Paper SAS3078-2019

What's New for Report Localization in SAS® Visual Analytics 8.3

Elizabeth Bales and James Holman, SAS Institute Inc.

ABSTRACT

Report localization was first introduced in SAS® Visual Analytics Designer 7.1. This feature made it easy for a user or a language specialist to translate a single report into multiple languages for report users. Many SAS® sample reports and customer reports have been translated using the 7.x releases of SAS Visual Analytics Designer. However, a user had to have a report-designing role to translate report content, which meant that the design of a report could be altered during localization. SAS Visual Analytics 8.3 is the first release based on SAS® Viya® that lets you localize reports. Starting in the 8.3 release, report content can be translated without the possibility of altering your report design. There is a new set of command-line interfaces (CLIs) in SAS Viya. The SAS reports CLI enables you to localize report content without requiring the report to be open for editing in SAS Visual Analytics. In this presentation, we introduce the SAS reports CLI features and demonstrate how you can use them to localize your reports.

INTRODUCTION

Reports created in early versions of SAS Visual Analytics only store labels and other descriptive text in a single language. If you wanted to localize those reports, you were required to copy the report, translate the text in the copy, and save a new report. You had to repeat the process for each language in which you wanted your report to display. The ability to store localizations in a single report was introduced in the 7.1 release of SAS Visual Analytics Designer (the designer). The paper "One Report, Many Languages: Using SAS Visual Analytics 7.1 to Localize Your Reports" explains how your reports can be translated so that they can display in many languages. All the translated text is stored in the report definition. Therefore, you do not need to create a separate copy of the report for each translation.

SAS Visual Analytics 8.3 takes report localization to a new level by moving management of report localization to your command line. This paper provides instructions that help you get started using the reports plug-in to the sas-admin CLI (SAS reports CLI) to simplify the report localization process.

REPORT LOCALIZATION IN SAS VISUAL ANALYTICS 7.X

To localize a report in SAS Visual Analytics 7.x, you must open your report in the designer and then select File->Localize Report. The Localize Report Text window displays the list of the localizable strings that are part of the report. You use the window to save the strings, also known as the *translation worksheet,* to a file. The text of the translation worksheet is in the original language.

After you save the translation worksheet to a file, you must create a copy of the file for each translation that you plan to apply to the report. There is no naming convention for the file name. However, including the locale in the file name helps you remember which translation the file contains. You also must remember to change the locale name on the first line of the worksheet. Otherwise, the translated text overwrites the text of the original language.

Once the text is translated, you must enter the designer again and go to the Localize Report Text window. Within the window, you must open the file containing the translated text and apply the translation to the report.

The workflow for localizing a report using the Localize Report Text window in the designer is simple. However, the report must be opened in the designer every time a new language is added to the report. This practice increases the risk that the report design could be changed.

REPORT LOCALIZATION IN SAS VISUAL ANALYTICS 8.3

In SAS Visual Analytics 8.3, the process of localizing a report is moved from the designer to a powerful set of CLIs. The *report translation* functionality is available through the SAS reports plug-in to the sas-admin command-line interface to the SAS Viya REST APIs. This functionality provides the flexibility to add new languages to your reports from your command line while protecting the design of your reports from the potential of unwanted updates.

You can use the commands in the SAS reports CLI to query for information about your reports, export and import the translation worksheets for one or more reports, and write scripts to automate each process. This paper demonstrates the flexibility of the SAS Reports CLI.

GETTING STARTED WITH THE SAS CLI

The SAS CLIs are available in your SAS Viya deployment. In SAS Viya, a CLI is an interface to a SAS Viya REST service. SAS Viya uses a plug-in approach for the various command-line interfaces. Getting started involves launching the sas-admin CLI executable directly.

If you have access to machines in the SAS Viya deployment, the CLIs are present if the command-line tools were installed. The sas-admin CLI can be found in the following locations:

Linux: /opt/sas/viya/home/bin

Windows: \Program Files\SAS\Viya\bin

If you prefer to use a unique copy of the sas-admin CLI, you can download it from the SAS Support / Downloads and Hot Fixes page at http://support.sas.com/downloads/package.htm?pid=2133. The SAS Viya CLIs are supported on 64-bit Linux, Windows, and Mac OSX regardless of where your SAS Viya server is installed.

Note: You must have a valid SAS profile to download the sas-admin CLI.

Profile setup

Whether you use the copy of the sas-admin CLI that is included in your SAS Viya deployment or download your own copy, your first task is to set up your profile. The examples in this paper assume that you are running the commands from the directory where the sas-admin CLI is installed. If that is not the case for you, making sas-admin available in your path is sufficient.

You need to run the following command only once to set up the default profile for your SAS Viya deployment:

```
sas-admin profile init
```

The command prompts you for the following information:

- Service Endpoint: This is the URL of your SAS Viya deployment. For example: https://host.example.com.
- Output Type: The options for this prompt are text, json, and fulljson. Translation worksheets are always returned as text files regardless of this profile setting. Therefore, you can select the output type that works best for other commands that you plan to run.
- Enable ANSI colored output: Enter y or n. Either value is fine for localizing reports.

If you need to change the configuration of your profile, you can run the profile init command again. You can also set up a named profile and provide responses that apply to your new configuration. To create a named profile, add the option --profile *profilename* to all your sas-admin commands. For example, the following command sets up a profile named *adminuser*.

```
sas-admin --profile adminuser profile init
```

Logging in

Once your profile is set up, you need to log on and provide your credentials by running the following command:

```
sas-admin auth login
```

The login command prompts you for your user ID and password. Once you successfully log on, SAS Viya returns a login token that remains active for the duration set by your environment. By default, the duration for login tokens is 12 hours.

The "Preliminary Instructions" chapter of the *SAS Viya 3.4 Administration: Using the Command-Line Interfaces* provides detailed information about the initialization and login process.

Reports interface

The CLI for the SAS Reports service APIs are installed on the server where SAS Viya is located. If you are running the sas-admin CLI on the SAS Viya server, then you can run the SAS reports CLI commands after you log on. However, if you choose to install your own copy of the sas-admin CLI, you must also download the reports plug-in. After you acquire the login token, run the following command:

```
sas-admin plugins install –repo SAS reports
```

The "Preliminary Instructions" topic in *SAS Viya 3.4 Administration: Using the Command-Line Interfaces* provides detailed information about downloading plug-ins.

USING THE SAS REPORTS CLI

Get the report ID

After you install the SAS reports CLI, you need to find the report identifier (report ID) for each report that you want to localize. The list command in the SAS reports CLI displays a list of reports that have been created in SAS Visual Analytics. The output of the list command also shows the unique report ID for each report.

Note: Some commands in the reports interface, such as list, require the authenticated user to be a member of the SAS Administrator group.

By default, the list command displays an alphabetized list of the first 20 reports in the SAS Viya deployment. If your organization has created dozens, or even hundreds, of reports, the information for your report might not be displayed. The list command offers options that allow you to filter the results that are displayed.

Every command in the sas-admin CLI has integrated help that shows proper usage for the command and the available options. Figure 1 shows the Help documentation for the list command, including the available options.

```
$ ./sas-admin reports list --help
NAME:
   sas-reports list - Lists the reports in a deployment.

USAGE:
   sas-reports list [command options] [arguments...]

COMMANDS:

OPTIONS:
   --created-after             Returns reports that are created after the specified date and
                               time. For example: 2017-05-23 2017-05-23T18:15Z
   --created-before            Returns reports that are created before the specified date and
                               time. For example: 2017-05-23 2017-05-23T18:15Z
   --created-by                Returns reports that are created by the specified user.
   --created-by-contains       Returns reports that are created by one or more users whose names
                               contain the specified pattern.
   --details                   Shows more detailed output, including the following attributes:
                               createdBy, creationTimeStamp, modifiedBy, modifiedTimeStamp
   --help, -h                  Shows help.
   --id                        Returns the report with the specified ID.
   --limit "20"                Specifies the maximum number of reports to return. The default
                               value is 20.
   --modified-after            Returns reports that are modified after the specified date and
                               time. For example: 2017-05-23 2017-05-23T18:15Z
   --modified-before           Returns reports that are modified before the specified date and
                               time. For example: 2017-05-23 2017-05-23T18:15Z
   --modified-by               Returns reports that are created by the specified user.
   --modified-by-contains      Returns reports that are modified by one or more users whose
                               names contain the specified pattern.
   --name                      Returns the report with the specified name.
   --name-contains             Returns reports whose names contain the specified pattern.
   --name-starts-with          Returns reports whose names start with the specified pattern.
   --sort-by "name"            Specifies the sort criteria for returned reports. By default, the
                               list is sorted by name and is in ascending order. Prefix the
                               field with a tilde(~) to change the sort order to descending. The
                               valid report attributes are as follows: id, name, createdBy,
                               creationTimeStamp, modifiedBy, modifiedTimeStamp
   --start "0"                 Specifies the 0-based offset of the first report to return. The
                               default value is 0.
```

Figure 1 Help Documentation for the SAS Reports CLI LIST Command

For example, suppose you have created a set of reports to show the fuel efficiency for specific types of vehicles. The name of all your reports start with the word “Vehicle”. Figure 2 demonstrates the list command using the --name-starts-with option to display your vehicle reports:

```
$ ./sas-admin reports list --name-starts-with "Vehicle"
Name                         Folder    Report ID
Vehicle Comparison           /Public   08ca950d-f43e-4be8-9738-ecc11d77b4f3
Vehicle Performance Report   /Public   433e4e74-d1e7-45dd-8d2f-582216f7a5b3
```

Figure 2 Reports with a Name that Starts with "Vehicle"

Create the translation worksheets

Now that you have the report ID, you can use it to get the translatable strings for your report. The export command in the SAS reports CLI exports the set of strings from the report in the SAS Visual Analytics repository and places them in a file in your file system. Figure 3 shows the Help documentation for the export command:

```
$ ./sas-admin reports translations export --help
NAME:
   sas-reports translations export - Exports a translation worksheet based on the specified
report and locale.

USAGE:
   sas-reports translations export [command options] [arguments...]

COMMANDS:

OPTIONS:
   --help, -h          Shows help.
   --output-location   Specifies the output location for the translation worksheet.
   --report-id         Specifies the ID of the report for which you want to export the
                       translation worksheet.
   --report-locale     Specifies the locale of the report for which you want to export the
                       translation worksheet.
```

Figure 3 Help Documentation for the SAS Reports CLI EXPORT command

The export command creates a file in the specified output location that contains a translation worksheet. The file name includes the name of the report with spaces preserved, an underscore character (_), and the locale ID. The file extension is *.reports*.

The locale ID is specified by a 2-letter ISO 639 language code. An optional 2-letter ISO 3166 region code can also be included following the language code and a hyphen (-) character. For example, the locale ID for English (United States) is *en-US*.

In addition to being appended to the file name, the locale ID is included in the metadata of the translation worksheet. As a best practice, SAS recommends running the export command once for each translation that you plan to create for your report. The command creates one file per translation. The worksheet in every file contains text in the original language. However, the locale name in the metadata and file name will be the locale for your translation and does not need to be updated.

For example, your "Vehicle Comparison" report is localized for French, Polish, and Japanese. You need to create a worksheet file for each language so that you can send it to the translators. Figure 4 shows the export commands that create the "Vehicle Comparison" worksheet files for each language.

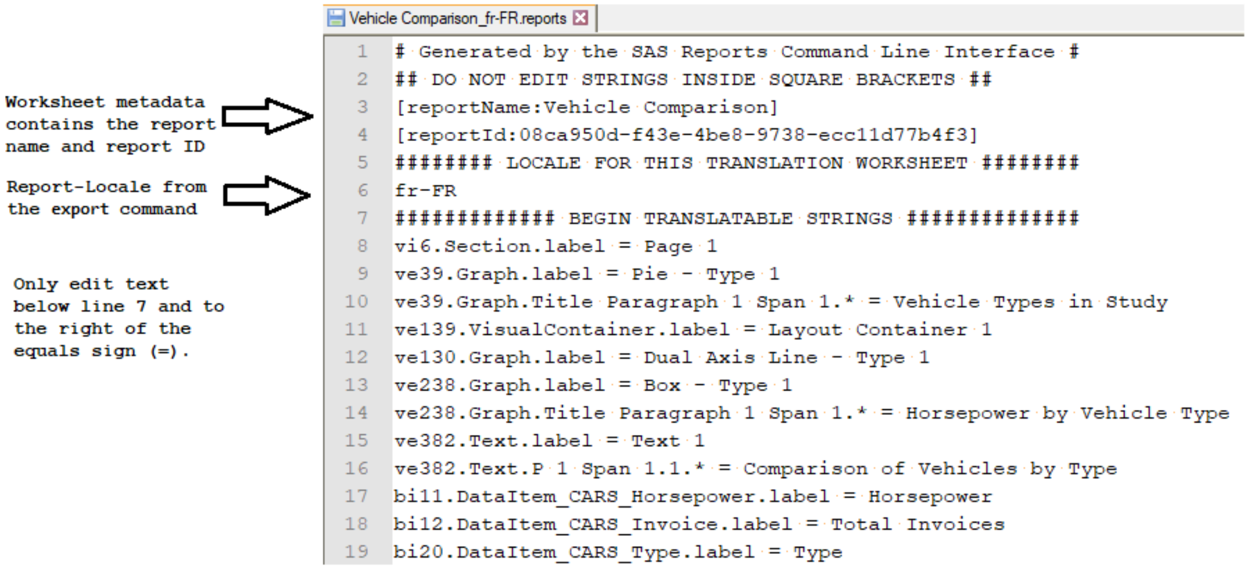
```
$ ./sas-admin reports translations export --output-location vehicle --report-id 08ca950d-f43e-
4be8-9738-ecc11d77b4f3 --report-locale fr-FR
Processing the report "Vehicle Comparison" with ID of "08ca950d-f43e-4be8-9738-ecc11d77b4f3".

Processing complete.

Translatable text has been exported.

Report Name       Vehicle Comparison
Report ID         08ca950d-f43e-4be8-9738-ecc11d77b4f3
Output Filename   vehicle/Vehicle Comparison_fr-FR.reports

$ ./sas-admin reports translations export --output-location vehicle --report-id 08ca950d-f43e-
4be8-9738-ecc11d77b4f3 --report-locale pl-PL
Processing the report "Vehicle Comparison" with ID of "08ca950d-f43e-4be8-9738-ecc11d77b4f3".

Processing complete.

Translatable text has been exported.

Report Name       Vehicle Comparison
Report ID         08ca950d-f43e-4be8-9738-ecc11d77b4f3
Output Filename   vehicle/Vehicle Comparison_pl-PL.reports

$ ./sas-admin reports translations export --output-location vehicle --report-id 08ca950d-f43e-
4be8-9738-ecc11d77b4f3 --report-locale ja-JP
Processing the report "Vehicle Comparison" with ID of "08ca950d-f43e-4be8-9738-ecc11d77b4f3".

Processing complete.

Translatable text has been exported.

Report Name       Vehicle Comparison
Report ID         08ca950d-f43e-4be8-9738-ecc11d77b4f3
Output Filename   vehicle/Vehicle Comparison_ja-JP.reports

$ ls vehicle
Vehicle Comparison_fr-FR.reports  Vehicle Comparison_ja-JP.reports  Vehicle Comparison_pl-PL.reports
```

Figure 4 Example EXPORT commands for French, Polish, and Japanese

Note: The `export` command always returns text in the language of the original report. For example, if you created the Vehicle Comparison report in English, the strings in all of the *.reports* files are English. The locale ID in the worksheet is the only difference between the files. The benefit of this practice is that the file name and worksheet locale are already correct.

Translation worksheet

Each worksheet includes strings from the report, such as labels, tooltips, and other descriptive text. The worksheet also includes metadata about the report, such as the report name, report ID, and the locale ID that was specified on the `export` command line. The worksheet does not include content from the data sources included in your report. For example, Figure 5 shows the translation worksheet for the Vehicle Comparison report. The worksheet was created by the `export` command shown in Figure 4 for the fr-FR locale:



Worksheet metadata contains the report name and report ID

Report-Locale from the export command

Only edit text below line 7 and to the right of the equals sign (=).

```
 1  # Generated by the SAS Reports Command Line Interface #
 2  ## DO NOT EDIT STRINGS INSIDE SQUARE BRACKETS ##
 3  [reportName:Vehicle Comparison]
 4  [reportId:08ca950d-f43e-4be8-9738-ecc11d77b4f3]
 5  ######## LOCALE FOR THIS TRANSLATION WORKSHEET ########
 6  fr-FR
 7  ############## BEGIN TRANSLATABLE STRINGS ##############
 8  vi6.Section.label = Page 1
 9  ve39.Graph.label = Pie - Type 1
10  ve39.Graph.Title Paragraph 1 Span 1.* = Vehicle Types in Study
11  ve139.VisualContainer.label = Layout Container 1
12  ve130.Graph.label = Dual Axis Line - Type 1
13  ve238.Graph.label = Box - Type 1
14  ve238.Graph.Title Paragraph 1 Span 1.* = Horsepower by Vehicle Type
15  ve382.Text.label = Text 1
16  ve382.Text.P 1 Span 1.1.* = Comparison of Vehicles by Type
17  bi11.DataItem_CARS_Horsepower.label = Horsepower
18  bi12.DataItem_CARS_Invoice.label = Total Invoices
19  bi20.DataItem_CARS_Type.label = Type
```

Figure 5 Translation Worksheet for the Vehicle Comparison Report

A translator should modify only the text to the right of the equal sign. The equal sign, space after the equal sign, and all the text to the left of the equal sign should never be edited.

Also, you should not modify the report metadata that is included in the first few lines of the worksheet. If you need to change the local name, use the locale name format documented in the *Create the translation worksheets* section. However, if you followed the best practice to specify the locale of the translation worksheet on the export command, the locale name is correct in the worksheet.

Importing translated worksheets

After your worksheets have been translated, run the import command. Specify the name of the file that contains the worksheet using the source-file option. You do not need to specify the report ID or the report locale since both are included in the worksheet. However, the locale ID must be included as part of the file name. If you ran the export command for every translation that you add to your report, the correct locale is included in the file name.

Here is an example of the import command used to import the French localization into the Toy Sales report:

```
sas-admin reports translations import
    --source-file "worksheets/Vehicle Comparison_fr-FR.reports"
```

After you update the report to add the French translations, the report text will be displayed as French in a browser with the language set to French.

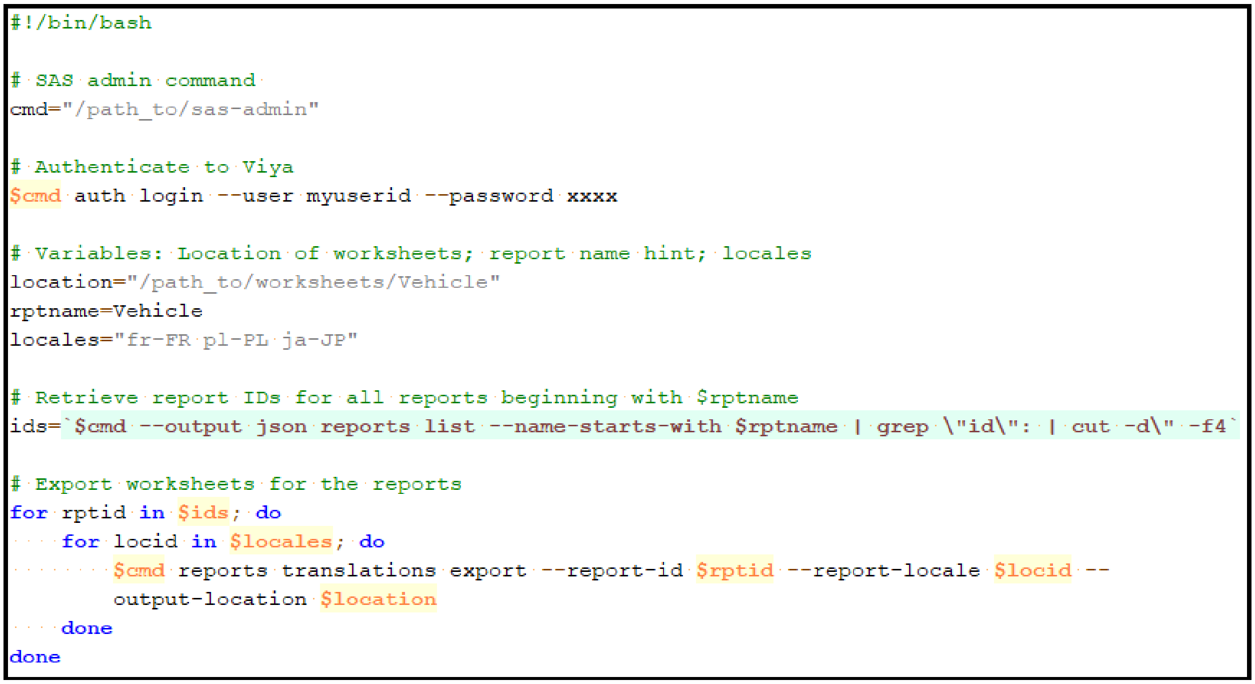
Managing worksheets with scripts

Once you identify the set of reports to translate, you will need to write the export and import commands to run for every locale. To make this process easier, your sas-admin CLI commands can be placed in scripts to ensure that the location, locale names, report ID, and so on, are specified the same way for each command.

For example, in previous examples the individual commands in the SAS reports CLI were run to do the following:

1. Log on to SAS Viya,
2. Acquire the report ID for the "Vehicle Comparison" report
3. Fetch translation worksheets for the French (fr-FR), Polish (pl-PL), and Japanese (ja-JP) locales for that report.

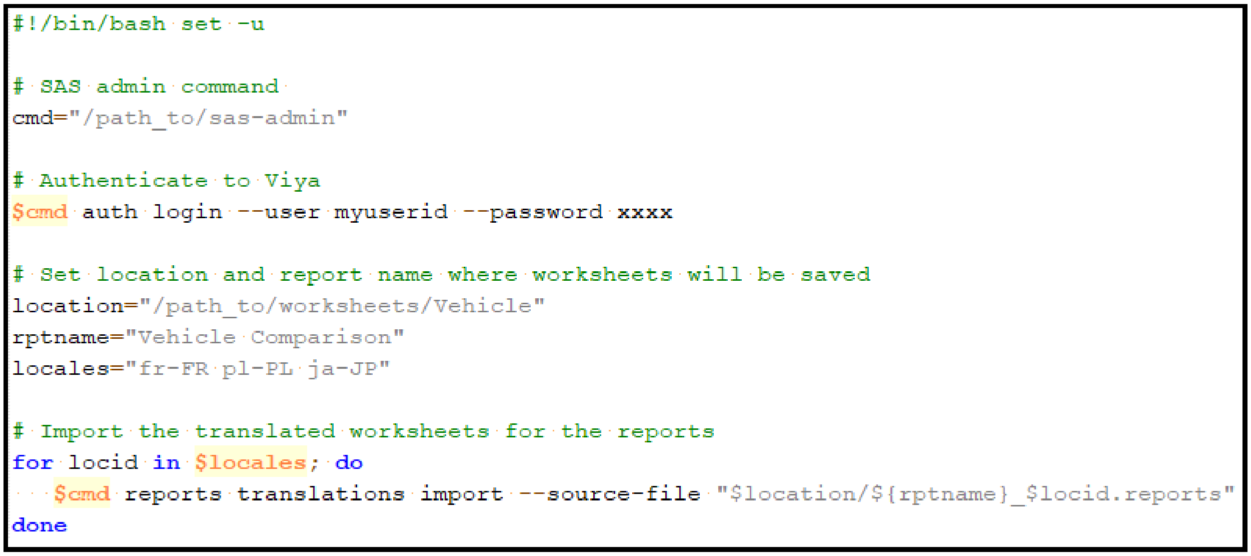
The sample script shown in Figure 6 runs all commands that are required to fetch the worksheets for the "Vehicle Comparison" report for the same set of locales. The script assumes that there is a default profile defined with the connection options for the SAS Viya deployment containing the "Vehicle" reports.



```
#!/bin/bash

# SAS admin command
cmd="/path_to/sas-admin"

# Authenticate to Viya
$cmd auth login --user myuserid --password xxxx

# Variables: Location of worksheets; report name hint; locales
location="/path_to/worksheets/Vehicle"
rptname=Vehicle
locales="fr-FR pl-PL ja-JP"

# Retrieve report IDs for all reports beginning with $rptname
ids=`$cmd --output json reports list --name-starts-with $rptname | grep \"id\": | cut -d\" -f4`

# Export worksheets for the reports
for rptid in $ids; do
    for locid in $locales; do
        $cmd reports translations export --report-id $rptid --report-locale $locid --
        output-location $location
    done
done
```

Figure 6 Script to Export Translation Worksheets

Figure 7 shows a script that runs all of the commands required to apply localizations to the "Vehicle Comparison" report. The script assumes that there is a default profile defined with the connection options for the SAS Viya deployment containing the "Vehicle" reports.



```
#!/bin/bash set -u

# SAS admin command
cmd="/path_to/sas-admin"

# Authenticate to Viya
$cmd auth login --user myuserid --password xxxx

# Set location and report name where worksheets will be saved
location="/path_to/worksheets/Vehicle"
rptname="Vehicle Comparison"
locales="fr-FR pl-PL ja-JP"

# Import the translated worksheets for the reports
for locid in $locales; do
   $cmd reports translations import --source-file "$location/${rptname}_$locid.reports"
done
```

Figure 7 Script to Import Translation Worksheets

VIEWING THE LOCALIZED REPORT

After the translation worksheets are applied, the report should still look the same to you if your browser locale is the same as the original language of the report. However, if you change the browser locale to match the locale of a worksheet that you applied, you will see the report text in a different language. For example, Figure 8 shows the "Vehicle Comparison" report displayed in the original language as well as in browsers where the language is set to the French and Japanese.

Note: If you change the browser language, you must sign out from SAS Visual Analytics and sign in again before the browser language takes effect.

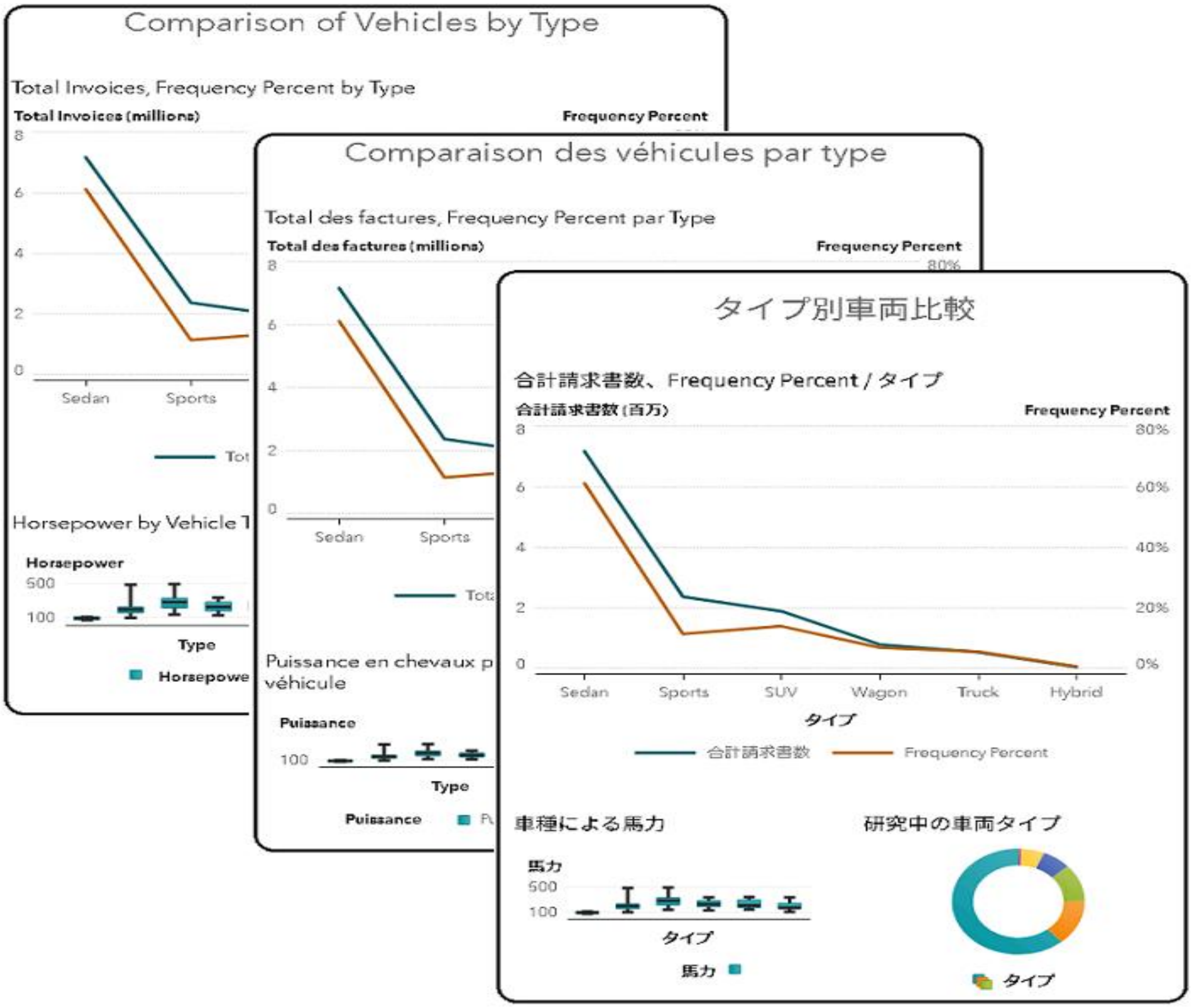


Figure 8 Vehicle Comparison Report in English, French, and Japanese

ADDITIONAL TIPS

PREPARING FOR FUTURE TRANSLATIONS

If you know which translations you will add to your report, it is a best practice to create one file per translation. However, if you are not sure how many translations you will need to add, fetch the worksheet for the original locale of the report. When you decide to add a new localization to the report, simply copy the worksheet file and change the locale name to the new locale that you want to add. Remember to edit the locale name in the worksheet metadata. The locale inside the worksheet must match the locale in the file name. Locale names must follow the guidelines described in the *Create the translation worksheets* section.

For example, the report titled “Vehicle Performance Report” was created in English and will be translated. Run the `export` command to fetch the worksheet for the default locale, en-US. The command creates a file called “Vehicle Performance Reports_en-US.reports”. After you create the English worksheet, a decision is made to translate the report to German. Here are the steps to prepare for the German translation:

1. Create a copy of the worksheet file.
2. Change the name of the new file to "Vehicle Performance Report_de-DE.reports", using an underscore (_) character to separate the report name from the language or locale name.
3. Edit the file, change the locale name in the worksheet from "en-US" to "de_DE", and save the file.

The worksheet is now ready to translate. After the file is translated, run the `import` command to apply the German localization to the Vehicle Performance Report.

HELP FOR SAS-ADMIN CLI COMMANDS

The `help` option is available for every sas-admin CLI command. For example, Figure 9 and Figure 10 show the Help documentation the SAS reports CLI and `translations` command, respectively.

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

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

COMMANDS:
   clear-results-cache  Clears the results cache of report data for SAS Visual Analytics reports
                        on SAS Viya.
   delete               Deletes the specified report by ID.
   help, h              Shows a list of commands or help for one command.
   list                 Lists the reports in a deployment.
   show-info            Shows information about the specified report.
   themes               Manages the themes that are used for SAS Visual Analytics reports.
   translations         Manages the translations worksheets associated with a specified SAS
                        Visual Analytics report.
```

Figure 9 Help Documentation for the SAS Reports CLI Interface

```
$ ./sas-admin reports translations --help
NAME:
   sas-reports translations - Manages the translations worksheets associated with a specified SAS
                              Visual Analytics report.

USAGE:
   sas-reports translations command [command options] [arguments...]

COMMANDS:
   export       Exports a translation worksheet based on the specified report and locale.
   help, h      Shows a list of commands or help for one command.
   import       Imports a translation worksheet.

OPTIONS:
   --help, -h   Shows help.
```

Figure 10 Help Documentation for the SAS Reports CLI TRANSLATIONS Command

CONCLUSION

The commands in the SAS reports CLI provide you with the flexibility to add and update report translations from your command line. Using these commands to manage report translations protects the integrity of your report design and enables you to localize your SAS Visual Analytics reports so that you can make them available for everyone in your organization.

REFERENCES

Ballard, William, and Bales, Elizabeth. 2015. "One Report, Many Languages: Using SAS Visual Analytics to Localize Your Reports". Available at https://support.sas.com/documentation/onlinedoc/va/7.1/LocalizeReports.pdf

Hamrick, Danny. 2017. "Command-Line Administration in SAS Viya" Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/2233-2018.pdf

ISO 639 Language Codes. Available at http://www.iso.org/iso/language_codes.htm

ISO 3155 Country Codes. Available at http://www.iso.org/iso/country_codes.htm

ACKNOWLEDGEMENTS

We would like to express our gratitude to the people who made report localization a reality in SAS Visual Analytics. Will Ballard designed the localization support for SAS Visual Analytics reports. Harry Maxwell created the SAS APIs that make report localization possible in SAS Visual Analytics 8. Without support from Jeff Diamond, the project would never have been accomplished. We would also like to thank Penny Downey, Joy Lashley and Dan Harrell for their timely and helpful review comments.

Finally, thanks to the localization teams at SAS who worked with us to design localization tools for a command-line interface that real translators can to use.

RECOMMENDED READING

- *SAS® Viya® 3.4 Administration: Using the Command-Line Interfaces*

CONTACT INFORMATION

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

Elizabeth Bales

SAS Institute Inc.

Elizabeth.Bales@sas.com

James Holman

SAS Institute Inc.

James.Holman@sas.com