



# SAS<sup>®</sup> Viya<sup>®</sup> 3.3 Quick Start

## Quick Start

### About the SAS Viya Programming Documentation

This collection of documents provides task-based programming examples, syntax, and concepts for products in SAS Viya, such as:

- SAS Visual Analytics 8.2 programming
- SAS Visual Statistics 8.2 programming
- SAS Visual Data Mining and Machine Learning 8.2 programming

**Note:** On most hosts, SAS 9.4M5 is tightly integrated with SAS Viya. See [SAS 9.4M5 Integration with SAS Viya](#) in *What's New in Base SAS: Details*. (The exceptions are z/OS and 32-bit Windows.)

To search this collection, click . For search tips, click , and select **Help Tips**.

**TIP** If you are viewing this document in a stand-alone context, [access the collection](#) before you use the preceding information.

**Note:** Your site must license and install one or more SAS Viya products to access this functionality.

## Orientation

This software supports analytical data preparation, variable transformations, exploratory analysis, analytical modeling, integrated model comparison, and scoring. Here are the main components:

SAS 9.4M5	SAS 9.4M5 is integrated with SAS Viya. SAS programs submitted from SAS 9.4M5 clients can call procedures that are unique to SAS Viya, as well as CAS-enabled DATA step code and procedures that have been modified to leverage a CAS server and its tables.
SAS Viya	The third generation of high-performance in-memory analytics.
SAS Studio	The integrated programming environment.
SAS Cloud Analytic Services (CAS)	The analytic engine. CAS uses high-performance, multithreaded analytic code to rapidly process requests against data of any size.

SAS Visual Analytics	Programming tools that provide baseline functionality, including reporting and basic analytics, such as: <ul style="list-style-type: none"> <li>■ analytical data preparation</li> <li>■ variable transformations</li> <li>■ exploratory analysis</li> <li>■ descriptive statistics</li> </ul>
SAS Visual Statistics	An additional set of advanced analytic functionality that builds on SAS Visual Analytics, such as: <ul style="list-style-type: none"> <li>■ building predictive models</li> <li>■ integrated model comparison</li> </ul>
SAS Visual Data Mining and Machine Learning	An additional set of advanced analytic functionality that builds on SAS Visual Statistics, such as: <ul style="list-style-type: none"> <li>■ tune machine learning algorithm hyperparameters</li> <li>■ advanced statistical operations</li> <li>■ analyzing complex data</li> </ul>
SAS Econometrics	A set of functionality that provides techniques to model complex business and economic scenarios and to analyze the dynamic impact that specific events might have over time.
SAS Visual Forecasting	Provides automatic variable, event, and model selection. It then automatically generates your forecasts.
SAS Visual Text Analytics	A text analytics framework combining text mining, contextual extraction, categorization, sentiment analysis and search.
SAS Optimization	A set of procedures for exploring models of distribution networks, production systems, resource allocation problems, and scheduling problems using the tools of operations research.

CAS-based procedures run against data that is in CAS. For example, before you can use CAS to work with a SAS data set, you must load that data set into CAS. The following instructions demonstrate basic mechanics.

**TIP** If you cannot securely sign in and start a CAS session, contact your administrator or see the troubleshooting topic in [SAS Viya Administration: Identity Management](#).

## Demonstration: Load Personal Data

- 1 Sign in to SAS Studio.
  - a Open SAS Studio from a URL that is provided by your SAS administrator. For example, you might enter: `https://webserver-host-name/SASStudio`.
  - b Enter the credentials for your operating system account.
- 2 Start a CAS session.
  - a In the navigation pane, open the **Snippets** section.
  - b Select **Snippets** ⇒ **Cloud Analytic Services**.
  - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.

d In the toolbar, click  to run the **New CAS Session** code.

### 3 Load a table.

a In the navigation pane, right-click **Load data to Caslib** and select **Open**.

b In the code editor, edit the SAS data set section so that it looks like this:

```
PROC CASUTIL;
  LOAD DATA=sashelp.cars OUTCASLIB="CASUSER"
  CASOUT="demoTable" PROMOTE;
RUN;
```

**TIP** CASUSER is your personal caslib. It is available across your sessions, but it is always and intrinsically private to you. You cannot enable other users to access it.

c Select the preceding code. In the toolbar, click  to run only the four lines of selected code.

### 4 Verify that you can access the loaded data.

a In the navigation pane, right-click **Generate SAS librefs for Caslibs** and select **Open**.

b In the toolbar, click  to run the code.

c In the navigation pane, open the **Libraries** section.

d Select **My Libraries** ⇒ **CASUSER**.

e Double-click **demoTable** to open it.

## Demonstration: Provide Shared Data

1 Sign in to SAS Studio and start a CAS session, if you have not already done so.

2 Create a container for shared CAS data.

a In the navigation pane, open the **Snippets** section.

b Select **Snippets** ⇒ **Cloud Analytic Services**.

c Right-click **New Caslib for Path** and select **Open**.

d In the code editor, edit the snippet so that it looks like this:

```
CASLIB demoCas PATH="/filePath/" DATASOURCE=(SRCTYPE="path") GLOBAL;
```

**Note:** Enter a path that is relative to and accessible from your CAS server. You can reference an empty directory.

e Click  to run the code.

**Note:** If an error indicates that you do not have permission to create global caslibs, see [“Adjust Caslib Management Privileges” in SAS Viya Administration: SAS Cloud Analytic Services](#). Initially, only administrators can add global caslibs. An administrator can enable non-administrators to add global caslibs.

3 Give all users Read access to the new caslib.

## 4

- a In the banner, click , and select **CAS Administration**.
  - b In the sign-in window, enter your operating system credentials.
  - c In CAS Server Monitor, beneath the **SAS Cloud Analytic Services** banner, click .
  - d On the **Configuration** page, select **Access Controls**.
  - e In the **Caslibs** list, select the caslib.
  - f In the upper right, click **Edit**.
  - g In the **Edit Access Controls** window, adjust settings as follows:
    - In the **Read Info** row for Authenticated Users, select the **Grant** radio button.
    - Click **Add Row**. In the new row at the end of the page, select **Authenticated Users**, the **Grant** radio button, and the **Select** activity.
  - h Click **OK** to save your changes.
  - i Under **Access Controls**, review the results of your changes.
  - j At the right edge of the banner, click your user name, and select **Sign Out**.
- 4 Load data to the new caslib.
- a In the navigation pane in SAS Studio, right-click the snippet **Load data to Caslib** and select **Open**.
  - b In the code editor, edit the SAS data set section so that it looks like this:

```
PROC CASUTIL;  
  LOAD DATA=sashelp.cars OUTCASLIB="demoCas"  
  CASOUT="demoTable" PROMOTE;  
RUN;
```
  - c Select the preceding code. In the toolbar, click  to run only the four lines of selected code.
- 5 Verify that other users can see the data. For example, ask them to complete these steps:
- a Sign in to SAS Studio, start a CAS session, and run the **Generate SAS librefs for Caslibs** snippet.
  - b In the navigation pane, open the **Libraries** section.
  - c Select **My Libraries** ⇒ **demoCas**.
  - d Double-click **demoTable** to open it.

### Tip: Automatically Connect and Generate Librefs

For convenience, you can configure SAS Studio to perform the following tasks each time you sign in:

- Start a CAS session.
- Generate SAS librefs for existing caslibs that have names that are no more than eight characters long.

Complete the following steps:

- 1 In the SAS Studio banner, click , and select **Edit Autoexec File**.
- 2 On the **Autoexec.sas** tab of the Edit Autoexec File window, paste the following code:

```
cas casauto;
caslib _all_ assign;
```

- 3 Run the code.
- 4 Save the code.

## Documentation: References by Task

Task	Refer to:
Do you need to migrate your data to UTF-8?	<a href="#">Migrating Data to UTF-8 for SAS Viya</a>
Access your data	<a href="#">DATA Step Examples</a> <a href="#">Common Tasks for Accessing and Manipulating Data</a>
Manipulate your data	<a href="#">Data Step Basics</a>
Prepare, Model, Assess	<a href="#">MDSUMMARY Procedure</a>  <a href="#">SAS Visual Data Mining and Machine Learning: Data Mining and Machine Learning Procedures</a> <a href="#">SAS Visual Data Mining and Machine Learning: Statistical Procedures</a>
Graph your output data	<a href="#">Introduction to SAS Platform Graphing</a>
Program with the Lua, Python, R, or CASL languages	<a href="#">Getting Started with SAS Viya for Lua</a> <a href="#">Getting Started with CASL Programming</a> <a href="#">Getting Started with SAS Viya for Python</a> <a href="#">Getting Started with SAS Viya for R</a>
Learn more about SAS Viya and SAS 9 interactions.	<a href="#">An Introduction to SAS Viya Programming</a> <a href="#">SAS Guide to Software Updates</a>

