompts you for the required information.

**Note:** Beginning with SAS Decision Manager 5.1, SAS Decision Manager does not support rule flows. You cannot export complex rule flows. However, you can export simple rule flows from previous releases of SAS Decision Manager and import them as decisions into the current release. If you specify ruleflows, the rule flows are exported as decisions. If you specify decisions, information about the rule flows that are referenced in the decision is added to the mapping file for the decision.

**Restriction**  You cannot export objects that use DATAGRID variables.

**See**  “Options for the export9x Command” on page 5

“Transferring Content from a SAS 9.4 Environment to a SAS Viya Environment” on page 7

**import9x object-type <import-options>**

imports SAS Decision Manager objects into SAS Viya. For the object type, specify one of the following: decisions, lookups, ruleflows, or rulesets.

**See**  “Options for the import9x Command” on page 6

“Transferring Content from a SAS 9.4 Environment to a SAS Viya Environment” on page 7

### Options for the export9x Command

**--content filename**

specifies the name of the file to which you want to write the exported content.

Default  content.json

**--mappings filename**

specifies the name of the mapping file. This file describes the relationship between the exported content and any associated objects.

Default  mappings.json

See  “Modifying the Mapping File” on page 9

**--report filename**

specifies the file to which you want the CLI to write messages that are generated while the content is exported.

Default  export_report.json
--uri URI

specifies the URI for the objects that you want to export. This option is required.

You can specify the URI in any of the following forms.

--uri /SASBusinessRulesManagerWeb/rest/object-type?limit=214783647
--uri /SASDecisionManager/rest/decisions?limit=214783647

specifies all objects of the type object-type up to a maximum of 21478367. (This number is the maximum value for an integer.)

--uri /SASBusinessRulesManagerWeb/rest/object-type?filter_query
--uri /SASDecisionManager/rest/decisions?filter_query

specifies all objects that meet the filter criteria and that are of the type specified by the object-type argument.

Restriction This option is not available for lookup tables.

See “Query Parameters for the --uri Option” on page 9

--uri /SASBusinessRulesManagerWeb/rest/object-type/object-ID
--uri /SASDecisionManager/rest/decisions/object-ID

specifies the object with the ID object-ID that matches the type specified by the object-type argument.

--uri /SASWIPClientAccess/rest/navigation/814100/folder-ID

specifies all of the objects in the folder with the ID folder-ID that match the type specified by the object-type argument.

Restriction This form is not valid for exporting decisions.

Tip The number 814100 is the object type for a business rules folder.

--uri /SASWIPClientAccess/rest/navigation/814100/DCMFOLDER_ROOT_ID

specifies all objects in the business rules database that match the type specified by the object-type argument.

Restriction This form is not valid for exporting decisions.

Tip DCMFOLDER_ROOT_ID is the ID for the root folder.

--uri @filename.txt

specifies all of the objects that are identified by the URIs in the specified text file. The URIs in the text file must all be URIs for objects of the type specified by the object-type argument. Each URI must be on a separate line.

Options for the import9x Command

--content filename

specifies the file that contains the objects that you want to import.

Default content.json

--force <true | false>

specifies whether you want existing objects to be replaced if they already exist in the target SAS Viya environment. If you specify true, then existing objects with the same name are deleted and re-created from the imported content. If you specify false, error messages are generated for the duplicate objects.

The CLI searches the SAS Viya environment for existing objects before it imports new objects. The search criteria that the CLI uses to determine whether an object
already exists depends on the information in the mapping file. If the target
information for an object is available, the CLI uses it as the search criteria. If the
target information is not available, the CLI uses the source information. For example,
if you are importing a rule set, the CLI uses the target.id and
target.revisionId as the search criteria. If that information is not available, it
uses the source.name and source.folderpath.

Specifying --force without specify true or false is equivalent to specifying
--force true.

Alias   -f
Default false

See “Modifying the Mapping File” on page 9

--mappings filename
specifies the name of the mapping file.

Default mappings.json

See “Modifying the Mapping File” on page 9

--report filename
specifies the file to which you want the CLI to write messages that are generated
while the content is imported.

Default import_report.json

--target-folder-path /pathname
Specifies the folder in the SAS Viya environment that you want to import the
decisions into.

Default /Public

Restriction This option is valid only when the object type in the import9x
subcommand is decisions.

Details

Transferring Content from a SAS 9.4 Environment to a SAS Viya Environment

Note: See “Command-Line Interface: Preliminary Instructions” in SAS Viya
Administration: Using the Command-Line Interfaces for additional information.

To transfer business rules and decision content from a SAS 9.4 environment to a SAS
Viya environment:

1. Log on to the machine where SAS Viya is running. For example, on Linux systems,
you can use the ssh command:
   
   ssh -y machine-name

2. Change to the SAS Viya bin directory:
   
   cd /opt/sas/viya/home/bin

3. Create a default profile:
   
   ./sas-dcmtransfer-cli profile init
The CLI prompts you for the URL (service endpoint) for SAS Viya, for the output type that you want, and whether you want to enable color output.

4. Respond to the prompts as described in the following table.

<table>
<thead>
<tr>
<th>Service Endpoint</th>
<th>Specify the URL for the SAS Viya environment. Use the following format: communications-protocol://web-server-host-name:web-server-port</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>For example: <a href="http://host.example.com:7980">http://host.example.com:7980</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output type</th>
<th>Specify your preferred format for CLI output (text, json, or fulljson).</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Note: For information about the output types, see “Output Type” in SAS Viya Administration: Using the Command-Line Interfaces.</td>
</tr>
</tbody>
</table>

| Enable ANSI colored output | Specify whether to enable colored output (y or n). |

5. Log on to the SAS Viya environment:

   ./sas-dcmtransfer-cli auth login

   The CLI prompts you for the user ID and password for the SAS Viya environment.

6. Log on to the SAS 9.4 environment:

   ./sas-dcmtransfer-cli auth9x login

   The CLI prompts you for the URL (service endpoint) for the SAS 9.4 environment and the user ID and password for that environment.

7. Export the SAS 9.4 content:

   ./sas-dcmtransfer-cli export9x object-type <export-options>

   If you do not specify any options, the CLI prompts you for the required information.

   By default, the export9x command creates three files:

   • contents.json
     This file contains the exported content. Do not modify the contents of this file.

   • mappings.json
     This file describes the relationships between the exported content and any associated objects. For example, if you export rule sets, the mapping file contains information about any lookup tables that are referenced in the rule sets and all folders where the rule sets and lookup tables reside. You can modify some of the data in this file before you import content into the SAS Viya environment. For more information, see “Modifying the Mapping File” on page 9.

   • export_report.json
     This file contains any messages that were generated when the content was exported.
You can use the `--content`, `--mappings`, and `--report` options to change these filenames.

8. Import the content into the SAS Viya environment:

   ```
   ./sas-dcmtransfer-cli import9x object-type <import-options>
   ```

   By default, the `import9x` command looks for a file named contents.json that contains the content that was exported from SAS 9.4, and a file named mappings.json that describes the relationship between the exported content and any associated objects. If you specified the `--content` or `--mappings` options on the `export9x` command in Step 7 on page 8, then specify the same options on the `import9x` command.

   **Note:** When you view imported content in SAS Decision Manager, any errors in the content are marked with the error icon ☢.

9. Log off from the SAS 9.4 environment:

   ```
   ./sas-dcmtransfer-cli auth9x logout
   ```

10. Log off from the SAS Viya environment:

    ```
    ./sas-dcmtransfer-cli auth logout
    ```

**Query Parameters for the `--uri` Option**

The following table lists the query parameters that you can specify as part of the URI when you are exporting SAS 9.4 content. The parameters that are available depend on the object type that you are exporting.

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Available Parameters</th>
<th>Objects Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Sets</td>
<td>name=&quot;rule-set-name&quot;</td>
<td>Rule sets that match the specified name. This parameter is case sensitive.</td>
</tr>
<tr>
<td></td>
<td>vocabularyName=&quot;vocabulary-name&quot;</td>
<td>Rule sets that use the specified vocabulary</td>
</tr>
<tr>
<td>Rule flows</td>
<td>name=&quot;rule-flow-name&quot;</td>
<td>Rule flows that match the specified name. This parameter is case sensitive.</td>
</tr>
<tr>
<td></td>
<td>simpleOnly=&quot;true&quot;</td>
<td>Beginning with SAS Decision Manager 5.1, SAS Decision Manager does not support complex rule flows.</td>
</tr>
<tr>
<td>Decisions</td>
<td>name=&quot;decision-name&quot;</td>
<td>Decisions that match the specified name. This parameter is case sensitive.</td>
</tr>
<tr>
<td></td>
<td>vocabularyName=&quot;vocabulary-name&quot;</td>
<td>Decisions that use the specified vocabulary</td>
</tr>
</tbody>
</table>

**Modifying the Mapping File**

The mapping file provides information about exported content and any associated objects. The content of the mapping file depends on the content that is exported. For
example, if you export rule flows, then the mapping file contains information about the rule sets that are included in the rule flows and all folders where the rule sets and rule flows reside. If you export decisions, the mapping file contains information about the models and rule sets that are referenced in the decision. (Rule flows are exported as rule sets. Also, in SAS 9.4, decisions are not in folders, so the mapping file for decisions does not contain folder information.)

By default, rule flows that are imported as decisions, rule sets, and lookup tables are imported into a folder that has the same name as the folder in which it resided in the SAS 9.4 environment. To import the content into a different folder, edit the mapping file.

You can change some of the information in the mapping file. You can change the \texttt{target.\_*} fields, but do not change the \texttt{source.\_*} fields. For example, you can change the folders into which the content is imported and the IDs that are assigned to the imported content. The following table lists the fields in the mapping file for each object type and specifies whether you can change the field before the content is imported.

<table>
<thead>
<tr>
<th>Object</th>
<th>Field</th>
<th>Description</th>
<th>Can Be Changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>folders</td>
<td>source.folderPath</td>
<td>Folder path in the SAS 9.4 environment</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>target.folderPath</td>
<td>Folder path in the SAS Viya environment. By default, this field is set to the same path as source.folderPath.</td>
<td>Yes</td>
</tr>
<tr>
<td>lookups</td>
<td>source.id</td>
<td>Lookup table ID in the SAS 9.4 environment</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>source.name</td>
<td>Lookup table name</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>source.folderPath</td>
<td>Lookup folder path in the SAS 9.4 environment</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>target.id</td>
<td>Lookup table ID in the SAS Viya environment</td>
<td>Yes</td>
</tr>
<tr>
<td>ruleSets</td>
<td>source.id</td>
<td>Rule set ID in the SAS 9.4 environment</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>source.name</td>
<td>Rule set name</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>source.folderPath</td>
<td>Rule set folder path</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>target.id</td>
<td>Rule set ID in the SAS Viya environment</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>target.revisionId</td>
<td>Rule set version ID in the SAS Viya environment. If no version ID is specified, then the revision ID is set to \texttt{current}. When the rule set information is displayed in the user interface, the current version number is displayed.</td>
<td>Yes</td>
</tr>
<tr>
<td>models</td>
<td>source.id</td>
<td>Model ID in the SAS 9.4 environment</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>source.name</td>
<td>Model name</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>target.id</td>
<td>Model ID in the SAS Viya environment</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For example, if you export a rule set that references two lookup tables, the mappings file might look like this:
If you export rule flow that references two rule sets, the mapping file might look like this:

```json
{
  "folders": [
    {
      "source": {
        "folderPath": "/A"
      },
      "target": {
        "folderPath": "/B"
      }
    },
    {
      "source": {
        "id": 10093,
        "name": "LookupXYZ",
        "folderPath": "/A"
      },
      "target": {
        "id": ""
      }
    },
    {
      "source": {
        "id": 10094,
        "name": "LookupABC",
        "folderPath": "/B"
      },
      "target": {
        "id": "2e95e765-1242-4c06-ad5a-e05fdc3932fb"
      }
    }
  ],
  "ruleSets": [
    {
      "source": {
        "id": 10043,
        "name": "RuleSetXYZ",
        "folderPath": "/A"
      },
      "target": {
      }
    }
  ]
}
```
Examples

**Example 1**
The following command exports the rule set with the ID 10093:

```bash
./sas-dcmtransfer-cli export9x rulesets
--uri /SASBusinessRulesManagerWeb/rest/ruleSets/10093
```

**Example 2**
The following command exports the decision with the ID d246d283-fd80-4d37-96ad-18fd9cd508c8:

```bash
./sas-dcmtransfer-cli export9x decisions
--uri /SASDecisionManager/rest/decisions/d246d283-fd80-4d37-96ad-18fd9cd508c8
```

**Example 3**
The following command exports all rule flows:

```bash
./sas-dcmtransfer-cli export9x ruleflows
--uri /SASBusinessRulesManagerWeb/rest/ruleFlows?limit=2147483647
```

**Example 4**
The following command exports all lookup tables, writes the exported content to the file lookupContent.json, and writes the mapping information to the file lookupMappings.json:

```bash
./sas-dcmtransfer-cli export9x lookups --mappings lookupMappings.json
--content lookupContent.json
--uri /SASWIPClientAccess/rest/navigation/814100/DCMFOLDER_ROOT_ID
```

**Example 5**
The following command exports all rule sets that use the vocabulary named LoanApplications:

```bash
./sas-dcmtransfer-cli export9x rulesets
--uri /SASBusinessRulesManagerWeb/rest/ruleSets?vocabularyName="LoanApplications"
```
Example 6
The following command imports all of the rule sets in the file content.json. It uses the mapping file mappings.json:

```
./sas-dcmtransfer-cli import9x rulesets
```

Example 7
The following command imports all lookup tables in the file lookupContent.json. It uses the mapping file lookupMappings.json.

```
./sas-dcmtransfer-cli import9x lookups --content lookupContent.json
--mappings lookupMappings.json
```

Example 8
The following command imports all rule flows in the file content.json. It uses the mapping file mappings.json. The rule flows are imported as decisions.

```
./sas-dcmtransfer-cli import9x ruleflows
```

Example 9
The following command imports all rule flows as decisions. The decisions are imported into the folder named /LoanApplications. Any decision that has the same name as the rule flow that is being imported is replaced.

```
./sas-dcmtransfer-cli import9x ruleflows --force
--target-folder-path /LoanApplications
```

sas-scoreexecution-cli
Lists or deletes resources such as log files, code files, jobs, and SAS Cloud Analytic Services (CAS) tables that were not deleted when the associated rule set, decision, or model test was deleted.

**Requirement:** You must create a profile and sign in before using the CLI. See “Command-Line Interface: Preliminary Instructions” in SAS Viya Administration: Using the Command-Line Interfaces for information.

**See:**
“Global Options” on page 2
“Global Commands” on page 1

**Syntax**

```
sas-scoreexecution-cli <global-options> command <command-options>
```

**Commands**

In addition to the global commands in “Global Commands” on page 1, you can specify the following commands:

**list-hanging-resources**
lists the resources that are no longer used by the score execution service.

```
--file filename
```

specifies the file to write the list of resources to. You can use this file with the **remove-hanging-resources** command to delete the resources.

Alias   

```
-f
```
remove-hanging-resources
   deletes resources that are no longer used by the score execution service.
   
   --file filename
   Specifies the file that contains the URIs of the resources that you want to remove.
   
   Alias -f

Examples

Example 1
The following command displays detailed information about the unused resources:

   sas-scoreexecution-cli --output fulljson list-hanging-resources

Example 2
The following command displays information about unused resources in a table format:

   sas-scoreexecution-cli --output text list-hanging-resources

Example 3
The following command displays only the URIs of the unused resources:

   sas-scoreexecution-cli --output text --quiet list-hanging-resources

Example 4
The following command writes the URIs for unused resources to a file named uris.txt:

   sas-scoreexecution-cli list-hanging-resources --file uris.txt

Example 5
The following command deletes the unused resources listed in the file uris.txt:

   sas-scoreexecution-cli remove-hanging-resources --file uris.txt
Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore for additional books and resources.