# Contents

*What's New in SAS Studio 3.8* .................................................. vii

## Chapter 1 • Introduction to SAS Studio ........................................ 1
- About SAS Studio ........................................................................ 1
- Using SAS Studio ...................................................................... 2

## Chapter 2 • Working with Programs ........................................... 15
- About the Code Editor ................................................................. 15
- Opening and Creating Programs .................................................. 16
- Working with Code Snippets ....................................................... 30
- Customizing the Code Editor ....................................................... 40

## Chapter 3 • Working with Queries ............................................. 41
- What Is a Query? ........................................................................ 41
- Creating a Query ........................................................................ 42
- Understanding Joins ................................................................... 43
- Selecting Data ........................................................................... 46
- Filtering Data ............................................................................. 50
- Managing Output ........................................................................ 53

## Chapter 4 • Working with Process Flows .................................... 59
- Understanding Process Flows ....................................................... 59
- Add Data to the Process Flow ...................................................... 63
- Adding a SAS Program to the Process Flow .................................. 64
- Adding a Query to a Process Flow ................................................ 69
- Add a Task to a Process Flow ........................................................ 70
- Understanding Subflows ............................................................... 71
- Linking Nodes in a Process Flow ................................................... 73
- Generating Code from a Process Flow .......................................... 73
- Running a Process Flow ............................................................... 74
- Save a Process Flow .................................................................... 74

## Chapter 5 • Working with Data ................................................... 75
- About the Table Viewer ............................................................... 75
- Opening and Viewing Data .......................................................... 77
- Viewing the Code That Is Used to Create a Table ......................... 79
- Filtering and Sorting Data ............................................................. 80
- Importing Data ........................................................................... 83
- Exporting Data ........................................................................... 92

## Chapter 6 • Working with Results ................................................ 93
- Viewing Results .......................................................................... 93
- Default SAS Studio Output ......................................................... 93
- Sending Your Results to Another User ......................................... 96
- About the SAS Output Delivery System ....................................... 97
- About SAS ODS Statistical Graphics .......................................... 97

## Chapter 7 • Understanding Git Integration in SAS Studio ............. 101
- About Git Integration in SAS Studio ............................................ 101
Creating a Git Profile .................................................. 101
Cloning a Repository ..................................................... 103
Committing Changes to Your Local Repository .................. 105
Viewing the Commit History ........................................... 108
Pulling and Pushing Files .............................................. 109
Resetting Your Local Repository ..................................... 109
Working with Branches in Git ......................................... 110
Creating a Branch ....................................................... 111
Merging Branches ....................................................... 112
Deleting a Repository ................................................... 112
Sample Git Workflow Scenario ....................................... 113

Chapter 8 • Understanding Tasks in SAS Studio ..................... 123
What Is a Task? .......................................................... 123
How to Run a Task ...................................................... 125
Save a Task and Its Option Settings ................................. 127
Edit a Predefined Task .................................................. 127
Create a New Task ...................................................... 128
Customizing the Task Code and the Task Layout in the Workspace ............ 129

Appendix 1 • Customizing SAS Studio ................................. 131
About Setting Your Preferences ....................................... 131
Setting General Preferences .......................................... 131
Setting the Start Up Preferences ...................................... 133
Setting the Code and Log Preferences ................................ 133
Setting the Results Preferences ....................................... 136
Setting Table Preferences .............................................. 137
Setting Task Preferences .............................................. 139
Accessing Repositories in SAS Studio ............................... 139
Setting Preferences for Background Job Submissions ............... 140
Setting Preferences for Git Profiles .................................. 141
Setting Preferences for Git Repositories ............................ 141

Appendix 2 • Converting SAS Enterprise Guide Projects to SAS Studio Process Flows .... 143
Opening SAS Enterprise Guide Projects in SAS Studio ............. 143
Your Server Environments Must Match ................................ 145
Open a SAS Enterprise Guide Project File ........................... 145
Understanding the Conversion Report ................................ 146
Supported Nodes ....................................................... 147
Unsupported Nodes .................................................... 152
SAS Enterprise Guide Project Items That Are Not Supported ........... 153
Prompts ................................................................. 153
Limitations ............................................................... 155

Appendix 3 • Text Encoding Options and Language Mappings ................................. 157
About the Text Encoding to Language Mappings .................... 157
Text Encoding Options and Language Mappings ..................... 157

Appendix 4 • Customized Output Environment .......................... 159
Overview ............................................................... 159
Generate Output for Other Output Destinations ...................... 160
Send Your Results to Another Location .............................. 160
Use a Custom Style for Your Output .................................. 161
Use an Image Format Other Than the Default ....................... 161
Create a Drill-down Graph ............................................. 161
Create an Animated GIF or SVG Image .............................. 162
Appendix 5 • Keyboard Shortcuts ................................................................. 163
  Keyboard Shortcuts in SAS Studio Workspace ...................................... 163
  Keyboard Shortcuts for the Code Editor ............................................ 165

Recommended Reading ................................................................. 171
Index .................................................................................................. 173
What’s New in SAS Studio 3.8

Overview

SAS Studio 3.8 includes these changes and enhancements:

- ability to append and clear the log. For more information, see “Append and Clear the Log” on page vii.
- access to the Git family of SAS DATA step functions and Git integration user interface. For more information, see “SAS Studio Git Integration” on page vii.
- support for the VALIDMEMNAME system option. For more information, see “Support for VALIDMEMNAME Option” on page viii.
- new preferences to control the number of rows and columns in the table viewer. For more information, see “New Preferences for Rows and Columns in the Table Viewer” on page viii.
- state of trees in the navigation pane or an open window persists after a refresh. For more information, see “Tree Location Persistence” on page viii.
- several new tasks including several new map tasks, a combine data task, econometric tasks, and multiple SAS Viya tasks. To use the new SAS Viya tasks, you must license and install SAS Viya 3.4.

Append and Clear the Log

You can now append new log information to the existing logs for programs and tasks. You can also automatically clear the log each time you submit code. For more information, see “Setting the Code and Log Preferences” on page 133.

SAS Studio Git Integration

Git is a version control system for tracking changes in files and coordinating work on those files among multiple people. You can now use the Git family of SAS DATA step functions in SAS Studio. You can also use the user interface that integrates some of the most common Git features into SAS Studio. You can stage changes and create commits,
create and switch branches, and resolve merge conflicts from within SAS Studio. For more information, see “About Git Integration in SAS Studio” on page 101.

Note: The SAS Studio documentation assumes that you are already familiar with Git.

Note: The Git user interface in SAS Studio became a production feature with the SAS Studio 3.8 hot fix. You can download the hot fix from SAS 9.4 Hot Fix Downloads.

Support for VALIDMEMNAME Option

You can now use the SAS member name policy preference to specify a set of rules for SAS data set names, SAS data view names, and item store names. For more information, see “Setting Table Preferences” on page 137.

New Preferences for Rows and Columns in the Table Viewer

You can now specify the number of rows to display per page of data and the maximum number of columns to display in the table viewer. For more information, see “Setting Table Preferences” on page 137.

Tree Location Persistence

The state of SAS Studio library, file, and folder trees persists after any activity that causes the tree to refresh in the navigation pane or an open window.

New Tasks

Data Tasks

The new Combine Tables task provides a variety of methods for combining two data tables. You can combine tables using DATA step code or PROC SQL. For more information, see “Combine Tables” in SAS Studio: Task Reference Guide.

Graph Tasks

In the Line Chart task, you can now include error bars in the chart when you calculate the mean of a variable. For more information, see “Line Chart” in SAS Studio: Task Reference Guide.
**Map Tasks**

Several new map tasks are available in this release:

- The Bubble Map task creates a map that is overlaid with a bubble plot. For more information, see “Bubble Map” in *SAS Studio: Task Reference Guide*.
- The Choropleth Map task creates a map of polygonal areas. For more information, see “Choropleth Map” in *SAS Studio: Task Reference Guide*.
- The Scatter Map task creates a map that is overlaid with a scatter plot. For more information, see “Scatter Map” in *SAS Studio: Task Reference Guide*.
- The Series Map task creates a map that is overlaid with a series plot. For more information, see “Series Map” in *SAS Studio: Task Reference Guide*.
- The Text Map task creates a map that is overlaid with a text scatter plot. For more information, see “Text Map” in *SAS Studio: Task Reference Guide*.

**Econometrics**

- The Severity Models task estimates parameters of any arbitrary continuous probability distribution that is used to model the magnitude (severity) of a continuous-valued event of interest. For more information, see “Severity Models” in *SAS Studio: Task Reference Guide*.
- The Spatial Regression Models task analyzes a class of linear spatial econometric models for cross-sectional data whose observations are spatially referenced or georeferenced. For more information, see “Spatial Regression Models” in *SAS Studio: Task Reference Guide*.
- The Time Series Analysis task has been divided into these two tasks:
  - The Univariate Time Series Analysis task analyzes a single time series where the values are equally spaced. This task provides analysis for these model types: ARIMA (autoregressive integrated moving average), ARIMAX, unobserved components, and regression with autocorrelated and heteroscedastic errors. For more information, see “Univariate Time Series Analysis” in *SAS Studio: Task Reference Guide*.
  - The Multivariate Time Series Analysis task analyzes a vector of time series that are equally spaced. For more information, see “Multivariate Time Series Analysis” in *SAS Studio: Task Reference Guide*.

**SAS Viya Evaluate and Implement**

The new Register task imports an analytic model from an analytic object in a CAS table into SAS Model Manager. Once in SAS Model Manager, you can score new data with the model, compare the model with other models, and publish the model to a production system. For more information, see “SAS Viya: Register” in *SAS Studio: Task Reference Guide*.

**SAS Viya Supervised Learning**

- The new Bayesian Network task trains a predictive model of a nominal target using different types of Bayesian network structures, including naive, tree-augmented

- In the Neural Network task, you can now specify up to 10 hidden layers. In the Code Generation options, you can use the NNET procedure or the CAS procedure to run the task. For more information, see “SAS Viya: Neural Network” in SAS Studio: Task Reference Guide.

- In the Decision Tree task, the default value for the Maximum depth of a tree option is 10. In the Code Generation options, you can now use the TREESPLIT procedure to run the task. For more information, see “SAS Viya: Decision Tree” in SAS Studio: Task Reference Guide.

**SAS Viya Text Analytics**

- The Segmentation task segments text data using k-means clustering. For more information, see “SAS Viya: Segmentation” in SAS Studio: Task Reference Guide.

- The Text Scoring task scores a data table using tables that are generated by another text analytics task. For more information, see “SAS Viya: Text Scoring” in SAS Studio: Task Reference Guide.

- In the Text Parsing and Discovery task, the Latent Dirichlet allocation (LDA) model is now available if you license and install SAS Visual Text Analytics. For more information, see “SAS Viya: Text Parsing and Topic Discovery” in SAS Studio: Task Reference Guide.

**SAS Viya Network Analysis and Optimization**

The new Summary Statistics task calculates various summary statistics for a graph and its nodes. For more information, see “SAS Viya: Summary” in SAS Studio: Task Reference Guide.

**SAS Viya Econometrics**


- The new Spatial Regression Models task analyzes a class of linear spatial econometric models for cross-sectional data with observations that are spatially referenced or georeferenced. For more information, see “SAS Viya: Spatial Regression Models” in SAS Studio: Task Reference Guide.

- The new Hidden Markov Models task analyzes the same time series or panel data by using a statistical Markov model to infer hidden states through a Markov process. For more information, see “SAS Viya: Hidden Markov Models” in SAS Studio: Task Reference Guide.

- The Severity Models task now includes an output option to create a table in an item store format. You can use this table in the new Aggregate Loss Models task. For more information, see “SAS Viya: Severity Models” in SAS Studio: Task Reference Guide.
In the Time Series Exploration task, creating output data tables is now optional. For more information, see “Time Series Exploration” in SAS Studio: Task Reference Guide.
Chapter 1
Introduction to SAS Studio

About SAS Studio

SAS Studio is a development application for SAS that you access through your web browser. With SAS Studio, you can access your data files, libraries, and existing programs, and you can write new programs. You can also use the predefined tasks in SAS Studio to generate SAS code. When you run a program or task, SAS Studio connects to a SAS server to process the SAS code. The SAS server can be a hosted server in a cloud environment, a server in your local environment, or a copy of SAS on your local machine. After the code is processed, the results are returned to SAS Studio in your browser.
SAS Studio supports multiple web browsers, such as Microsoft Internet Explorer, Apple Safari, Mozilla Firefox, and Google Chrome.

In addition to writing and running your own SAS programs, you can use the predefined tasks that are included with SAS Studio to analyze your data. The tasks are based on SAS procedures and provide access to some of the most commonly used graph and analytical procedures. You can also use the default task template to write your own tasks.

SAS Studio includes two different perspectives: the SAS Programmer perspective and the Visual Programmer perspective. A perspective is a predetermined set of features that is customized to meet the needs of a specific user type. By selecting a specific perspective, you can narrow the choices that are available in the interface and focus on the features that you need to use regularly. By default, when you open SAS Studio, the SAS Programmer perspective is selected. After you open SAS Studio, you can change the perspective by using the perspectives menu on the toolbar. The SAS Programmer and Visual Programmer perspectives are identical except that the Visual Programmer perspective includes access to process flows. For more information, see “Understanding Perspectives” on page 12.

Using SAS Studio

About Using SAS Studio

When you sign on to SAS Studio, the main SAS Studio window appears with a blank program window so that you can start programming immediately. You also have access to the navigation pane on the left.

Note: To sign out of SAS Studio, click Sign Out on the toolbar. Do not use the Back button on your web browser.
The main window of SAS Studio consists of a navigation pane on the left and a work area on the right. The navigation pane provides access to your server files and folder shortcuts, your tasks and snippets, the libraries that you have access to, and your file shortcuts. The **Server Files and Folders** section is displayed by default.

The work area is used to display your data, code, tasks, logs, and results. As you open these items, they are added to the work area as windows in a tabbed interface.

### Using the Navigation Pane

#### About Using the Navigation Pane
You can expand the sections of the navigation pane by clicking the section that you want to view.

#### Working with Server Files and Folders
The **Server Files and Folders** section of the navigation pane enables you to access files and folders from your SAS server.

*Note:* Access to FTP shortcuts in SAS Studio is disabled by default. Contact your system administrator if this feature is required.

The contents of the **Server Files and Folders** section depend on the type of SAS Studio deployment. The default folder shortcuts and root directories can be configured by your SAS administrator. For more information, see *SAS Studio: Administrator’s Guide*.

You can open files that are saved on the SAS server or the FTP server, such as SAS program files or program package files. You can also create a folder shortcut to access your z/OS files, and you can open SAS tables that are saved on the SAS server.

*Note:* You cannot open SAS tables from a remote FTP server.

You can use the **Server Files and Folders** section to create folders and folder shortcuts, download and upload files, and create a new SAS program. From the folders tree, you can expand and collapse folders, copy and move items, and open items in folders by double-clicking them or dragging them to the work area. In addition, you can open files from your folders and folder shortcuts by clicking 🔗 on the SAS Studio toolbar. You can
also view items in a folder as text by right-clicking them and selecting View File as Text. If the file you are viewing is larger than 3 MB, SAS Studio truncates the portion of the file that exceeds that limit.

*Note:* Files and folders that are located on an FTP server and are accessible by using an FTP shortcut cannot contain double-byte characters or any of the following characters in their names:

? / \ * " | : ; < >

SAS Studio cannot access, move, rename, or delete files and folders on an FTP server whose names contain invalid characters. In addition, you cannot rename or delete any global folder shortcuts.

To create a new folder shortcut:

1. Click Server Files and Folders in the navigation pane. Then click and select Folder Shortcut. The New Folder Shortcut window appears.
2. In the Name box, enter the name of the folder.
3. From the Folder Type drop-down list, specify whether the shortcut refers to a SAS server folder or an FTP folder.
4. If you are creating a shortcut to a SAS server folder, enter the physical path for the directory in the Directory box or click Browse to select a directory. To create a new folder when you are selecting a directory, click . If you are creating a shortcut to an FTP folder, enter the network address of the FTP host in the Host Name box as well as your user name and password. By default, the directory is the home directory of the FTP user. You can use the Directory box to specify another directory that is relative to the home directory. For example, if the home directory of the FTP user is c:\homedir, and you specify data in the Directory box, then the root directory of the shortcut is c:\homedir\data on the FTP server. You can validate your connection to the FTP server by clicking Test.

*Note:* If your mid-tier and FTP servers are running different operating systems, you must fully qualify the name of the FTP host in the Host Name box. If this name is not fully qualified, then the connection might fail.
5. Click **Save** to create the folder shortcut. The new shortcut is added to the list of folder shortcuts.

To create a new folder, select the folder in the **Server Files and Folders** section in which you want to create the new folder. Click ⬆️ and select **Folder**. The New Folder window appears. Enter the name of the new folder. The new folder is added to the list of folders.

To create a new XML file, Click ⬆️ and select **XML**. The new file opens in the XML editor in the work area.

To download a file, select the file that you want to download and click ⬇️. You are prompted to open the file in the default application or save it to your local computer.

To upload one or more files from your local computer, select the folder to which you want to upload the files and click ⬇️. The Upload Files window appears. Click **Choose Files** to browse for the files that you want to upload.

**Working with Tasks**

The **Tasks and Utilities** section of the navigation pane enables you to access tasks in SAS Studio. Tasks are based on SAS procedures and generate SAS code and formatted results for you. SAS Studio is shipped with several predefined tasks that you can run. You can also edit a copy of these predefined tasks, and you can create your own new tasks.

To create a new task, click ⬆️ and select **New Task**. SAS Studio creates a template in the work area that you can use to create custom tasks for your site. Custom tasks can be
accessed from the My Tasks folder or from the Server Files and Folders section of the navigation pane. For more information, see Chapter 8, “Understanding Tasks in SAS Studio,” on page 123.

To edit a task that you have created, select the task from the My Tasks folder and click \[\text{Edit}]. The XML code that is used to create the task is opened in the work area. If you want to edit a predefined task, you must first right-click the task and select Add to My Tasks or Add to Folders. For more information, see “Edit a Predefined Task” on page 127.

**Working with Snippets**

The Snippets section of the navigation pane enables you to access your code snippets. Code snippets are samples of commonly used SAS code that you can insert into your SAS program. SAS Studio is shipped with several predefined code snippets that you can use. You can also edit a copy of these snippets and create your own custom snippets. Your custom snippets can be accessed from the My Snippets folder. For more information, see Chapter 2, “Working with Programs,” on page 15.

To edit a snippet that you have created, select the snippet from the My Snippets folder and click \[\text{Edit}]. If you want to edit a predefined snippet, you must first right-click the snippet and select Add to My Snippets.

*Note:* You can edit only the snippets that are in the My Snippets folder.

**Working with Libraries**

The Libraries section of the navigation pane enables you to access your SAS libraries. SAS tables are stored in SAS libraries. From the Libraries section, you can open SAS tables and add them to your programs. You can use the Libraries section to expand a table and view the columns in that table. The icon in front of the column name indicates the type.

Here are examples of common icons for the column types.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Type of Column</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Character Icon" /></td>
<td>Character</td>
</tr>
<tr>
<td><img src="image" alt="Numeric Icon" /></td>
<td>Numeric</td>
</tr>
<tr>
<td><img src="image" alt="Date Icon" /></td>
<td>Date</td>
</tr>
<tr>
<td><img src="image" alt="Datetime Icon" /></td>
<td>Datetime</td>
</tr>
</tbody>
</table>

You can drag tables and columns from the Libraries section to a program, and SAS Studio adds code for the dragged items to your program. For more information, see “Opening and Creating Programs” on page 16.

*Note:* The Sasuser library is read only, as in any SAS server environment. You cannot save content to this library.

You can also create new libraries and assign existing libraries.
To create a new library:

1. Click **Libraries** in the navigation pane and then click 📚. The New Library window appears.

![New Library Window](image)

2. In the **Name** box, enter the libref for the library. The libref must be eight characters or fewer.

3. In the **Path** box, enter the physical path where the library resides or click **Browse** to select a location. To create a new folder when you are selecting a location, click 🚀.

4. In the **Options** box, specify any configuration options that you need. For the appropriate options, see the documentation for your operating environment.

5. If you want to access this library each time you use SAS Studio, select **Re-create this library at start-up**.

6. Click **OK** to create the library. The new library is added to the list of libraries in the navigation pane.

In the SAS Studio Mid-Tier (the enterprise edition) deployment, you can assign unassigned metadata libraries by clicking 🛡️. The libraries that you can assign must already be defined in your metadata. If you want to access the selected libraries each time you use SAS Studio, select **Assign selected libraries at start-up**. If a library is unassigned, then you cannot access the tables in that library.

**Using File Shortcuts**

File shortcuts enable you to quickly access files that you specify. You can create a file shortcut to a file on your SAS server, via a URL, or on your FTP server.

*Note:* You can create a file shortcut to a file on an FTP server only if you have created a folder shortcut to an FTP folder. FTP folders are not available by clicking **Browse** and using the Select File window, however. You must enter the pathname.

To create a new file shortcut, click 📄. You can define the shortcut by specifying a complete path and filename or by specifying a URL. If you want this shortcut to be
available the next time you use SAS Studio, select **Re-create this file shortcut at start-up**.

You can open a file from a file shortcut by double-clicking it or dragging it to the work area.

**Customizing the Navigation Pane**

By default, five sections of the navigation pane are displayed when you open SAS Studio in the SAS Programmer perspective. To customize which sections are displayed, click  and select **View**. Select or clear any sections that you want to add or remove. The navigation pane is updated immediately.

*Note:* The **File Shortcuts** section is not displayed by default in the Visual Programmer perspective.

**Using the Work Area**

**About Using the Work Area**

The work area is the main portion of the SAS Studio application for accessing programs and tasks and for viewing data. The work area is always displayed and cannot be minimized. When you open a program, task, or table, the windows open as new tabs in the work area. The code, log, and results that are associated with programs and tasks are grouped together under the main tab for the program or task.
Customizing the Work Area

By default, the work area is displayed beside the navigation pane, but you can maximize the work area and hide the navigation pane. You can also close all of the tabs in the work area at once.

To maximize the work area, click on the toolbar.

Note: To reopen the navigation pane, click again.

To close all tabs that are open in the work area, click and select Close All Tabs. You are prompted to save any unsaved programs or tasks.

Rearranging the Tabs in the Work Area

In the work area, you can rearrange the tabs by using a drag-and-drop operation to move them to the left or right. You can also dock a tab on the right side or bottom of the work area to view more than one tab at a time.

To rearrange a tab:

1. Select the tab that you want to move.
2. Move the tab icon to the location where you want to view this content. The icon indicates a valid location.
Note: The Results tab in Program 2 has also been moved to the right side of the program tab.

**Searching in SAS Studio**

You can use the Search feature to search all of the sections of the navigation pane. The types of items that you can search for depend on what is selected in the navigation pane. For example, if a folder or folder shortcut in the Server Files and Folders section is selected, you can search for folders and files, and you can choose whether to include subfolders in the search. If a task category in the Tasks and Utilities section is selected, you can search for task names, associated SAS procedures, and task descriptions.

To access the Search feature, click 🔍. The search box appears so that you can enter the text that you want to search for.

You can limit the scope of your search by selecting or clearing any of the Narrow by options. These options vary depending on what is selected in the navigation pane. By default, the search is case sensitive. To search for both uppercase and lowercase text, clear the Match case check box.

The following example shows a search for “class” in the Sashelp library. The search includes all tables and columns in the Sashelp library, and it is not case sensitive.

![Search Results](image)

You can open an item from your results by double-clicking it. To return to your search results list, click 🔍 again.

To clear the search results, click ✖️.

**Using the Messages Window**

The Messages window displays information about the programs, tasks, queries, and process flows that you run in SAS Studio. To open the Messages window, click
Messages in the lower right corner of your SAS Studio browser window. You can subset the messages that are displayed by clicking the Filter by drop-down list and selecting the type of message that you want to view.

If you have run a SAS program in batch mode, you can click View to open the Background Job Status window. For more information, see “Running a Program as a Background Job” on page 17. If you have lost your connection to a repository, a message is displayed along with a Test button that you can use to test the connection.

Note: If you have selected the Capture all log events option, then you can also choose to display debug notifications in the Messages window. For more information, see “Setting General Preferences” on page 131.

Understanding Perspectives

Because SAS Studio can be used by a variety of people and groups within an organization, you can choose to view a specific subset of features, or perspective, that meets your needs best. Perspectives are sets of functionality that are customized to meet the needs of different types of users. SAS Studio includes two perspectives: the SAS Programmer perspective and the Visual Programmer perspective.

The SAS Programmer perspective is for users who intend to use SAS Studio mainly for writing and editing SAS programs. By default, the SAS Programmer perspective opens with a new program window and includes four sections of the navigation pane so that you can easily manage all of your program files and code snippets: Server Files and Folders, Tasks and Utilities, Snippets, and Libraries.

The Visual Programmer perspective is designed for users who want to work with process flows in a project-based environment. You can use process flows in the Visual Programmer perspective to combine individual processes into one repeatable process flow that you can save, reuse, and share with other users. By default, the Visual Programmer perspective opens with a new process flow window and includes the same four sections in the navigation pane.
Note: If you open a process flow while you are using the SAS Programmer perspective, you are prompted to switch to the Visual Programmer perspective.

In both perspectives, you can run the predefined tasks that are shipped with SAS Studio, and you can create and edit tasks. You can also use both perspectives to create and run SAS programs and queries.

The differences between the perspectives can be viewed in the following table:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation Pane sections</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Server Files and Folders</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>• Tasks and Utilities</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>• Snippets</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>• Libraries</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>• File Shortcuts</td>
<td>Yes, but not displayed by default</td>
<td>Yes, but not displayed by default</td>
</tr>
<tr>
<td>• Git Repository (experimental)</td>
<td>Yes, but not displayed by default</td>
<td>Yes, but not displayed by default</td>
</tr>
<tr>
<td>SAS Programs</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Queries</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Process Flows</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

After you have started SAS Studio and selected a perspective, you can change the perspective that you are using by clicking the perspectives menu on the toolbar and selecting the perspective that you want to use.

You can specify which sections of the navigation pane are displayed in SAS Studio by clicking and selecting View.

**Editing the Autoexec File**

The autoexec.sas file includes SAS statements that run each time you start SAS Studio and connect to your SAS server. For example, you can use the autoexec.sas file to assign libraries that you want to be available every time you use SAS Studio in both interactive and noninteractive modes. The autoexec file is saved in your home directory.

Note: If you create a new library by using the New Library window, you can select the Re-create this library at start-up option to automatically add the LIBNAME statement to the autoexec.sas file. For more information, see “Working with Libraries” on page 6.
To edit the autoexec.sas file:
1. Click and select Edit Autoexec File.
2. Enter the code that you want to include in the autoexec.sas file.
3. To validate your syntax, click Run. The Log tab opens so that you can view the log.
4. Click Save to save and close the autoexec file.

Changing Your SAS Workspace Server

If you have access to more than one SAS workspace server, you can change the server that SAS Studio connects to. To change the server, click and select Change SAS Workspace Server. Select the server that you want to use. When you change servers, any libraries and file shortcuts that you created are deleted. For more information, see SAS Studio: Administrator’s Guide.
Chapter 2
Working with Programs

About the Code Editor

SAS Studio includes a color-coded, syntax-checking editor for editing new or existing SAS programs. You can also edit SOURCE entries in SAS catalogs. The editor includes a wide variety of features such as autocompletion, automatic formatting, and pop-up syntax help. With the code editor, you can write, run, and save SAS programs. You can also modify and save the code that is automatically generated when you run a task.

SAS Studio also includes several sample code snippets that you can use to make programming common tasks easier.

In the code editor, you can use the programming language for SAS 9.4 or SAS Viya. For more information, see these resources:
Opening and Creating Programs

Opening a Program

You can open SAS programs from the Server Files and Folders section of the navigation pane. To open a program, expand the appropriate folder and double-click the program that you want to open, or drag it into the work area. The program opens on a new tab in the work area.

Note: Opening very large program files can affect your performance. If you open a program file that is greater than 10 MB, you are prompted to confirm whether you want to continue opening the file. If you are using Internet Explorer, you are prompted for confirmation if you open a program file that is greater than 3 MB.

Creating a Program

You can create a SAS program from the Server Files and Folders section of the navigation pane. To create a program, click and select SAS Program. A program window appears on a new tab in the work area.

Note: You can also click on the main application toolbar and select New SAS Program.

After you save your program, you can use the Enable autosave option to automatically create a copy of your new program as well as each previously saved program file that you are editing. You can use the auto-saved copies to recover the files if you lose your network connection or your browser closes unexpectedly. For more information, see “Setting the Code and Log Preferences” on page 133.

Note: The Enable autosave option is available only for programs that you have saved at least once. This option does not apply to programs that are opened from a folder shortcut to an FTP folder.

Running a Program

After you have written your program, you can run the entire program or you can select specific lines of code to run. To run the entire program, click . To run a portion of the program, select the lines of code that you want to run and then click .

If there are no errors, the results open automatically. If there are errors, the Log tab opens by default. You can expand the Errors, Warnings, and Notes sections to view the messages. When you click a message, SAS Studio highlights it for you in the log so that you can see exactly where the message occurs in the log.
Using the Background Submit Feature

About the Background Submit Feature
You can run a saved SAS program as a background job, which means that the program can run while you continue to use SAS Studio. You can view the status of programs that have been submitted as background jobs, and you can cancel programs that are currently running in the background.

Running a Program as a Background Job
To run a program as a background job, right-click the program in the navigation pane and select Background Submit. Before the program is run, the background process changes the current working directory to the directory in which the program is located.

Note: You cannot run a program as a background job if the program is saved on an FTP server or on a SAS server running on the native z/OS file system. Programs that are saved on a z/OS SAS server running the HFS file system can be run as background jobs.

By default, a notification message is displayed when the program is submitted and again when the program has finished running. If you log off from SAS Studio while the program is running, the program continues to run, but the notification message that indicates when the program is finished is not displayed.
Note: Because a background job uses a separate workspace server, any libraries that are created by your program do not appear in the Libraries section of the navigation pane in SAS Studio.

To view the status of your background jobs, click and select Background Job Status.

You can use the Background Job Status window to perform these tasks:

- Displays the background job properties, including file pathnames, start time, end time, and status.
- Opens the program log in the SAS Studio work area.
- Opens the program results in a separate window or download the results file.
- Cancels a running background job. Any output that has been created by the job is not deleted.

After the background job completes, you must refresh the Server Files and Folders pane to view any output and log files.

Customizing Your Background Job Submissions

The Preferences window enables you to customize how to handle background job submissions.

To change whether existing log and output files are deleted or overwritten when you rerun a background job, click and select Preferences. Click Background Jobs.
For more information about each option, see “Setting Preferences for Background Job Submissions” on page 140.

Using the Autocomplete Feature

About the Autocomplete Feature
The autocomplete, or code completion, feature in the code editor can predict the next word that you want to enter before you actually enter it completely. The autocomplete feature can complete keywords that are associated with SAS procedures, statements, macros, functions, CALL routines, formats, informats, macro variables, SAS colors, style elements, style attributes, and statistics keywords, and various SAS statement and procedure options. The autocomplete feature can also complete librefs and table names.

Note: The autocomplete feature is available only for editing SAS programs.

TIP If you want to see autocomplete suggestions only when you need them, you can disable the feature by clicking and selecting Preferences. Click Code and Log, and then clear the Enable autocomplete check box. When you want an autocomplete suggestion, position your cursor in the appropriate text and press Ctrl + Spacebar.

This example shows the keywords and help that appear when you enter proc a in the code editor.
In this example, you select **APPEND** from the list of procedures so that `proc append` appears in the code editor. When you enter a space, the code editor displays a list of options for the APPEND procedure.

---

**How to Use the Autocomplete Feature**

To use the autocomplete feature:

1. How you open the autocomplete list depends on the keyword that you want to add.
   - If you want to add a global statement, DATA step statement, CALL routine, procedure, macro statement, or automatic macro variable, enter the first one or more letters of the keyword that you want to use.

   A window appears with a list of suggested keywords that begin with those letters.

   - If you want to specify colors, formats, informats, macro functions, SAS functions, statistics keywords, style elements, or style attributes, position your mouse pointer in a comment and press Ctrl+spacebar. To navigate through the list of options backward, press Ctrl+Shift+spacebar or use the arrow keys at the top of the pop-up window.
Note: These shortcuts work even if you have deselected the Enable autocomplete option in the Preferences window. For more information, see “Customizing the Code Editor” on page 40.

2. You can navigate to the keyword that you want to use in several ways:
   • Continue to type until the correct keyword is selected (because the matching improves as you type).
   • Scroll through the list by using the up and down arrow keys, the Page Up and Page Down keys, or your mouse.

3. You can add the keyword to your program by double-clicking the selected keyword or by pressing the Enter key.

**Using the Syntax Help**

The code editor displays brief SAS syntax documentation as you write and edit your programs. You can display the Help in these ways:

• Right-click a keyword in your program and select Syntax Help.

• Start entering a valid SAS keyword, and then click a suggested keyword in the autocomplete window.

• Position the mouse pointer over a valid SAS keyword in your program. This works only if you have selected the Enable hint option in the Editor preferences. For more information, see “Customizing the Code Editor” on page 40.

The SAS Product Documentation provides more comprehensive usage information about the SAS language, but the syntax help in the code editor can get you started with a hint about the syntax or a brief description of the keyword. You can get additional help by clicking links in the syntax help window as follows:

• Click the keyword link at the top of the window to search the support.sas.com website for the keyword.
• Click the links at the bottom of the window to search for the keyword in the SAS Product Documentation, Samples and SAS Notes, and SAS Technical Papers.

Matching Parentheses

You can use the parenthesis matching feature to track nested parentheses within a program. The code editor highlights both the open and close parentheses. If only one parenthesis is highlighted, then you know that you are missing a parenthesis. This feature can be used to match parentheses, square brackets, and braces.

```
   do y = -5 to 5 by .5;
      z = sin(sqrt(y^2 + x^2));
```
To match parentheses, position the cursor in front of the open parenthesis or just after the close parenthesis that you want to match. The parenthesis and its match are highlighted. If the parenthesis does not have a match, then it is not highlighted.

**Selecting Columns of Text to Edit**

You do not have to select entire horizontal lines of text. You can select columns or vertical blocks of text.

To select a column or vertical block of text:

- In Window environments, press the Alt key while you select the text with the left mouse button.
- In Mac OS X environments, press the Option key while you select the text with the left mouse button.

---

**Adding Table Names and Column Names**

From the Libraries section of the navigation pane, you can use a drag-and-drop operation to move table names and column names into the SAS code. For example, you can move the Sashelp.Cars table into the DATA option for the PRINT procedure. When you release the mouse, the fully qualified name for the table appears in your code.
Editing the Code from a Task

You can edit the code that is generated automatically when you run a task and then run it with your modifications. When you edit the code, SAS Studio opens it in a separate program window. The code is no longer associated with the original task.

To edit a program generated by a task:

1. On the appropriate task tab in the work area, click Code to display the code that is associated with the task.

   Note: In order to edit the code that is associated with a task, you must first display the code with the task. If the task code is not displayed, click Preferences and select Show Task Code.

2. On the toolbar, click Edit. The code is opened in a new program window.
Creating a SAS Program Package

A SAS Program Package is a file that contains a snapshot of a SAS program along with its log and HTML results. You can create a program package from code that you have written as well as code that is automatically generated when you run a task. When you open a program package in SAS Studio, you can access the code, log, and results without running the program again. If you make changes to the code and rerun it, the package is not automatically updated. You must save the package again to keep the changes.

Note:

• The program package does not include PDF or RTF results.
• You cannot create a program package if you are running your program in interactive mode.

To create a program package file, open the code that you want to use and click . Specify the file location and name, and then select SAS Program Package as the Save as type option.

Note: If you open a program package file and want to save the program or log individually or download the results as an HTML, PDF, or RTF file, you must resubmit the program after you open the program package file.
Creating a Program Summary

You can create a summary page for code that you have written as well as code that is automatically generated when you run a task. The Program Summary page is an HTML file that opens in a separate browser tab and includes information about the program execution, the complete SAS source code, the complete SAS log, and the results. To view the Program Summary page for a program, click  .

Note: The Program Summary is available only after you have run the program.

You can also save a Program Summary file to a folder that you specify by clicking and selecting **SAS Program Summary** as the **Save as type** option. The Program Summary is saved as an HTML file.

Using Macro Variables

Macro variables can be used to add information that is obtained when a program or task is run, such as the name and version number of the application. You can reference these items within code, titles, or footnotes by preceding them with "&". To resolve a macro variable reference that occurs within a literal string, enclose the string in double quotation marks.

For example, you can use macro variables in a FOOTNOTE statement:

```
footnote "Generated with &_CLIENTAPP &_CLIENTVERSION";
```

Note: In addition to the following macro variables, you can run this code to see other user-generated and automatic macro variables that are available:

```
%PUT _ALL_;
```

For information about SAS macro functions and variables, see *SAS Macro Language: Reference*.

<table>
<thead>
<tr>
<th>Macro Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_CLIENTAPP</td>
<td>Name of the client application.</td>
</tr>
<tr>
<td>_CLIENTAPPABREV</td>
<td>Abbreviated name of the client application.</td>
</tr>
<tr>
<td>_CLIENTMACHINE</td>
<td>Node name of the client machine.</td>
</tr>
<tr>
<td>_CLIENTMODE</td>
<td>Type of SAS Studio deployment: Mid-Tier (enterprise), Single-User, or Basic.</td>
</tr>
<tr>
<td>_CLIENTUSERID</td>
<td>User ID of the client user.</td>
</tr>
<tr>
<td>_CLIENTUSERNAME</td>
<td>Full user name, if that information is available.</td>
</tr>
<tr>
<td>_CLIENTVERSION</td>
<td>Application version, including build number.</td>
</tr>
<tr>
<td>_SASHOSTNAME</td>
<td>Server node name (IP address or DNS name).</td>
</tr>
</tbody>
</table>
### Macro Variable

<table>
<thead>
<tr>
<th>Macro Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_SASPROGRAMFILE</td>
<td>The full path and file name of the SAS program that is currently being run. This macro variable is available only for SAS program files that are saved on the same server on which your SAS Studio code is running.</td>
</tr>
<tr>
<td>_SASPROGRAMFILEHOST</td>
<td>The server node name on which the current SAS program is being run.</td>
</tr>
<tr>
<td>_SASWORKINGDIR</td>
<td>Current working directory.</td>
</tr>
<tr>
<td>SASWORKLOCATION</td>
<td>Location of the Work library.</td>
</tr>
<tr>
<td>SYSPROCESSNAME</td>
<td>Name of the current SAS process.</td>
</tr>
<tr>
<td>SYSPROCESSMODE</td>
<td>Current SAS session run mode or server type name.</td>
</tr>
<tr>
<td>SYSVLONG4</td>
<td>SAS software release number, maintenance level, and four-digit year.</td>
</tr>
</tbody>
</table>

**Note:** If you specify `%put _all_` or `%put _global_` in your SAS program, the output does not include any special characters. For example, slashes are not included in directory paths. To view the output with these special characters, you must specify the individual macro variable by name, such as `%put &_sasprogramfile;`.

### Using Your Submission History

SAS Studio maintains a log with entries for each time you run a program or task. You can use this log, or submission history, to access prior versions of your submitted code. To view your submission history, click the **Code** tab in your program or task window.

On the toolbar, click ![code](image) and select the version that you want to open. The prior version of the program opens in a new window from which you can copy and paste the code as needed.

**Note:** The submission history is cleared when you sign off from SAS Studio.

### Automatically Formatting Your SAS Code

You can use the code editor to make your programs easier to read by automatically formatting your code. When you automatically format your code, line breaks are added, and each line is correctly indented according to its nesting level. To format the code in the code editor, click ![format](image).

For example, the following code is difficult to read because it lacks indentation and logical line breaks:

```sas
data topn;
length rank 8; label rank="Rank";
set topn; by &category descending &measure;
```
After you use the automatic code-formatting feature, the program looks like this:

data topn;
  length rank 8;
  label rank="Rank";
  set topn;
  by &category descending &measure;
  if first.&category then
    rank=0;
    rank+1;
  if rank le &n then
    output;
run;

Working in Interactive Mode

What Is Interactive Mode?
Some SAS procedures are interactive, which means they remain active until you submit a QUIT statement, or until you submit a new PROC or DATA step. In SAS Studio, you can use the code editor to run these procedures, as well as other SAS procedures, in interactive mode.

By using interactive mode, you can run selected lines of code from your SAS program and use the results to determine your next steps. For example, the OPTMODEL procedure in SAS/OR enables you to model and solve mathematical programming models. By running this procedure interactively, you can quickly check results for parts of the program and determine whether you need to make any modifications without running the entire program.

Note: Interactive mode is available only if you are running SAS 9.4M1 or later releases.

Running a Program in Interactive Mode
To run a program in interactive mode, click on the toolbar. To turn off interactive mode, click again. If you change modes while a program is open, the log and results for that program are cleared. You can also clear the log and results manually by clicking on the appropriate toolbar.

When you run a program in interactive mode, SAS Studio does not add any automatically generated code, such as ODS and %LET statements, to your program. In addition, results are generated only in HTML. In interactive mode, the log and results are appended to the existing log and results. Previously submitted code remains active until you terminate it.

For example, suppose you have the following program:

proc sql;
  select * from sashelp.cars;
  select * from sashelp.class;

quit;

In noninteractive mode, if you select the first two lines of code and submit them, the code runs successfully. If you then select the last two lines of code and submit them, the code fails because the PROC SQL statement is missing.

If you switch to interactive mode and follow the same steps, the last two lines of code run successfully because the PROC SQL statement is still active.

Note: For documentation about specific procedures, see the SAS Programmer’s Bookshelf on support.sas.com.

About Libraries in Interactive Mode

When you use interactive mode, you are creating a new workspace server session with its own list of libraries that are available only in that session. If you assign a new library when you are in interactive mode, the library might not be available when you turn off interactive mode. In order to share data in a library between interactive mode and noninteractive mode, the library must be included in your autoexec.sas file so that it is assigned every time you use SAS Studio. For more information, see “Editing the Autoexec File” on page 13.

Webwork is the default output library in interactive mode. If you refer to a table without specifying both the libref and the table name, SAS Studio assumes it is stored in the Webwork library. The Webwork library is shared between interactive mode and noninteractive mode. Any data that you create in the Webwork library in one mode can be accessed in the other mode.

Note: If you run a program that creates output data in interactive mode and this data is saved in a library that is available in both interactive and noninteractive mode (such as the Webwork library), you must refresh the Libraries section of the navigation pane to view the new data from the user interface.

The Work library is not shared between interactive mode and noninteractive mode. Each workspace server session has its own separate Work library, and data cannot be shared between them. Any data that you save to the Work library in interactive mode cannot be accessed from the Work library in noninteractive mode. In addition, you cannot view data in the Work library from the Libraries section of the navigation pane if the data was created in interactive mode. However, you can access that data programmatically in interactive mode.

Special Considerations When Using Interactive Mode

The following list summarizes some of the special considerations for using interactive mode:

• If you run a program that creates output data in interactive mode, you must refresh the Libraries section of the navigation pane to view the new data.

• You cannot view any data that you create in the Work library in interactive mode in the Libraries section of the navigation pane. However, you can access that data programmatically.

• Because each program that is opened in interactive mode creates a new workspace server session, any options, macros and macro variables, and other data that is specific to your workspace server session cannot be shared with other programs in either interactive or noninteractive mode. Programs that are opened in noninteractive mode use the same workspace server session and can therefore share options, macros and macro variables, and other data.

• When you run a program in interactive mode, results are generated only in HTML.
• You can turn interactive mode on and off for an individual program tab by clicking 

When interactive mode is turned on, the button appears to be highlighted on the toolbar. If you want to use interactive mode for all program tabs, click 

Preferences. Click General, and then select Start new programs in interactive mode. If you do not select this option, you must turn interactive mode on each time you want to use it.

---

**Working with Code Snippets**

**Why Use Code Snippets?**

Code snippets enable you to quickly insert SAS code into your program and customize it to meet your needs. SAS Studio is shipped with several code snippets. You can also create your own snippets and add snippets to your list of favorites.

<table>
<thead>
<tr>
<th>Snippet Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalogs</td>
<td>Enables you to specify the content for the SOURCE entry type in a catalog. The SOURCE entry can contain the same code as SAS programs.</td>
</tr>
<tr>
<td>Edit a SOURCE Entry</td>
<td>Lists the contents of a specified type in a specified catalog. By default, the snippet shows how to list all types as well as only the CLASS type in the catalog Sashelp.FSP.</td>
</tr>
<tr>
<td>List Catalog Entries</td>
<td>Creates and displays a SAS table that lists all of the categories in a specified library. By default, the table is saved as Work.catalogs.</td>
</tr>
<tr>
<td>Print GRSEG Entry</td>
<td>Prints the GRSEG entry to the open destinations specified by the SAS Output Delivery System (ODS output).</td>
</tr>
</tbody>
</table>

**Data**

<table>
<thead>
<tr>
<th>Snippet Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS2 Code</td>
<td>Provides a template for a DS2 program. DS2 is a SAS programming language that is appropriate for advanced data manipulation. DS2 is included with Base SAS and shares core features with the SAS DATA step. DS2 exceeds the DATA step by adding variable scoping, user-defined methods, ANSI SQL data types, and user-defined packages. The DS2 SET statement accepts embedded FedSQL syntax, and the run-time-generated queries can exchange data interactively between DS2 and any supported database. This data exchange allows SQL preprocessing of input tables, which effectively combines the power of the two languages. For more information, see <em>SAS DS2 Language Reference</em>.</td>
</tr>
<tr>
<td>Snippet Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DS2 Package</td>
<td>Provides a template for a DS2 package. A package is similar to a DS2 program. The package body consists of a set of global declarations and a list of methods. The main syntactical differences are the PACKAGE and ENDPACKAGE statements. These statements define a block with global scope. For more information, see <em>SAS DS2 Language Reference</em>.</td>
</tr>
<tr>
<td>DS2 Thread</td>
<td>Provides a template for a DS2 threaded program. Typically, DS2 code runs sequentially. That is, one process runs to completion before the next process begins. It is possible to run more than one process concurrently, using threaded processing. In threaded processing, each concurrently executing section of code is said to be running in a thread. For more information, see <em>SAS DS2 Language Reference</em>.</td>
</tr>
<tr>
<td>Generate CSV File</td>
<td>Exports SAS data as a comma-separated text file.</td>
</tr>
<tr>
<td>Generate PowerPoint Slide</td>
<td>Streams Microsoft PowerPoint output to your web browser.</td>
</tr>
<tr>
<td>Generate XML File</td>
<td>Exports SAS data as an XML file that you can view in your web browser.</td>
</tr>
<tr>
<td>Import CSV File</td>
<td>Imports a comma-separated file and writes the output to a SAS data set.</td>
</tr>
<tr>
<td>Import XLSX File</td>
<td>Imports a Microsoft XLSX file and writes the output to a SAS data set.</td>
</tr>
<tr>
<td>Simulate Linear Regression Data</td>
<td>Creates an input data source that you can use for linear regression analysis. Linear regression analysis tries to assign a linear function to your data by using the least squares method.</td>
</tr>
<tr>
<td>Simulate One-Way ANOVA Data</td>
<td>Creates an input data source that considers one treatment factor with three treatment levels. When you analyze this data by using the One-Way ANOVA task, the goal is to test for differences among the means of the levels and to quantify these differences.</td>
</tr>
<tr>
<td>Descriptive</td>
<td></td>
</tr>
<tr>
<td>Custom ODS Output</td>
<td>Provides a template for creating HTML, PDF, and RTF output by using the SAS Output Delivery System. For more information, see <em>SAS Output Delivery System: User’s Guide</em>.</td>
</tr>
<tr>
<td>PROC SQL</td>
<td>Provides a template for writing SQL queries. For more information, see <em>SAS SQL Procedure User’s Guide</em>.</td>
</tr>
</tbody>
</table>

**Graph**

*Note:* For more information about the SGPLOT, SGPANEL, and SGSCATTER procedures, see *SAS ODS Graphics: Procedures Guide*. 
<table>
<thead>
<tr>
<th>Snippet Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bar Panel</td>
<td>Uses the VBAR statement in the SGPANEL procedure and enables you to create multiple bar charts.</td>
</tr>
<tr>
<td>Box Panel</td>
<td>Uses the VBOX statement in the SGPANEL procedure and enables you to create multiple box plots.</td>
</tr>
<tr>
<td>Comparative Scatter Plot</td>
<td>Uses the COMPARE statement in the SGSCATTER procedure. This code snippet creates a comparative panel of scatter plots with shared axes.</td>
</tr>
<tr>
<td>Dot Plot</td>
<td>Uses the DOT statement in the SGPLOT procedure. Dot plots summarize horizontally the values of a category variable. By default, each dot represents the frequency for each value of the category variable.</td>
</tr>
<tr>
<td>Fit Plot</td>
<td>Uses the REG statement in the SGPLOT procedure. This code snippet produces a regression plot with a quadratic fit and includes confidence limits.</td>
</tr>
<tr>
<td>HBar Plot</td>
<td>Uses the HBAR statement in the SGPLOT procedure. This code snippet creates a horizontal bar chart that summarizes the values of a category variable.</td>
</tr>
<tr>
<td>HighLow Plot</td>
<td>Uses the HIGHLOW statement in the SGPLOT procedure. High-low charts show how several values of one variable relate to one value of another variable. Typically, each variable value on the horizontal axis has several corresponding values on the vertical axis.</td>
</tr>
<tr>
<td>Histogram Plot</td>
<td>Uses the HISTOGRAM statement in the SGPLOT procedure. This code snippet produces a histogram with two density plots. In this snippet, one density plot uses a normal density estimate and the other density plot uses a kernel density estimate.</td>
</tr>
<tr>
<td>Scatter Plot Matrix</td>
<td>Uses the MATRIX statement in the SGSCATTER procedure. This code snippet creates a scatter plot matrix.</td>
</tr>
<tr>
<td>VBox Plot</td>
<td>Uses the VBOX statement in the SGPLOT procedure. A box plot summarizes the data and indicates the median, upper and lower quartiles, and minimum and maximum values. The plot provides a quick visual summary that easily shows center, spread, range, and any outliers. The SGPLOT and the SGPANEL procedures have separate statements for creating horizontal and vertical box plots.</td>
</tr>
</tbody>
</table>

**IML**

*Note:* These snippets are available only if your site licenses SAS/IML. For more information about SAS/IML, see *SAS/IML User’s Guide*.

<table>
<thead>
<tr>
<th>Snippet Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find Roots of Nonlinear Equation</td>
<td>Enables you to find the roots of a function of one variable. Finding the root (or zero) of a function enables you to solve nonlinear equations.</td>
</tr>
<tr>
<td>Snippet Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Fit by Using Maximum Likelihood</td>
<td>Uses maximum likelihood estimation to estimate parameters for the normal density estimate.</td>
</tr>
<tr>
<td>Generate a Bootstrap Distribution</td>
<td>Uses the IML procedure to create and analyze a bootstrap distribution of the sample mean.</td>
</tr>
<tr>
<td>Integrate a Function</td>
<td>Enables you to numerically integrate a one-dimensional function by using the QUAD subroutine in SAS/IML software. Use the QUAD subroutine to numerically find the definite integral of a function on a finite, semi-infinite, or infinite domain.</td>
</tr>
<tr>
<td>Simulate Multivariate Normal Data</td>
<td>Simulates data from a multivariate normal distribution with a specified mean and covariance.</td>
</tr>
<tr>
<td>Macro</td>
<td>For more information about SAS macros, see <em>SAS Macro Language: Reference</em>.</td>
</tr>
<tr>
<td>SAS Macro</td>
<td>Provides a basic template for working with SAS macros. Macros enable you to perform many tasks, including substituting text in a program. A SAS program can contain any number of macros, and you can invoke a macro multiple times in a single program.</td>
</tr>
<tr>
<td>SAS Macro Variables</td>
<td>Provides examples of how to create user-defined global and local macro variables. Macro variables are tools that enable you to dynamically modify the text in a SAS program through symbolic substitution. You can assign large or small amounts of text to macro variables. Then you can use that text by simply referencing the variable that contains the text. Macro variables that are defined by the macro programmer are called user-defined macro variables. Macro variables that are defined by the macro processor are called automatic macro variables. You can define and use macro variables anywhere in SAS programs, except within data lines.</td>
</tr>
<tr>
<td>SAS Macro Do Statement</td>
<td>Designates the beginning of a section of a macro definition that is treated as a unit until a matching %END statement is encountered. This macro section is called a %DO group. A simple %DO statement often appears in conjunction with %IF-%THEN-%ELSE statements to designate a section of the macro to be processed depending on whether the %IF condition is true or false. Note: SAS also provides a %DO iterative statement, which is different from the code that is generated by this snippet. For more information, see <em>SAS Macro Language: Reference</em>.</td>
</tr>
<tr>
<td>SAS Macro If Statement</td>
<td>Conditionally processes a portion of a macro. The expression that is the condition for the %IF-%THEN-%ELSE statement can contain only operands that are constant text or text expressions that generate text.</td>
</tr>
<tr>
<td>Snippet Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SAS Macro Parameters</td>
<td>Names one or more local macro variables whose values you specify when you invoke the macro. There are two types of macro variables: positional and keyword. Parameters are local to the macro that defines them. You must supply each parameter name. You cannot use a text expression to generate it. A parameter list can contain any number of macro parameters separated by commas. The macro variables in the parameter list are usually referenced in the macro.</td>
</tr>
</tbody>
</table>
SAS Macro Quoting

Provides examples of macro functions that tell the macro processor to interpret special characters and mnemonics as text rather than as part of the macro language.

- The %STR function masks special characters and mnemonic operators in constant text at macro compilation. This function masks these special characters and mnemonic operators: + - * / < > = ¬ ^ ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN

This function also masks these characters when they occur in pairs and when they are not matched and are marked by a preceding %: ' " ()

- The %NRSTR function masks special characters and mnemonic operators in constant text at macro compilation. This function masks all of the special characters and mnemonic operators listed for the %STR function. In addition, the %NRSTR function masks these characters: & %

- The %BQUOTE function masks special characters and mnemonic operators in a resolved value at macro execution. This function masks these special characters and mnemonic operators: ' " () + - * / < > = ¬ ^ ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN

- The %SUPERQ function masks all special characters and mnemonic operators at macro execution but prevents further resolution of the value. This function masks these special characters and mnemonic operators: & % ' " () + - * / < > = ¬ ^ ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN

- The %QSCAN function searches for a word and masks special characters and mnemonic operators.

- The %QSUBSTR function produces a substring and masks special characters and mnemonic operators.

- The %QUPCASE function converts a value to uppercase and returns a result that masks special characters and mnemonic operators.

- The %UNQUOTE function unmasks a value during macro execution so that any special characters and mnemonic operators are interpreted as macro language elements instead of text.

For more information about macro compilation and macro execution, see SAS Macro Language: Reference.
**SAS Macro Char Functions**

Provides several examples of these SAS macros that work with character values:

- The `%EVAL` function evaluates arithmetic and logical expressions by using integer arithmetic. This function operates by converting its argument from a character value to a numeric or logical expression. After the expression is evaluated, the result is converted back to a character value.

  This function is useful because the SAS Macro Facility is basically a text generator. As a result, an arithmetic expression is first converted to a numeric expression. After this numeric expression is evaluated, it is converted back to an arithmetic expression.

- The `%INDEX` function returns the position of the first character of a string.

- The `%LENGTH` function returns the length of a string.

- The `%SCAN` function searches for a word that is specified by its position in a string.

- The `%SUBSTR` function produces a substring of a character string.

- The `%UPCASE` function converts values to uppercase.

---

**SAS Viya Cloud Analytic Services**

Create CAS Connection

Creates a connection to a CAS server. You must specify values for the CASHOST= and CASPORT= system options. The CAS statement connects the default session to the specified CAS server and CAS port.

**Note:** To connect to the CAS Server, a valid certificate must be in place. For information about obtaining and configuring certificates, see Configure SAS 9.4 Clients to Work with Viya in Encryption in *Encryption in SAS Viya: Data at Rest.*

New CAS Session

Starts a new CAS session named mySession using the existing CAS server connection.

When starting a new session, you can specify the CAS library to use, the time-out (in seconds) for the session, and the locale of the session.

Disconnect CAS Session

Disconnects from the CAS session named mySession. Before you disconnect, you can specify a value for the time-out (in seconds). You can reconnect to the session before the time-out expires. Otherwise, the session is terminated.

Reconnect CAS Session

Reconnects to a CAS session named mySession.

Terminate CAS Session

Terminates the CAS session named mySession. No reconnection is possible.

List CAS Session Options

Lists session options for the specified CAS session.
<table>
<thead>
<tr>
<th>Snippet Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List CAS Sessions for SAS Client</td>
<td>Lists all the CAS sessions and session properties that are created by the SAS client or reconnected to by the SAS client.</td>
</tr>
<tr>
<td>List CAS Sessions for User ID</td>
<td>Lists all the CAS sessions that are known to the CAS server for the user ID that is associated with yourSessionName.</td>
</tr>
<tr>
<td>New caslib for Path</td>
<td>Creates a CAS library (myCaslib) for the specified path (/filePath/) and session (mySession). If you omit the SESSREF= option, the caslib is created and activated for the current session. The SUBDIRS option extends the scope of myCaslib to include subdirectories of /filePath/. The LIBNAME statement creates a SAS libref for the CAS library.</td>
</tr>
<tr>
<td>Generate SAS librefs for caslibs</td>
<td>Creates a default CAS session and generates SAS librefs for existing CAS libraries so that the librefs are visible in the Libraries pane.</td>
</tr>
<tr>
<td>Save Table to caslib</td>
<td>Creates a permanent copy of an in-memory table (sourceTableName) from a CAS library (sourceCaslib). The in-memory table is saved as targetTableName in the target caslib targetCaslib.</td>
</tr>
<tr>
<td></td>
<td>Note: You can determine the caslib that is associated with a CAS engine libref by right-clicking the libref in the Libraries pane and selecting Properties. The Server Session CASLIB field displays the caslib.</td>
</tr>
<tr>
<td>Load Data to caslib</td>
<td>Loads a table to the specified CAS library.</td>
</tr>
<tr>
<td>Delete Table or File from caslib</td>
<td>Deletes a table or file from the CAS library. You can also remove an in-memory table.</td>
</tr>
<tr>
<td>Delete caslib</td>
<td>Deletes the specified CAS library.</td>
</tr>
<tr>
<td>SAS Viya Machine Learning</td>
<td>You must license and install SAS Visual Data Mining and Machine Learning to use these snippets.</td>
</tr>
<tr>
<td></td>
<td>For more information, see SAS Visual Statistics: Procedures.</td>
</tr>
<tr>
<td>Load Data</td>
<td>Demonstrates how to load local data into CAS.</td>
</tr>
<tr>
<td>Prepare and Explore Data</td>
<td>Demonstrates various tools for assaying, assessing, modifying, and preparing data prior to modeling. This snippet uses the Hmeq data set from the Machine Learning sample data library. The Hmeq data set is used as input and creates the Hmeq_prepped data set. The Hmeq_prepped data set is used in subsequent examples.</td>
</tr>
<tr>
<td></td>
<td>Note: You might see a warning message about missing values when you run this snippet. The sample data includes some missing values that can be used by other procedures.</td>
</tr>
<tr>
<td>Snippet Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Compare Two ML Algorithms</td>
<td>Demonstrates fitting and comparing two Machine Learning algorithms for predicting the binary target in the Hmeq data set, which is included in the Machine Learning sample data library. You must run the Load Data and Prepare and Explore Data snippets before you run this snippet.</td>
</tr>
<tr>
<td>Compare Several ML Algorithms</td>
<td>Demonstrates fitting and comparing several Machine Learning algorithms for predicting the binary target in the Hmeq data set, which is included in the Machine Learning sample data library. You must run the Load Data and Prepare and Explore Data snippets before you run this snippet.</td>
</tr>
<tr>
<td>Generalized Linear Models</td>
<td>Demonstrates fitting and assessing generalized linear models using the GENSELECT procedure. You must run the Load Data and Prepare and Explore Data snippets before you run this snippet.</td>
</tr>
<tr>
<td>Unsupervised Learning</td>
<td>Shows the entire workflow process, including data preparation, analysis, and visualization of the results. This snippet uses the Sashelp.Iris data set.</td>
</tr>
<tr>
<td>Supervised Learning</td>
<td>Shows the entire workflow process, including data exploration and preparation, modeling, and evaluation. This snippet uses the Sampleml.Hmeq data set, which is included in the Machine Learning sample data library.</td>
</tr>
<tr>
<td>SAS Viya Image Processing</td>
<td>You must license and install SAS Visual Data Mining and Machine Learning to use these snippets. For more information, see <em>SAS Visual Statistics: Procedures</em>.</td>
</tr>
<tr>
<td>Load Images</td>
<td>Loads the image action set from the specified path and creates a CAS table. The path parameter points to the directory that contains the images that you want to load. The path parameter can specify an image file, a directory that contains image files, or a URL.</td>
</tr>
<tr>
<td></td>
<td>A typical workflow includes loading images and then processing them by using one of the other snippets to resize, rescale, or mutate them. The output from one snippet can be used as input for another snippet. Images can be saved to your computer for further processing by using the Save Images snippet.</td>
</tr>
<tr>
<td>Resize Images</td>
<td>Resizes one or more images based on the HEIGHT and WIDTH parameters that you specify. The HEIGHT parameter corresponds to the number of rows, and the WIDTH parameter corresponds to the number of columns. The input is a CAS table that contains the images that you want to resize, and the output is a CAS table that contains the resized images.</td>
</tr>
<tr>
<td>Snippet Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Rescale Images</td>
<td>Changes the depth of one or more images based on the options that you specify. You can specify any of these values for the TYPE parameter: “TO_8U”, “TO_32F”, or “TO_64F”. You can also use the ALPHA and BETA parameters to scale the values. The input is a CAS table that contains the images that you want to rescale, and the output is a CAS table that contains the rescaled images.</td>
</tr>
<tr>
<td>Mutate Images</td>
<td>Mutates one or more images using different augmentation techniques. You can specify any of these values for the TYPE parameter: ‘COLOR_JITTERING’, ‘COLOR_SHIFTING’, ‘DARKEN’, ‘HORIZONTAL_FLIP’, ‘INVERT_PIXELS’, ‘LIGHTEN’, ‘ROTATE_LEFT’, ‘ROTATE_RIGHT’, ‘SHARPEN’, or ‘VERTICAL_FLIP’. The input is a CAS table that contains the images that you want to change, and the output is a CAS table that contains the mutated images.</td>
</tr>
<tr>
<td>Convert Color</td>
<td>Converts the color space of one or more images. You can specify any of these values for the TYPE parameter: ‘BGR2HSV’, ‘BGR2RGB’, ‘BGR2YUV’, ‘COLOR2GRAY’, ‘GRAY2COLOR’, ‘HSV2BGR’, ‘RGB2BGR’, or ‘YUV2BGR’. The input is a CAS table that contains the images that you want to convert, and the output is a CAS table that contains the converted images.</td>
</tr>
<tr>
<td>Save Images</td>
<td>Saves the specified images to the specified subdirectory of a previously defined caslib, such as CASUSER. You must specify the name of a subdirectory in the caslib root directory. The names of the saved images start with the value of the PREFIX parameter.</td>
</tr>
<tr>
<td>Display Image</td>
<td>Displays an image that is saved from SAS Studio to your computer. This snippet creates an annotation data set that contains a reference to the image and uses PROC SGPLOT to display the image. Use the IMAGE= variable to specify the fully qualified name and location of the image that you want to display.</td>
</tr>
</tbody>
</table>

**Create a Code Snippet**

To create your own snippet:

1. Open your .sas file in SAS Studio and select the code that you want to save as a snippet.

2. On the Code tab, click ![Add to My Snippets](image). The Add to My Snippets dialog box appears.

   **Note:** You can also create a snippet by right-clicking the selected code and selecting Add to My Snippets.

3. Enter a name for the snippet and click Save.

This snippet is now available from the My Snippets folder.
How to Insert a Code Snippet

To include a code snippet in your program:

1. Click the location in your program where you want to insert the snippet.
2. In the navigation pane, open the Snippets section.
3. You can add the snippet to your program in these ways:
   - use a drag-and-drop operation to add the snippet.
   - double-click the name of the snippet.
   - right-click the name of the snippet and select Insert. To select multiple snippets, use the Ctrl key. Then right-click and select Insert.

The following example shows the Import XLSX File snippet inserted into a program.

Customizing the Code Editor

The Preferences window enables you to change several options that affect the features in the code editor, including autocompletion and color coding.

To access the editor options, click and select Preferences. Click Code and Log.

For more information, see “Setting the Code and Log Preferences” on page 133.
What Is a Query?

A query enables you to extract data from one or more tables according to criteria that you specify. You can create a query that is based on only one table, or you can join tables together. When you create a query, SAS Studio generates Structured Query Language (SQL) code, which you can view. You can create queries in both the SAS Programmer and the Visual Programmer perspectives.
Creating a Query

Creating a New Query

On the main SAS Studio toolbar, click and select New Query. A query window opens on a new tab in the work area.

Adding Tables to a Query

On the Tables tab in the query window, click and select Table. From the Choose a Table window, expand the appropriate library and select the table that you want to use. The table is added to the query.

Note: You can also add tables to the query by using the Libraries of the navigation pane. Click Libraries and expand the appropriate library. Drag the table that you want to use to the Tables tab of the query window.
Understanding Joins

Joining Tables

When you create a query, you can join multiple tables together. SAS Studio can automatically join the tables together for you, or you can manually create the join. SAS Studio attempts to join tables by the first column in each table with the same name and data type. If no match for column name and type is found, then you can specify the join criteria.

Note: You can add join conditions by modifying the join that SAS Studio creates. For more information, see “Modifying an Existing Join” on page 46.

Note: If you have more than one table in your query and you do not specify join criteria, then your output data includes the Cartesian product, or every possible combination, of the data values.

Creating a Join

To add a table and automatically create a join:

From the Libraries of the navigation pane, drag the table that you want to add to the query to the Tables tab. Next, drop that table on top of the first table in the query to join the two.

The Join window displays the join criteria. In the following example, the Classfit table is automatically joined to the Class table by using the Name column in both tables.
If a join cannot be created automatically, you can specify the join condition manually.

To manually create a join:

1. On the **Tables** tab of the query window, make sure that you can view the tables that you want to join.

2. Click † on the toolbar and select **Join**. The New Join window opens.

3. From the **Left table** drop-down list, select the table for the left side of the join.

4. From the **Join type** drop-down list, select the type of join that you want to use. The default join type is Inner join.

5. From the **Right table** drop-down list, select the table for the right side of the join.

6. Click **Save**. A join is created between the tables. If the tables include columns with matching names and data types, then a join condition is automatically created. If the tables do not include columns with matching names and data types, then you can select the columns for the join condition from the column drop-down lists.
7. To add another join condition to the join, click + and select the columns that you want to use from the column drop-down lists.

**Understanding the Types of Joins**

SAS Studio supports four different types of joins. You can select the type of join you want by modifying an existing join.

You can select the join option that you want to use in the Join window.

<table>
<thead>
<tr>
<th>SAS Studio Join Type</th>
<th>Join Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inner Join</td>
<td>![Icon]</td>
<td>The output rows include those for which the column in the first table matches the joining criterion of the column in the second table. Joins are inner joins by default.</td>
</tr>
<tr>
<td>Left Join</td>
<td>![Icon]</td>
<td>The output rows include all rows from the first table and the rows from the second table in which the joining criterion is met.</td>
</tr>
</tbody>
</table>
### SAS Studio Join Type

<table>
<thead>
<tr>
<th>SAS Studio Join Type</th>
<th>Join Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Join</td>
<td><img src="image" alt="Right Join Icon" /></td>
<td>The output rows include all rows from the second table and the rows from the first table in which the joining criterion is met.</td>
</tr>
<tr>
<td>Full Join</td>
<td><img src="image" alt="Full Join Icon" /></td>
<td>The output rows include all matching and nonmatching rows from both tables.</td>
</tr>
</tbody>
</table>
To specify an alias for a column:

- On the Select tab, enter the alias that you want to use for each column. The alias is used as the column heading for the output data.
Using Summary Functions

You can perform summary functions on any of the columns in your query. To perform a summary function, select the column on which you want to perform a summary function. Use the drop-down list in the Summary column to select the function that you want to use. By default, the query generates an output data set. The following example shows you how to find the average age of all of the students:
```
5: PROC SQL;
6: CREATE TABLE WORK.QUERY
7: AS
8: SELECT AVG(CLASS.Age)
9:  AS Age
10: FROM SASHELP.CLASS CLASS
11: INNER JOIN SASHELP.CLASSFIT CLASSFIT
```
By default, the query displays the results in the **Output Data** tab and generates an output table in the Work library:

By default, when you summarize a column, your output is grouped by all of the columns without summaries. For more information, see “Grouping Your Output” on page 54.

### Filtering Data

**Creating a Filter**

When you query data, you might want to retrieve only rows that meet certain criteria, based on values of columns in the data. The process of telling SAS Studio which rows to retrieve is called setting a filter and is done on the **Filter** tab. This corresponds to using a WHERE clause in an SQL query.

1. In the query window, click the **Columns** tab to view the list of columns from the tables in the query.
2. You can add one or more columns to the filter by dragging them from the columns list to the **Filter** tab. You can also click the **+** on the **Filter** tab toolbar and select one or more columns from the Choose Column window.
3. Select a comparison operator from the **Operator** drop-down list. The default value is **Equals**.

4. If the operator that you have selected requires a value, click to enter or select a value in the Select Value window. To choose from a list of values, click to expand the **Value** list. Select the values that you want to use and click **Add**.

   **Note:** If you are selecting values for a character column, and you want the values to be enclosed in single quotation marks, select the **Enclose values in quotes** option. This option is selected by default. If you are using a macro variable or other value that is evaluated when the filter is run, you should clear this option.

   If you want to choose from a list of columns, click to expand the **Column** list. Select the column that you want to use.
5. Click **OK** to add the values to the filter.

**Changing the Relationship between Filters**

You can use only one column in a filter, or you can use multiple columns to create several comparison expressions. If you create more than one comparison expression in your filter, then the default relationship between these filter elements is AND. You can change the relationship between filter elements from AND to OR.

To change the relationship between filters:

- On the Filter tab, click the relationship value and select a new value.
Managing Output

**Sorting Your Output**

You can sort the output from your query by one or more columns from the tables that are used in the query.

*Note:* It is possible to sort the output table by columns that are not selected for the output.

To sort your output:

1. In the query window, click the **Sort** tab.

2. You can add one or more columns to the **Sort** tab by dragging them from the columns list to the **Sort** tab. You can also click on the **Sort** tab toolbar and select one or more columns from the Choose Column window.

3. Click the **Sort** box for the column on which you want to sort the data. From the drop-down list, select **Ascending** or **Descending**. The default sort direction is **Ascending**.
4. If you are sorting by multiple columns, the output table is sorted first by the column that is listed first. Within each level of the first column, the rows are sorted by the second column in the list, and so on. You can change the sort order by selecting a column and clicking and to move the column up and down the list.

Eliminating Duplicate Rows in Output

Some types of queries output multiple, identical rows. Because these duplicate rows are generally not useful, SAS Studio enables you to keep only one of the identical rows and eliminate the duplicates.

To eliminate duplicate rows, click the Select tab and select the Select distinct rows only check box.

Grouping Your Output

If you have created a summarized column, you can choose to classify your data into groups based on the values in a column. This is equivalent to using the GROUP BY clause in an SQL query. For example, if you are calculating the average height of a group of students, you might want to group the results by age so that you can see the average height for each age group.

By default, the Automatically select groups option is selected on the Group tab. When this option is selected and you have performed a summary function on a column, your
query is automatically grouped by all columns without summary functions. You can choose to edit the list of columns that the query is grouped by.

To group your output:

1. In the query window, click the **Group** tab.

2. You can add one or more columns to the **Group** tab by dragging them from the columns list to the **Group** tab. You can also click 🔄 on the **Group** tab toolbar and select one or more columns from the Choose Column window.

   **Note:** To remove all of the automatically selected columns from the **Group** tab, clear the **Automatically select groups** option.

3. To change the order in which the columns are used to group the data, select the column that you want to move and click ↑ and ↓.

The following example shows you how to find the average weight of students in each age group. First, add the Age and Weight columns to the Select tab, and then select the AVG summary function for the Weight column.
To see the average weight of students by age, the query is grouped by the Age column. The results show the average weight for each age group:
Note: By default, the query generates a table of the result. To generate a report of the results (which is displayed in Results tab), you must specify report as the output type for the query. For more information, see “Saving Your Results” on page 57.

**Saving Your Results**

You can choose to generate your results in any one of three formats: data table, data view, or report.

If you save your results as a data table or data view, you can specify the library and filename that you want to use. If you don’t specify the library and filename, the results are saved in the Work library.

To specify the results format:

1. In the query window, click the **Settings** tab.
2. Click the **Properties** tab. In the Results area, select the format that you want to use from the **Output type** drop-down list.

   - **Report** saves the query results as a report that you can download as an HTML, PDF, or RTF file. Query results in this format are not updated until you rerun the query. You cannot run SAS tasks against query results in this format.
Table saves the query results as a static data table against which you can run SAS tasks. Query results in this format are not updated until you rerun the query. By default, the data table is stored in the Work library.

View saves the query results as a dynamic data view against which you can run SAS tasks. Each time you open query results in the data view format, the results are updated with any changes to the data that is used in the query. By default, the data view is stored in the Work library.

To save your results to a specific location:

1. In the query window, click the Settings tab.
2. Click the Properties tab. In the Results area, enter the name of the library in which you want to save your results in the Output location box.
3. To specify a name for the results, enter the name that you want to use in the Output name box.

Running a Query

After you specify all the criteria for your query, you can generate your results by clicking on the query window toolbar. The output data opens in the workspace on a separate tab.
Chapter 4
Working with Process Flows

Understanding Process Flows

What Is a Process Flow? ................................. 59
Creating Process Flows .................................. 59
What Are Ports? ........................................ 60
Understanding the Status of Each Node .................. 60
Customizing a Process Flow ............................. 61
Viewing the Properties of a Process Flow .................. 63

Add Data to the Process Flow .................. 63
Adding a SAS Program to the Process Flow .......... 64
  Create a New SAS Program .......................... 64
  Add an Existing SAS Program ...................... 66
  Add a Snippet .................................... 66

Adding a Query to a Process Flow .................. 69
  Create a New Query .................................. 69
  Add an Existing Query to a Process Flow ............ 69

Add a Task to a Process Flow ...................... 70
Understanding Subflows ............................... 71
  What Is a Subflow? .................................. 71
  Create a New Subflow ............................... 72
  Add a Subflow to an Existing Process Flow ........... 73

Linking Nodes in a Process Flow .................... 73
Generating Code from a Process Flow ............... 73
Running a Process Flow ..................................... 73
Save a Process Flow ..................................... 74

Understanding Process Flows

What Is a Process Flow?

You must be working in the Visual Programming perspective to work with process flows. For more information about the Visual Programming perspective, see “Understanding Perspectives” on page 12.
A process flow consists of one or more objects. Each object is represented by a node in the process flow. The process flow shows the relationship between two or more objects, such as a SAS program, a task, a query, and so on.

This sample process flow contains three branches.

- In the first branch, a query is created for the Sashelp.Classfit data set. As a result of this query, only observations where Sex=M are included in the output data set (called Work.Query). The Rank Data task ranks the Height and Weight values by Age.

- In the second branch, you write a SAS program that generates an output data set. After this program runs, a subflow that contains more program and task nodes runs. Then, the Bar Chart node runs.

- In the third branch, the SAS program generates an error. The information for the **List Table Attributes** node is incomplete. SAS Studio cannot run the process flow until you provide the required data.

---

**Creating Process Flows**

In SAS Studio, you can have multiple process flows. These process flows run independently of each other. To create a new process flow, click ![New Process Flow](image) and select **New Process Flow**. A new **Process Flow** tab appears in the interface.

**What Are Ports?**

In a process flow, you can have two types of ports:

- control ports

  In the process flow, these types of ports appear as ![Control Port](image). You use these ports to specify the order in which nodes run in the process flow. For more information, see “**Linking Nodes in a Process Flow**” on page 73.
data ports

In the process flow, these types of ports appear as 🔄.

When you run a task, you must specify an input data source. You specify the input data source in the task interface. From the process flow, you can determine the name of the input data source by positioning your mouse pointer over the input data port. To view the data source on a separate tab, double-click 🔄.

In this example, the input data source for the Rank Data task is Work.Query.

Some nodes, such as queries, might have an output data source. You can use the output data port to determine the name of the output data source. To view the data source, double-click 🔄.

In this example, the output data source for the query node is Work.Query.

By default, ports are displayed in the process flow. To turn off the ports, click Show Ports.

**Understanding the Status of Each Node**

When you run a process flow, some nodes might run successfully, and others might not. To see the status of each node, how long it took to run the node, and whether any output data was created, click the Results tab.
Here are the contents of the **Results** tab for the previous process flow.

You can use these icons to determine the status of each node in the process flow:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="" alt="Icon" /></td>
<td>specifies that more information is needed before SAS Studio can run the node. For example, if you see this icon on a task node, you must specify values for the required options in the task. This icon can also appear on an empty subflow node after SAS Studio tries to run it.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>specifies that the node is in the queue to run.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>specifies that the code for the node was submitted successfully. No warnings or errors were returned.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>specifies that the code for the node generated a warning. Review the log for that node for more information.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>specifies that the code for the node generated an error. Review the log for that node for more information.</td>
</tr>
</tbody>
</table>
**Customizing a Process Flow**

You can customize your process flow in these ways.

- To specify a color for a node or a group of nodes, select the nodes and click **Color**. From the drop-down list, select the color that you want to use.
- By default, you can arrange objects in the process flow any way you like. However, your process flow might become confusing if it contains many objects. Click **for SAS Studio** to arrange the objects in your process flow. When arranging the nodes, SAS Studio considers any dependencies and the order in which the nodes were added to the process flow.

**Viewing the Properties of a Process Flow**

To view the properties of the current process flow, click the **Properties** tab. From the properties, you can specify the priority of execution of the nodes. The process flow runs the nodes in the order in which the nodes are added to the process flow. If node 1 is dependent on another node 2, node 2 must run completely before node 1 will run. You can also run the nodes in parallel, which means that multiple workspace servers are used to run the nodes. As a result, the nodes might not share a common Work library.

**Add Data to the Process Flow**

To add data to a process flow from an existing library:

1. In the navigation pane, click the **Libraries** section.
2. Expand the library that contains the data set that you want to add. Select the data set and drag it to the process flow.
In this example, the Sashelp.Air data set is now available from the process flow. You can now use this data set as the input data for a task.

Adding a SAS Program to the Process Flow

Create a New SAS Program

To add a new SAS program to a process flow:

1. Click \( \text{+} \) and select **SAS program**. A node for the SAS program is added to the process flow.
2. Select the node and click \( \text{=} \). (You can also select the node and press Enter.) The code editor appears.
3. Enter the code for your program.
4. To name the program, to provide a brief description for the program, and to include any notes, click the **Node** tab.

The SAS program node in the process flow now contains the name and description that you specified on the **Node** tab.
Add an Existing SAS Program

You might have already written a SAS program that you want to include in the process flow. To add this program:

1. In the navigation pane, click the Server Files and Folders section.
2. Expand the folders in the Server Files and Folders section until you find the program that you want to add.
3. Select the program that you want to add, and then drag it to the process flow. (A green check mark indicates that you can add this file to the process flow.)

Add a Snippet

You can use snippets as the starting point for your SAS programs.

To add a snippet to a process flow:

1. In the navigation pane, select Snippets.
2. In the Snippets section, select the snippet that you want to add, and then drag it to the process flow.
In this example, the Import XLSX file snippet is added to the process flow.

3. To view the code for the snippet, select the snippet node and click \( \text{Modify} \). (You can also select the node and press Enter.) Edit the code to meet your needs.

Here is the code for the Import XLSX file snippet. In this example, specify in the code the location of the XLSX file that you want to import.
4. To name the program that you created, to provide a brief description for the program, and to include any notes, click the **Node** tab.

The **Import XLSX File** node in the process flow now includes the description that you provided.
Adding a Query to a Process Flow

Create a New Query

To create a new query:

1. Click + and select Query. A node for the query is added to the process flow.
2. Select the node and click . The Query Builder appears.
3. Use the Query Builder to define your query. For more information, see “Creating a New Query” on page 42.
4. To define the type of results for the query, use the Properties tab. For more information, see “Saving Your Results” on page 57.
5. To name the query node, to provide a brief description for the query, and to include any notes, click the Node tab.

To return to the view of the process flow, select the name of the process flow in the breadcrumbs. The query node in the process flow now contains the name and description that you specified on the Node tab.

Add an Existing Query to a Process Flow

You might have already written a query that you want to include in the process flow. To add this query:

1. In the navigation pane, click the Server Files and Folders section.
2. Expand the folders in the Server Files and Folders section until you find the query that you want to add.
3. Select the query that you want to add, and then drag it to the process flow. (A green check mark indicates that you can add this file to the process flow.)

---

**Add a Task to a Process Flow**

You can add custom tasks and tasks that shipped with SAS Studio to your process flow.

1. In the navigation pane, click the **Tasks and Utilities** section.
2. Select the task that you want to add, and then drag it to the process flow.

   Here is an example of a process flow that contains the Sort Data task.

3. Select the task node and click ![Run Task](Image). To run the task, you must specify values for any required options.
Here is the user interface for the Sort Data task.

When you run the task, SAS Studio uses the values that you specified for the task options.

Understanding Subflows

What Is a Subflow?

Within a process flow, you could have subflows. These subflows contain one or more objects. The advantage to creating a subflow is that you can easily run the nodes within a subflow without running the entire process flow. If the subflow links to another node in the entire process flow, the subflow must run to completion before SAS Studio runs the subsequent node.

In this example, there are three nodes: a node for a subflow, a query node, and a programming node. All of the nodes in the subflow node must run before the query node can run. By default, the node for a subflow is white.
Note: SAS Studio cannot run a process flow that contains an empty subflow. An error message appears instead.

Create a New Subflow

To add a subflow to a process flow:

1. Click and select Sub-Flow. A node for the subflow appears in the process flow.
2. Select the subflow node and click . Use the breadcrumbs to verify that you are working in the subflow.
3. Add the content for the subflow. A subflow can contain other subflows.
4. To name the subflow node, to provide a brief description for the subflow, and to include any notes, click the Node tab.

After you enter this information, the subflow node in the process flow contains the name and description that you specified on the Node tab.
Add a Subflow to an Existing Process Flow

You might want to add an existing process flow as a subflow to another process flow. Process flow files have a CPF extension.

To add an existing process flow as a subflow:

1. In the navigation pane, click the Server Files and Folders section.
2. Expand the folders in the Server Files and Folders section until you find the program that you want to add.
3. Select the subflow that you want to add, and then drag it to the process flow. (A green check mark indicates that you can add this file to the process flow.)

Linking Nodes in a Process Flow

In a process flow, you can specify the order in which each node runs. Generally, each node has an input port and an output port. To view the ports for the nodes in your process flow, click Show Ports.

To link nodes in the process flow:

1. Select the node that you want to link from.
2. Click + and select Link. The Add Link window appears.
3. Select the nodes that you want to link to and click OK.
   
   If a link is allowed between the two nodes, the two nodes are connected by a dotted line in the process flow.

Generating Code from a Process Flow

You can create a SAS program from the nodes in a process flow. The code is listed in the program in the order in which it runs in the process flow.

To create a SAS program from a process flow, click Generate Code on the process flow toolbar. The code is added to a new program tab in the work area.

Running a Process Flow

To run all the nodes in the process flow, click Run.

To run a select group of nodes, use the mouse to draw a box around the nodes that you want to run. Right-click your selection. From the pop-up menu, select Run Selected.
Save a Process Flow

To save the current process flow, click 📌. The process flow is saved as a CPF file.
Chapter 5
Working with Data

About the Table Viewer ......................................................... 75
Opening and Viewing Data ..................................................... 77
Viewing the Code That Is Used to Create a Table ....................... 79
Filtering and Sorting Data ..................................................... 80
Importing Data ........................................................................ 83
  About Importing Data to SAS Studio .................................... 83
  Import an Excel Worksheet .................................................... 84
  Import a Delimited File ........................................................ 87
  Import a DBMS File ............................................................. 90
  Importing Data in a Process Flow ......................................... 91
  Save the Import Task ........................................................... 91
Exporting Data ....................................................................... 92

About the Table Viewer

When you open a table in SAS Studio, you use the table viewer.
Note: The table viewer displays the first 100 rows of the table. If the structure or data values of the table change while the table is open, you must refresh the table viewer to see the changes. If the structure of the table changes and you do not refresh the table, the columns that are listed in the Libraries section of the navigation pane might be different from the columns that are displayed in the table viewer.

You can view the properties of the table and its columns by clicking on the toolbar.
The extended attributes tabs enable you to associate additional user-defined characteristics with the table and columns in the table. For example, you could create extended attributes that contain a URL with information about your table or the formula that is used to create a column. For more information about creating extended attributes, see *Base SAS 9.4 Procedures Guide*.

*Note:* You can add extended attributes only to SAS tables that were created on a SAS 9.4 (or later) server.

---

**Opening and Viewing Data**

You can open files in SAS Studio in several ways:

- You can double-click a file in the *Server Files and Folders* and *Libraries* sections.
- You can drag a file from the *Server Files and Folders* and *Libraries* sections to the work area.
- You can search for a file and open it from the search results. You can open the file by double-clicking it or by dragging it to the work area.
- You can open a file by using a file shortcut in the File Shortcuts section. You can open the file by double-clicking it or by dragging it to the work area.

Note: SAS Studio provides native file support for z/OS. For more information about SAS in z/OS environments, see SAS Companion for z/OS.

When you open a table, the first 30 columns in the table are displayed. You can use the Columns area to specify which columns you want to include in the table viewer. If your table has more than 100 columns and you select Select all, you are prompted to confirm that you want to display all of the columns. Displaying more than 100 columns can affect your performance.

Note: If you are using Internet Explorer and you display more than 100 columns in a table, SAS Studio reverts to displaying only 30 columns when you close and reopen the table.

By default, the column names are displayed, but you can choose to display the column labels by selecting Column labels from the View drop-down list.

Note: By default, the table viewer displays the total number of rows in the table and the total number of filtered rows, if you have filtered the data. However, if SAS Studio is unable to determine the row counts without affecting performance, then the row counts are listed as “Unavailable.”

You can automatically resize the column widths to fit the current size of the column content. To resize the column widths, right-click any column heading and select Size grid columns to content. To set the columns back to their default widths, right-click any column heading and select Restore original column widths. You can choose to always resize the column widths when you open data by selecting Size grid columns to content in the General preferences. For more information, see “Setting General Preferences” on page 131.

Note: The column widths might need to be recalculated when the column content changes due to filtering, sorting, or paging through the data and can affect performance.

You can change the order of the columns in the table viewer by dragging a column to a new position in the table viewer or in the Columns area.
While you select options and customize the table to look the way you want it to, SAS Studio is generating SAS code that you can use. To view the code, click on the toolbar. A new program window appears with the code that was used to create the view of the table in the table viewer. The program is a copy of the code and is no longer associated with the original code. Editing the code does not affect the data that is displayed in the table viewer, and modifying the table viewer does not affect the contents of the code.
Filtering and Sorting Data

In the table viewer, you can right-click a column heading to filter and sort the data by that column. You can sort the data in ascending or descending alphabetical order or display the columns in the order in which they appear in the data table.

You can also filter the data in order to display only rows that meet certain criteria, based on values in the data. The filter options vary depending on the type of column that you have selected and the number of distinct values that a column has.

The Add Filter window for a numeric column enables you to specify one or two filter criteria for each column. To add a second filter criterion, click +.
The Add Filter window for a character column is case sensitive and searches for values that contain the text that you enter.

When a numeric column has 10 or fewer distinct values or a character column has 30 or fewer distinct values, the Add Filter window displays a list of values to choose from. The list of values includes both the unformatted and formatted values. If no format has been applied to the data, then the unformatted and formatted values are the same.
Note: SAS Studio always uses the unformatted values in the filter expression.

When you create a filter on a numeric column by selecting values from a list, SAS Studio filters the data differently depending on whether any of the values in the column have a fractional component.

If the values are all integers, then SAS Studio creates the filter by using the equality operator. For example, suppose your data includes integer values between 12 and 17, and you want to create a filter for all values equal to 15. SAS Studio creates the following filter expression:

\( \text{column-name} = 15 \)

If any of the values include a fractional component, then SAS Studio creates the filter by using a range of values to ensure that there are no rounding errors that might exclude the selected value. For example, suppose your data includes a value of 123.45678. If you select that value to use in a filter, SAS Studio creates the following filter expression:

\( (\text{column-name} \geq 123.456775 \text{ and } \text{column-name} < 123.456785) \)
Note: Generating a list of values for a filter can take a long time for large tables and Hadoop tables and can affect performance. If you are creating a filter on a very large table or on a Hadoop table, you cannot select the filter values. You must enter them in the value box. By default, if a table has more than 50,000 rows or the total number of rows is unknown, then you must enter filter values for it. The default value is controlled by a setting in the config.properties file. For more information, see *SAS Studio: Administrator’s Guide*.

The Add Filter window for a date column enables you to select a date value from a pop-up calendar.

![Add Filter Window](image)

When you create a filter on your data, the filter criteria are displayed at the top of the workspace. You can click \(\rightarrow\) to edit the filter and \(\rightarrow\) to delete the filter.

Note: The filter is not saved with the data set. When you reopen the data set, the filter is not available.

---

**Importing Data**

*About Importing Data to SAS Studio*

You can import these types of data files into SAS Studio:

- Microsoft Access database files.
- delimited files, such as files with comma-separated values.
- dBASE 5.0, IV, III+, and III.
- Stata files.
- Microsoft Excel files. To import XLSB and XLSM files, you must use the SAS LIBNAME statement.
- JMP files.
- Paradox DB files.
- SPSS files.
- Lotus 1-2-3 files from Releases 2, 3, 4, or 5.

If you are using the SAS Studio Enterprise Edition or the SAS Studio Basic Edition, your data file might be saved to your local computer. In this case, you must upload the file to SAS Studio before you can import it.

Whether data from another locale imports correctly depends on whether the SAS server supports the locale of the data that is being imported. If you are importing data that contains characters that are different from the current locale, use a Unicode (UTF–8) server to import your data. If you do not use a UTF–8 server and the locale of the data is not supported, unsupported values might appear as question marks (?) in your imported data. For more information about how to set the Default text encoding option, see “Setting General Preferences” on page 131.

*Note:* You cannot import remote files (files that are available through FTP file shortcuts).

**Import an Excel Worksheet**

To import an Excel worksheet:

1. Click Server Files and Folders in the navigation pane and browse to find the file that you want to import.

2. Right-click the file that you want to import and select Import Data. The top of the Import Data tab shows the name and location of this Excel file. It also shows several options that you can customize.
This example shows importing the as_products.xls file.

3. To import the data from a specific worksheet, enter the name of that worksheet in the **Worksheet name** box. By default, SAS Studio imports the data from the first worksheet.

4. To specify the location to save the output data set, click **Change**. By default, the output data set is saved to the Work library, which is a temporary location. The contents in this library are deleted when you exit SAS Studio.

5. To generate SAS variable names from the data values in the first row of the worksheet, select **Generate SAS variable names**. If a data value in the first row in the input file is read and it contains special characters that are not valid in a SAS name, such as a blank, then SAS converts the character to an underscore.
6. To import the Excel worksheet, click 

The Results tab shows the attributes of the new SAS data set.

<table>
<thead>
<tr>
<th>CODE</th>
<th>LOG</th>
<th>RESULTS</th>
<th>OUTPUT DATA</th>
</tr>
</thead>
</table>

Table of Contents

The CONTENTS Procedure

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>WORK.IMPORT</th>
<th>Observations</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member Type</td>
<td>DATA</td>
<td>Variables</td>
<td>8</td>
</tr>
<tr>
<td>Engine</td>
<td>V9</td>
<td>Indexes</td>
<td>0</td>
</tr>
<tr>
<td>Created</td>
<td>10/26/2016 10:50:16</td>
<td>Observation Length</td>
<td>80</td>
</tr>
<tr>
<td>Last Modified</td>
<td>10/26/2016 10:50:16</td>
<td>Deleted Observations</td>
<td>0</td>
</tr>
<tr>
<td>Protection</td>
<td></td>
<td>Compressed</td>
<td>NO</td>
</tr>
<tr>
<td>Data Set Type</td>
<td></td>
<td>Sorted</td>
<td>NO</td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Representation</td>
<td>SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encoding</td>
<td>utf-8 Unicode (UTF-8)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Engine/Host Dependent Information

<table>
<thead>
<tr>
<th>Data Set Page Size</th>
<th>65536</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Data Set Pages</td>
<td>1</td>
</tr>
<tr>
<td>First Data Page</td>
<td>1</td>
</tr>
<tr>
<td>Max Obs per Page</td>
<td>817</td>
</tr>
<tr>
<td>Obs in First Data Page</td>
<td>60</td>
</tr>
<tr>
<td>Number of Data Set Repairs</td>
<td>0</td>
</tr>
</tbody>
</table>
The **Output Data** tab shows the contents of the new data set. If this data set is in the Work library (as shown in this example), you might want to save it to a more permanent location. Data in the Work library is temporary and is deleted when you exit SAS Studio.

---

**Import a Delimited File**

*Note:* For some delimited files (such as files with a .dat extension), the Import Tool might not be available. To import these delimited files, save the file as a text file. Then you can use the Import Tool. To import a tab-delimited file, the filename must have a TAB extension.

To import a delimited file:

1. Click **Server Files and Folders** in the navigation pane and browse to find the file that you want to import.

2. To determine the delimiter for the file, right-click the filename and select **View File as Text**. The contents of the file open in a text editor. Note the delimiter that is used between values and whether the first row of the data includes headings.

3. After determining the delimiter of this file, click and select **New Import Data**. The **Import Data** tab opens in the SAS Studio workspace.
4. Drag the selected text file from Server Files and Folders in the navigation pane to the Import Data tab. The top of the Import Data tab now shows the properties of the file (such as the filename and location of the saved file), the name of the imported data source, and any options that you can specify. The bottom of the Import Data tab shows the SAS code that has been generated.

This example shows importing a text file called CommaSep.txt.

5. To specify the location to save the output data set, click Change. By default, the output data set is saved to the Work library, which is a temporary location. The contents in this library are deleted when you exit SAS Studio.

6. To generate SAS variable names from the data values in the first row in the text file, select Generate SAS variable names. If a data value in the first row in the input file is read and it contains special characters that are not valid in a SAS name (such as a blank), SAS converts the character to an underscore.

7. In the Delimiter box, enter the delimiter for the values in the file. The default delimiter is a space.

Note: If you use a hexadecimal value to specify the delimiter, you do not need to select the Quote delimiter check box.
8. (Optional) To start reading data from a specified row in the delimited text file, enter the starting row in the **Start reading data at row** box. You might want to use this option if you have comments at the top of the text file or the first row of the file is column headings.

9. (Optional) For SAS Studio to determine the appropriate data type and length of the variables, enter a value in the **Guessing rows** box. The task scans the input data file from row 1 to the number that you specified. By default, the first 20 rows are scanned.

10. To import the data, click **.**

    Click the **Results** tab to see the attributes of the imported data set.

---

**Table of Contents**

<table>
<thead>
<tr>
<th>The CONTENTS Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Set Name</strong></td>
</tr>
<tr>
<td><strong>Member Type</strong></td>
</tr>
<tr>
<td><strong>Engine</strong></td>
</tr>
<tr>
<td><strong>Created</strong></td>
</tr>
<tr>
<td><strong>Last Modified</strong></td>
</tr>
<tr>
<td><strong>Protection</strong></td>
</tr>
<tr>
<td><strong>Data Set Type</strong></td>
</tr>
<tr>
<td><strong>Label</strong></td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
</tr>
</tbody>
</table>

**Engine/Host Dependent Information**

| **Data Set Page Size** | 65536                   |
| **Number of Data Set Pages** | 1                           |
| **First Data Page**   | 1                       |
| **Max Obs per Page**  | 900                     |
| **Obs in First Data Page** | 74                        |
| **Number of Data Set Repairs** | 0                         |
Click the **Output Data** tab to view the new SAS data set. If this data set is in the Work library (as shown in this example), you might want to save it to a more permanent location. Data in the Work library is temporary and is deleted when you exit SAS Studio.

---

**Import a DBMS File**

When you import a file from a database management system (DBMS), the available options depend on the file type. For a list of the supported file types, see “About Importing Data to SAS Studio” on page 83.

1. Click **Server Files and Folders** in the navigation pane and browse to find the file that you want to import.

2. Right-click the file that you want to import and select **Import Data**. The **Import Data** tab opens. The options that are available on this tab depend on the file type.

3. To specify the location to save the output data set, click **Change**. By default, the output data set is saved to the Work library, which is a temporary location. The contents in this library are deleted when you exit SAS Studio.

4. (Optional) To generate SAS variable names from the data values in the first row of the worksheet, select **Generate SAS variable names**. If a data value in the first row in the input file is read and it contains special characters that are not valid in a SAS name, such as a blank, then SAS converts the character to an underscore.

5. To import the file, click **”**
Importing Data in a Process Flow

If you are using the SAS Visual Programmer perspective, you can also import data by using the process flow. You might want to add an import node if the file that you want to import is updated frequently.

To create the import node, drag the filename that you want to import into the process flow.

Here is an example of an import node (called Import as_products.xls) in a process flow. The imported data is then used in the Summary Statistics task.

Save the Import Task

You might want to save an instance of the Import Data tool so that you can share these settings for importing a specific file with others at your site. SAS Studio saves these instances as a CTL file. CTL files must be run in the same operating environment where they were created. For example, if you create a CTL file using Windows, this CTL file must be run in Windows.

To save the import task:

1. Click Save.

2. In the Save As window, specify the name and location, and then click Save. The file is saved with a CTL extension.
Exporting Data

You can use SAS Studio to export your data as another file type to a folder that you specify.

Note: You cannot export your data to an FTP folder.

To export your data:

1. Click Libraries in the navigation pane and browse to find the file that you want to export.

2. Right-click the file that you want to export and select Export. The Export Table window opens.

3. Select the folder in which you want to save the exported file.

4. In the Filename box, enter the name of the exported file.

5. From the File format drop-down list, select the format of the exported file.

6. Click Export to export the file.
## Viewing Results

When you run a task or a program in SAS Studio, the results are displayed in the work area. Here are ways that you can manage your results:

- Generate HTML5, PDF, and RTF output by default, and view the HTML5 output on the **Results** tab.
- Download your generated output. There is a download button for each of the three default output types.
- Change the default output style for each destination by using the Preferences window.
- Send results to another user.

## Default SAS Studio Output

### Viewing Default Results

In SAS Studio, by default, output is generated in the HTML5, PDF, and RTF formats.
If you want to change the default output, you can use the Preferences window to disable results in PDF or RTF format. You can also change the default style for your output to any of the ODS styles that are available. For more information, see “Setting the Results Preferences” on page 136.

By default, the HTML5 results are the only results that are displayed on the Results tab.

The PDF and RTF output is generated but is not displayed.

When you view your results, you can use the table of contents to navigate through the different sections. Click Table of Contents at the top of your results, and then click the section that you want to navigate to.
Downloading Default HTML5, PDF, and RTF Results from SAS Studio

If you want to save results from SAS Studio, you can download your results in the HTML5, PDF, or RTF output formats and save them or open them in the default application for that format:

- HTML5 file
- PDF file
- RTF file

Downloading Generated Data

You can download results that were generated as other types of data, such as a .csv, .pptx, or .xml file, by clicking 🖼.
Sending Your Results to Another User

You can send a copy of your results and the associated code and log files to another user through electronic mail. Files that you can send include results in HTML5, RTF, and PDF formats as well as the code and log files that are associated with the results. You can also send a Program Summary file, which includes information about the program execution, the complete SAS source code, the complete SAS log, and the results. The code is sent as a SAS program file, and the log and program summary files are sent as HTML5 files. To send files through email, you need access to an SMTP server. For more information, contact your site administrator.

Note: If your SAS Studio email messages are being marked as junk mail, see SAS Studio: Administrator’s Guide for information about your configuration file.

To send results by email:

1. On the toolbar for your results, click \( \text{Send Email} \). The Send Email window appears.

2. Select the items that you want to include as attachments to your email. By default, the HTML results are selected.

3. In the To box, enter the email addresses to which you want to send the files. Separate addresses with a semicolon.

4. If you want to send a copy of the email to another address, enter the address in the Cc box.

5. In the Subject box, enter a subject for the email. You can also add a message to include in the body of the email.

6. Click Send to send the message and attachments.
About the SAS Output Delivery System

The SAS Output Delivery System (ODS) gives you greater flexibility in generating, storing, and reproducing SAS procedure and DATA step output along with a wide range of formatting options. ODS provides formatting functionality that is not available when using individual procedures or the DATA step without ODS.

SAS Studio uses very specific ODS options and the GOPTIONS statements so that the output is displayed properly in the web environment. To view all of the ODS options in your code, click and select Preferences. In the Preferences window, click General and select the Show generated code in the SAS log option.

Note: To ensure that your output is displayed properly, do not change the settings of the ODS options or GOPTIONS statements in the generated code.

About SAS ODS Statistical Graphics

SAS ODS Statistical Graphics, more commonly referred to as SAS ODS Graphics, is an extension of the SAS Output Delivery System (ODS). ODS manages all output that is created by procedures and enables you to display the output in a variety of forms, including HTML and PDF.

Many SAS analytical procedures use ODS Graphics functionality to produce graphs. ODS Graphics uses the Graph Template Language (GTL) syntax, which provides the power and flexibility to create many complex graphs. The GTL is a comprehensive language for defining statistical graphics.

In SAS Studio, you can use the ODS Graphics Designer to define these statistical graphics without knowing the GTL. After a graph definition is created, you can use that graph definition to create an ODS statistical graph in SAS Studio.

SAS ODS Graphics Designer

What Is the SAS ODS Graphics Designer?

The SAS ODS Graphics Designer is an interactive graphical application that you can use to create and design custom graphs. The designer creates graphs that are based on the Graph Template Language (GTL), which is the same language that is used by SAS analytical procedures and SAS ODS Graphics procedures. The ODS Graphics Designer provides a graphical user interface so that you can design graphs easily without knowing the details of templates and the GTL.

Using point-and-click interaction, you can create simple or complex graphical views of data for analysis. The ODS Graphics Designer enables you to design sophisticated graphs by using a wide array of plot types. You can design multi-cell graphs, classification panels, and scatter plot matrices. Your graphs can have titles, footnotes, legends, and other graphics elements. You can save the results as an image for inclusion in a report or as an ODS Graphics Designer file (SGD) that you can later edit.
For more information, see *SAS ODS Graphics Designer: User’s Guide*, which is available from support.sas.com.

**How to Install the SAS ODS Graphics Designer**

If you have SAS Foundation installed on your machine, the SAS ODS Graphics Designer is already available. For example, if you are using the single-user edition of SAS Studio, the SAS ODS Graphics Designer is already installed because you are running SAS Foundation and SAS Studio on the same machine.

*Note:* Only the Windows version of the SAS ODS Graphics Designer is supported when you are running SAS Studio. The SAS ODS Graphics Designer must be installed on the same machine as the browser that you are using to access SAS Studio. (SAS Studio might be installed on a different machine.) If multiple users are accessing SAS Studio and these users are on different machines, SAS ODS Graphics Designer must be installed on each user’s machine.

To install the SAS ODS Graphics Designer:

1. Click ![Install](image). Select **Tools ⇒ Install ODS Graphics Designer**. The software page for SAS Studio opens.

2. In the right navigation pane under the **Software Downloads** heading, click **SAS ODS Graphics Designer**. The downloads page for SAS ODS Graphics Designer opens.

3. From the table, click the link in the **Request download** column for your operating environment and follow the subsequent installation steps.

**Open the SAS ODS Graphics Designer**

After the SAS ODS Graphics Designer is installed, you can open it by using a menu option in SAS Studio. To open SAS ODS Graphics Designer, click ![Open](image). Then select **Tools ⇒ ODS Graphics Designer**.
SAS ODS Graphics Editor

What Is the SAS ODS Graphics Editor?
The ODS Graphics Editor enables you to edit the various elements in the output graph while keeping the underlying data unchanged. In addition, you can annotate a graph by inserting text, lines, arrows, images, and other items in a layer above the graph. You can save the results of your customization as an ODS Graphics Editor (SGE) file and make incremental changes to the file. You can also save the results as a Portable Network Graphics (PNG) image file for inclusion in other documents.

For more information about the SAS ODS Graphics Editor, see SAS ODS Graphics Editor: User’s Guide, which is available from support.sas.com.

How to Install the SAS ODS Graphics Editor
Note: If you are running the single-user edition of SAS Studio, then the SAS ODS Graphics Editor is already installed.

To install the SAS ODS Graphics Editor:
1. Click . Then select Tools ⇒ Install ODS Graphics Editor. The software page for SAS Studio opens.

2. In the right navigation pane under the Software Downloads heading, click SAS ODS Graphics Editor. The downloads page for SAS ODS Graphics Editor opens.
3. From the table, click the link in the **Request download** column for your operating environment and follow the subsequent installation steps.

**How to Edit Your Graphics Output**

1. Include this statement in your SAS code so that you can edit your graphics output:
   
   ```sas
   ods listing sge on gpath="directory-path";
   ```
   
   When you run this program, the graphical output is saved as an SGE file in the directory specified by the GPATH= option.

2. In the **Server Files and Folders** section of the navigation pane, double-click the filename (with SGE extension) to open the graph in the SAS ODS Graphics Editor.

For example, here is the SGPanel1.sge file in the SAS ODS Graphics Editor.
Chapter 7
Understanding Git Integration in SAS Studio

About Git Integration in SAS Studio

SAS Studio includes integration with Git, a system for tracking changes and managing version control among multiple users. Git can be used with many different repository hosting services such as GitHub and BitBucket.

You can use the Git Repository section of the navigation pane to access the basic Git features from within SAS Studio, including cloning repositories, committing changes to files, pulling and pushing files, viewing your repository history, creating and merging branches, and performing a basic differentiation between files in your local repository. By default, SAS Studio uses SSH keys to authenticate with your repository hosting service. You can also authenticate using HTTPS with a user name and password.

Creating a Git Profile

Before you can get started using any of the Git features in SAS Studio, you must create a Git profile. By default, SAS Studio uses SSH keys for authentication. In order to use the
SSH authentication, you must upload your public and private SSH keys to the workspace server. For information about generating SSH keys, see the documentation for your repository hosting service.

**Note:** You can upload your SSH key files to the workspace server by clicking the **Server Files and Folders** section of the navigation pane. Specify the directory that you want to use on your workspace server, and then click **Choose Files** to browse for the SSH key files.

To create a Git profile, click **Git Repository** in the navigation pane, and then click **Create a Profile**. The Add a Profile window appears.

**Note:** If you already have an existing Git profile, the option to create a profile is not available from the **Git Repository** section of the navigation pane. To create an additional profile, click **+** and select **Preferences**. Click **Git Profiles**, and then click **+** to open the Add a Profile window.

To create a Git profile using SSH keys, enter your profile name, user name, and a valid email address in the appropriate boxes. Click **Browse** to find your public and private SSH key files or enter the path names in the appropriate boxes. Click **OK** to create your profile.

To create a Git profile using HTTPS authentication, click **Switch to HTTPS** and then enter your profile name, user name, password, and a valid email address in the appropriate boxes. This option is available only if you have installed the SAS Studio 3.8 hot fix and your SAS Studio administrator has set `webdms.showGitPassword=true` in the config properties file. You can download the hot fix from [SAS 9.4 Hot Fix Downloads](https://support.sas.com/software/sas9/).
Cloning a Repository

In order to access the files from a Git repository in SAS Studio, you must clone the repository to a local repository on your workspace server. When you clone the repository, a copy of all the files on the remote repository are copied to a working directory that you specify on your workspace server. You can clone a repository from a remote server using SSH or HTTPS authentication, and you can also clone a local repository from your server file system.

*Note:* If an existing Git repository has been cloned to the workspace server, then it is already a local repository and is available for you to point to. However, if the Git repository is not on the workspace server, you must clone it.

You can access the files in the cloned repository by navigating to the directory in the Server Files and Folders section of the navigation pane. You can make changes to the files in the working directory and then commit those changes to your local repository.

*Note:* Changes that you make to files in your local repository are not visible to other users until you push those changes to the remote repository.

To clone a Git repository:

1. In the navigation pane, click **Git Repository**, and then click **Clone a Repository**. The Cloning a New Repository window appears.

   *Note:* If you already have an existing Git repository, you can access the option to clone a repository in two ways:

   - Expand the repository drop-down list in the **Git Repository** section of the navigation pane and select **Clone a Repository**.
• Click and select **Preferences**. Click **Git Repositories**, and then click ➔ to open the Clone a New Repository window.

1. **Server Files and Folders**
2. **Tasks and Utilities**
3. **Snippets**
4. **Libraries**
5. **File Shortcuts**
6. **Git Repository**

   No repositories were found.

   **Clone a new Git repository.**

   ![Clone a Repository button](button.png)

   **Open an existing Git repository.**

   ![Open a Local Repository button](button.png)

   **Note:** You can also open an existing local repository by clicking **Open a Local Repository** and selecting a Git repository node in the Select a Folder window. The local repository must be associated with a remote repository in order to open it in SAS Studio.

2. In the **Remote repository** box, enter the remote repository that you want to clone. You can clone a repository in three ways:
   • Use SSH keys to authenticate with your repository hosting service - enter a repository name in a format similar to one of these examples:
     
     ```
     git@github.com:folder/sasgit.git
     
     or
     
     git@github.com:jquery/jquery-ui.git
     ```
   
   **Note:** By default, SAS Studio uses SSH keys to authenticate with your repository hosting service.
   
   • Use HTTPS authentication - enter a repository name in a format similar to this example:
     
     ```
     https://github.com/example/repo.git
     ```
   
   • Use a local repository - enter the full pathname of the local repository on your server file system.
Note: When you clone a local repository, you can pull files from the repository, but you cannot push any changes.

3. In the Local repository box, enter the path name of a folder on your workspace server for the cloned repository.

Note: You must select an empty folder for the cloned repository. If there are any files in the folder, the cloning operation cannot be completed.

4. From the Profile drop-down list, select the profile that you want to use to connect to the Git repository.

5. Click OK to clone the repository. You can view the files in the cloned repository by using the Server Files and Folders section of the navigation pane.

Committed Changes to Your Local Repository

Before you can push one or more changes from your local Git repository to the remote repository, you must first commit the changes from your working directory to your local repository. When you commit files, you stage the files that you want to include in the commit by moving the files from the Unstaged Files area to the Staged Files area. Next, you must enter a comment describing the commit and then commit the files. When you select files to commit, it is a good practice to group changes that are related and represent a state in your project development that you want to capture.

Note: When you commit a change to the local repository, those changes are still not viewable by other users, but are available for you to push to the remote repository.

To commit changes to your local repository:

1. After you have saved your changes to the files in your working directory, click Git Repository in the navigation pane. Click the Commit tab.

2. If your changed files are not listed in the Unstaged Files area, click < to refresh the list.

Files in the staging area are identified by a status icon that indicates the type of change:

- a new file to be added to the local repository
- a renamed file
- an existing file with changes
- a deleted file
• 文件内容与本地仓库中的另一个文件非常相似或完全相同
• 文件合并冲突

### Server Files and Folders
- Tasks and Utilities
- Snippets
- Libraries
- File Shortcuts

#### Git Repository

<table>
<thead>
<tr>
<th>Local</th>
<th>Commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unstaged Files (2)</td>
<td>🔄 🔄 🔄</td>
</tr>
</tbody>
</table>
  - `cars.sas`
  - `MyNewSASProgram.sas`
| Staged Files (0) | 🔄 🔄 |

**Note:** 自动保存的文件，其名称中包含~都不会包含在待提交文件列表中。

3. 选择您想提交的一个或多个更改，并点击Commit。当您选择要提交的文件时，将更改相关联并归类到一个单独的提交中是一个好习惯。

**TIP** 您也可以通过双击文件来选择和取消选择。

**Note:** 您不进行选择的更改不会包含在提交中。您的更改将留在工作目录中和Unstaged Files区域，直到您提交它们。
In the comments box, enter a description of the changes in the files that you are committing, and click **Commit Staged**. The changed files are now available to be pushed to the remote repository.

**Note:** To stage all changed files at one time, click 🔄.

**T I P** You can view the differences between the versions of a file in the staging area and your local repository by right-clicking the file and selecting **Show differences**. To discard the changes to a file in the staging area and reset the file to the version in your local repository, right-click the file and select **Reset file changes**. These options are available only if you have installed the SAS Studio 3.8 hot fix. You can download the hot fix from [SAS 9.4 Hot Fix Downloads](https://support.sas.com/downloads/).

4. In the comments box, enter a description of the changes in the files that you are committing, and click **Commit Staged**. The changed files are now available to be pushed to the remote repository.
Viewing the Commit History

You can use the commit history to view the history of all the changes to a repository. You can also view the details of a specific commit as well as the differences between two commits.

To view the history of a repository, click Git Repository in the navigation pane and ensure that the Local tab is selected. From the drop-down list, select the repository that you want to view and click History. A list of all the commits made in that repository is displayed in the work area in reverse chronological order. For each commit, you can view the commit message, the commit ID, the author, and the date and time of the commit.

To view the details of a specific commit, select the commit from the commit history and ensure that the Details tab is selected in the lower section of the work area. You can view the details of the parent of the selected commit by clicking the parent link in the details.

To view the differences between changes in the selected commit and the parent commits, click the Files tab in the lower section of the work area. You can also compare two adjacent commits from the commit history list by selecting the commits that you want to compare.

Note: To view the file differences in a separate window, right-click the differences section of the work area and select Open in new window.
Pulling and Pushing Files

The pull feature enables you to download the files from the remote repository and update your local repository to make sure you have the latest content. If you have made changes to some of the files in your local repository, Git merges the content from the remote repository with your local files. To pull files from the remote repository, click **Git Repository** in the navigation pane and ensure that the **Local** tab is selected. Then click **Pull**.

*Note:* SAS Studio displays a message if there is a conflict between files in the remote and local repositories when you pull the remote files. If a conflict occurs, the conflicted files are moved to the Unstaged Files area. After you resolve the conflicts, you can commit the files again. Files that are pulled from the remote repository and do not have any conflicts are staged for the commit.

After you have staged and committed your changed files from your working directory to your local repository, you can push them to the remote repository. When you push your committed changes to the remote repository, you are synchronizing files in your local repository with the remote repository. To push your files to the remote repository, click **Git Repository** in the navigation pane and ensure that the **Local** tab is selected. Then click **Push**. The commit history shows your committed changes merged into the remote repository.

Resetting Your Local Repository

You can undo commits that you have made by resetting your local repository to a prior commit.

To reset your local repository:

1. Make sure the commit history is open. Click **Git Repository** in the navigation pane and ensure that the **Local** tab is selected. From the drop-down list, select the repository that you want to view and click **History**.

2. Right-click the commit that you want to reset and select **Reset**. The Reset Local Repository window appears.

3. Select the type of reset that you want to perform:

   - **Soft** – resets the current branch to the prior commit. No changes are made to the staged changes (the index) or your working directory.
   - **Mixed** – resets the current branch to the prior commit and resets your staged changes (the index). No changes are made to your working directory.
   - **Hard** – resets the current branch to the prior commit, resets your staged changes (the index), and discards all changes in your working directory. You cannot undo a hard reset.

4. Click **OK** to reset your repository.
Your work in Git is managed in branches. By default, the main branch in your remote repository is identified as the origin/master branch. The main branch in your local repository is identified, by default, as the master branch. You can create additional branches and merge branches. When you push your committed changes to the remote repository, you are synchronizing files in a branch in your local repository with the branch in the remote repository. You can use the commit history feature to see the commits that are associated with a branch and whether branches are synchronized. For more information, see “Viewing the Commit History” on page 108.

In this example, the master branch is ahead of the origin/master commit by one commit. This indicates that you have made changes to the master branch that are not included in the origin/master branch because the "performance improvements" commit has not yet been pushed to the remote repository. For more information, see “Pulling and Pushing Files” on page 109.

After the "performance improvements" commit has been pushed to the remote repository, the origin/master and master branches are synchronized and are listed on the same line in the commit history.
For more information about managing branches in Git, see the documentation for your repository hosting service.

Creating a Branch

You can create a branch in Git in order to separate some of your work from other work that you are doing. When you create a branch, the commits in that branch are separated from your other commits until you merge the branch with another branch.

Note: When you have multiple branches, you can use the branches drop-down list at the top of your commit history to select the branch that you want to check out and work in.

To create a branch:

1. Make sure the commit history is open. Click Git Repository in the navigation pane and ensure that the Local tab is selected. From the drop-down list, select the repository that you want to view and click History.
2. Right-click the commit with which you want to start a new branch and select Create new branch. The Create branch window appears.
3. In the Branch name box, enter a name for your new branch. The branch name can contain only letters and numbers.
4. If you want to create a branch but remain on the current branch, clear the Checkout after create check box. This option is selected by default.
5. If you want to discard any changes in your working directory that you have not committed when the branch is created, select the Force check box. This option is not selected by default.
6. Click Create to create the branch. Your local repository now contains the files from the commit that you selected for creating the branch.
Merging Branches

When you have completed work in a branch, you can merge the work in that branch into another branch.

To merge a branch:

1. Make sure the commit history is open. Click **Git Repository** in the navigation pane and ensure that the **Local** tab is selected. From the drop-down list, select the repository that you want to view and click **History**.

2. In the branch drop-down list, select the branch into which you want to merge the new branch. For example, to merge `newbranch` with the master branch, you would need to check out the master branch. If necessary, in the confirmation window, click **Reload** to make sure that all of your changes are saved.

3. Right-click the latest commit in that branch and select **Merge into current branch** > `new-branch-name`. If necessary, in the confirmation window, click **Reload** to make sure that all of your changes are saved. The current branch in the commit history list is updated with the latest commit from the new branch.

   Note: If you are merging a branch into a branch that is not checked out, right-click the latest commit in the current branch, and select **Merge with branch** > `branch-name`.

   Note: SAS Studio displays a message if there is a conflict between files when you merge the branches. If a conflict occurs, the conflicted files are moved to the Unstaged Files area in the **Git Repository** section of the navigation pane. After you resolve the conflicts, you can complete the merge by staging and committing the files.

Deleting a Repository

You can delete the local repository on your workspace server. You can choose whether you want to delete all the files in the working directory when you delete the repository.

To delete the local repository:
1. Click and select **Preferences**. Click **Git Repositories**, and then click beside the repository that you want to delete. The Delete Local Repository window appears.

2. Select the type of delete that you want to perform:
   - **Soft** – deletes only the Git repository definition.
   - **Medium** – deletes the Git repository definition and removes any Git configuration information from the local repository directory. The files in your working directory are not deleted.
   - **Hard** – deletes the Git repository definition, the local repository directory, and all files in the local working directory.

   *Note:* You can also perform a hard delete of a repository from the **Server Files and Folders** section of the navigation pane. Right-click the working directory and select **Delete**.

3. Click **OK** to delete the repository.

---

### Sample Git Workflow Scenario

Suppose you need to help out on a project that stores its SAS programs in a Git repository. You need to make some minor changes to one program and then add a new program to the project.

In this sample Git scenario, you clone a repository to your workspace server, commit and push some changes, and create a branch. Then you merge the new branch with the master branch and push the changes.

*Note:* To use the Git integration feature in SAS Studio, you must first access the **Git Repository** section of the navigation pane. Click on the main toolbar and select **View > Git Repository**.

1. Create a profile so that you can access Git repositories in SAS Studio.
   a. Click **Git Repository** in the navigation pane. Click **Create a Profile** to open the Add a Profile window.
   b. Enter the credentials necessary to access your Git repository and click **OK**.
2. Clone your team’s repository to your workspace server so that you can access the files in the project.

   a. In the Git Repository section of the navigation pane, click Clone a Repository.

   b. In the Clone a New Repository window, specify the remote repository and the location on your workspace server that you want to use.

   c. Click OK to create the repository. The repository is added to the Git Repository section of the navigation pane.
3. Open the program that you need to update from your local repository, make your changes, and save the program.
   a. Click **Server Files and Folders** in the navigation pane and navigate to your working directory.
   b. Double-click the SAS program that you want to edit and make your changes. Click to save your changes.

4. Stage your changed files, enter a commit message, and commit your files to your local repository.
   a. Click **Git Repository** in the navigation pane, and then click the **Commit** tab. If your changed files are not listed in the Unstaged Files area, click to refresh the list.
   b. Select the updated file and click to move the file to the Staged Files area.
c. Enter your commit comments in the commit box and click Commit Staged.
Your changed file is committed to the local repository.

d. View your commit history by clicking **Local** and then clicking **History**. Notice that your commit is listed at the top of the commit history list. Your commit currently exists only in the master branch of your local repository. The master branch of your local repository is identified by **master**.

The second commit in the list represents the state of the master branch on the remote repository. The master branch of the remote repository is identified by **origin/master**.

When you push your changes to the remote repository, your commit is then part of the origin/master branch on the remote repository.
5. Pull the files from the remote repository and then push your committed files.

   Note: It is good practice to pull the files from the remote repository to make sure you do not have any conflicts.

   a. In the Git Repository section of the navigation pane, click the Local tab, and then click Pull. Your local repository is updated with the latest content from the remote repository. Any changes in the remote repository are merged with your local files. If there is a conflict between files in the remote and local repositories, SAS Studio displays a message so that you can resolve the conflict before you proceed.

   b. Click Push to push your changed file to the remote repository. Your commit history is updated. Notice that your commit is now part of the remote origin/master branch.
6. Create a branch to keep your work separate from the master branch. Add a new program, test it, and commit your changes to that branch.

   a. In the commit history list, right-click your commit and select Create new branch. The Create branch window appears.

   b. Enter a name for the branch and click Create. Notice that the Checkout after create option is selected so that the new branch is checked out and is the active branch that you are working in.

   ![Create branch window](image)

   In the commit history list, notice that the new classdata branch, the remote origin/master branch, and your local repository master branch are all synchronized and displayed on the same line.

   ![Commit history list](image)

   c. Use the Server Files and Folders section of the navigation pane to create a program and save it in your working directory.

   d. Stage the new program and commit it to the new classdata branch in your local repository. Notice in the commit history list that your commit in the classdata branch is not synchronized with the remote origin/master branch or the local master branch.
7. When your new program is complete, merge the new branch back into the master branch and perform a final pull from and push to the remote repository.

a. When you are ready to merge the `classdata` branch into the master branch, you must first check out the master branch. Select the master branch from the `Current branch` drop-down list. In the confirmation window, click `Reload` to make sure that all of your changes are saved.

In the commit history list, right-click the master branch and select **Merge with branch > classdata**. In the confirmation window, click `Reload` to make sure that all of your changes are saved.
Notice in the commit history list that the master branch has been updated with the new file from the `classdata` branch.

b. In the Git Repository section of the navigation pane, click the Local tab, and then click Pull.

c. Click Push to push your changes to the remote repository. Your commit history is updated. Notice that the local master branch, which was merged with the `classdata` branch, and the remote origin/master branch are now synchronized.
Chapter 7 • Understanding Git Integration in SAS Studio

Message
- origin/master
- master
- classdata
- Adding class analysis program

Details
Merge 'classdata' into branch master

Author: myusername
Date: 15 minutes ago (Oct 24, 2018, 3:34:34 PM)
Commit: b60783f1da1476b039b0b4c51f4f1f681d0cfc3c6f65
Parent(s): e84c4e512d4e8792490c236d294d1b3af5202ec14f0f19aa593d407284d818a19eb5a27fa5ecc37
Chapter 8
Understanding Tasks in SAS Studio

What Is a Task?

A task is an XML and Apache Velocity code file that generates SAS code and formats results for you. Tasks include SAS procedures from simple data listings to complex analytical procedures. SAS Studio is shipped with several predefined tasks. The tasks are organized into categories.

*Note:* Some categories and their tasks might not be available at your site because you do not have the required SAS product. For example, if you do not have SAS/ETS installed, then the Econometrics category is not available.

This table lists the task categories and their product dependencies.

<table>
<thead>
<tr>
<th>Task Category in SAS Studio</th>
<th>Required SAS Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Base SAS</td>
</tr>
<tr>
<td>Graph</td>
<td>Base SAS</td>
</tr>
<tr>
<td>Map</td>
<td>Base SAS</td>
</tr>
<tr>
<td>Statistics</td>
<td>SAS/STAT</td>
</tr>
<tr>
<td>Linear Models</td>
<td>SAS/STAT</td>
</tr>
<tr>
<td>Survival Analysis</td>
<td>SAS/STAT</td>
</tr>
<tr>
<td>Task Category in SAS Studio</td>
<td>Required SAS Product</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Multivariate Analysis</td>
<td>SAS/STAT</td>
</tr>
<tr>
<td>Cluster Analysis</td>
<td>SAS/STAT</td>
</tr>
<tr>
<td>Power and Sample Size</td>
<td>SAS/STAT</td>
</tr>
<tr>
<td>Statistical Process Control</td>
<td>SAS/QC</td>
</tr>
<tr>
<td>Combinatorics and Probability</td>
<td>Base SAS</td>
</tr>
<tr>
<td>Econometrics</td>
<td>SAS/ETS</td>
</tr>
<tr>
<td>Forecasting</td>
<td>SAS/ETS</td>
</tr>
<tr>
<td>Network Optimization</td>
<td>SAS/OR</td>
</tr>
<tr>
<td>Combinatorics and Probability</td>
<td>Base SAS</td>
</tr>
<tr>
<td>SAS Viya Prepare and Explore</td>
<td>SAS 9.4M6, SAS Visual Statistics</td>
</tr>
<tr>
<td>SAS Viya Unsupervised Learning</td>
<td>SAS 9.4M6, SAS Visual Data Mining and Machine Learning</td>
</tr>
<tr>
<td>SAS Viya Evaluate and Implement</td>
<td>SAS 9.4M6, SAS Visual Data Mining and Machine Learning, SAS Optimization</td>
</tr>
<tr>
<td>SAS Viya Network Analysis and Optimization</td>
<td>SAS/OR, SAS Optimization</td>
</tr>
<tr>
<td>SAS Viya Econometrics</td>
<td>SAS 9.4M6, SAS Econometrics Procedures</td>
</tr>
</tbody>
</table>

Note: Included in your SAS 9.4M6 order are procedures that enable you to take advantage of additional functionality that is provided by SAS Viya. You must have a SAS Viya license to run the SAS Studio tasks for these products:
- SAS Econometrics Procedures
- SAS Visual Data Mining and Machine Learning Procedures
- SAS Visual Statistics Procedures

For more information about the SAS Studio tasks, see *SAS Studio: Task Reference Guide*.

You can edit a copy of these predefined tasks in order to customize the tasks for your site. You can also build your own tasks.
How to Run a Task

To run a predefined task:

1. In the navigation pane, click the **Tasks and Utilities** section.
2. Expand the folder that contains the task.
3. Right-click the task name and select **Open**. Alternatively, you can double-click the task to open it.

   The task opens to the right of the work area.

4. If the **Data** tab is available, specify an input data source and select columns for the roles in the data source. A role is a description of a variable’s purpose in the task. To add a column to a role, click **+**. A list of available columns for that role appears. If only one column can be assigned to the role, you select a column and the list disappears. If multiple columns can be assigned, you can press Ctrl or Shift to select multiple columns from the list and click **OK**.

5. On the remaining tabs, specify any other required options, which are denoted with a red asterisk. As you assign values to the task, the relevant SAS code is generated. For more information about the options available for each task, see *SAS Studio: Task Reference Guide*.

6. To run the task, click **Run**.
If the task generates output data, the table opens in the **Output Data** tab.

If the task generates results, the output appears on the **Results** tab.
Save a Task and Its Option Settings

If you use a task frequently, you might want to save the task after you specify the input data source and the option settings. In SAS Studio, you can save a task as a CTK file on the server or in your My Tasks folder. The next time you need to run the task, double-click the task in the navigation pane, and the task appears with all of your previous settings.

Note: Before you can save a task, you must specify an input data set and all the options that are required to run the task.

To save a task:
1. Click . The Save As window appears.
2. Select the location where you want to save the task file. You can save this file in the Server Files and Folders section or in your My Tasks folder. Specify a name for this file. For the file type, select CTK Files (*.CTK). Click Save.

Note: When you save a task as a CTK file, the task is no longer attached to a corresponding task in the Tasks and Utilities section. For example, if you run the Bar Chart task that is available in the Tasks and Utilities section, no changes are made to the Bar Chart.ctk file in the Server Files and Folders section.

Edit a Predefined Task

To customize the predefined tasks for your site, you can edit the XML code that is used to create the task.

To edit a predefined task:
1. In the navigation pane, open the Tasks and Utilities section.
2. Expand the folder that contains the task.
3. Right-click the name of the task that you want to edit and select Add to My Tasks. The Add to My Tasks window appears.
4. Specify a name and description for the task. By default, the name and description from the predefined task is used. Using the Category drop-down list, you can also specify where to save a copy of this task in the My Tasks folder. If you select (none), the task is added directly to the My Tasks folder. Click Add.
5. Open the My Tasks folder and select the copied task.
6. Click . The XML file for the task appears.
7. Edit the XML file and save your changes. To preview your changes, click .

Edit a Predefined Task

To customize the predefined tasks for your site, you can edit the XML code that is used to create the task.

To edit a predefined task:
1. In the navigation pane, open the Tasks and Utilities section.
2. Expand the folder that contains the task.
3. Right-click the name of the task that you want to edit and select Add to My Tasks. The Add to My Tasks window appears.
4. Specify a name and description for the task. By default, the name and description from the predefined task is used. Using the Category drop-down list, you can also specify where to save a copy of this task in the My Tasks folder. If you select (none), the task is added directly to the My Tasks folder. Click Add.
5. Open the My Tasks folder and select the copied task.
6. Click . The XML file for the task appears.
7. Edit the XML file and save your changes. To preview your changes, click .
Create a New Task

SAS Studio provides a template that you can use to create custom tasks for your site. For help with writing your first custom task, see *SAS Studio: Writing Your First Custom Task*.

To create a custom task:

1. In the navigation pane, open the Tasks and Utilities section.

2. Click 🔄 and select **New Task**. A blank task template opens.

3. Edit the code in the task template to create your task. To view the user interface for the task template, click 📸. In the user interface for the task template, you can see examples of radio buttons, check boxes, combination boxes, and other types of options. For more information about this file, see *SAS Studio: Developer’s Guide to Writing Custom Tasks*.

4. Click ✅.

*Note:* The name of the task cannot include these special characters:

? * * | : ^
Customizing the Task Code and the Task Layout in the Workspace

The Preferences window enables you to change several options that affect what and how the task code is displayed.

To access these options, click ⬇️ and select Preferences. Click Tasks.

For more information, see “Setting Task Preferences” on page 139.
Appendix 1

Customizing SAS Studio

About Setting Your Preferences

The Preferences window enables you to customize several options in SAS Studio.

To change your preferences, click ⓢ and select Preferences.

Setting General Preferences

From the General page, you can set these options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include a Show Details button in error messages</td>
<td>Adds a Show Details button to any error messages that SAS Studio generates.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Size grid columns to content</strong></td>
<td>Automatically adjusts the width of the columns in the table viewer to fit the size of the column content.</td>
</tr>
<tr>
<td><strong>Start new programs in interactive mode</strong></td>
<td>Opens new programs with the interactive mode on. This option is available only if you are running the first maintenance release for SAS 9.4 or later. For more information, see “Working in Interactive Mode” on page 28.</td>
</tr>
<tr>
<td><strong>Automatically refresh libraries after each submission</strong></td>
<td>Refreshes the Libraries section of the navigation pane after you submit a program, task, or query so that any changes to the libraries are displayed immediately. However, selecting this option can cause you to lose your place in the list of libraries and affect performance. This option is selected by default.</td>
</tr>
<tr>
<td><strong>Automatically refresh files and folders after each submission</strong></td>
<td>Refreshes the Server Files and Folders section of the navigation pane after you submit a program, task, or query so that any changes to the files and folders are displayed immediately. However, selecting this option can cause you to lose your place in the list of files and folders and affect performance. This option is selected by default.</td>
</tr>
<tr>
<td><strong>Default text encoding</strong></td>
<td>Specifies the character-set encoding that is used when text files are read or written. The default value is UTF-8. This option is not available for SAS servers running on z/OS. For a list of some of the encoding options and the languages that they are associated with, see Appendix 3, “Text Encoding Options and Language Mappings,” on page 157. Note: You can also specify the text encoding when you open a single file by right-clicking the file and selecting Open with text encoding. The Choose Text Encoding window appears and you can select the encoding. This option applies to SAS program files (*.SAS).</td>
</tr>
<tr>
<td><strong>Display a message on arrival</strong></td>
<td>Displays a message generated by SAS Studio, such as when background jobs are started and when they are complete or when your SAS Studio session is reset. SAS Studio application errors or warnings are displayed as messages, but SAS program errors and warnings continue to be displayed in SAS log files. You can adjust the number of seconds that the message is displayed. The range of time that a message can be displayed is between 3 seconds and 30 seconds. The default value is 5 seconds.</td>
</tr>
</tbody>
</table>
### Setting the Start Up Preferences

From the Start Up page, you can specify these options for starting SAS Studio.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capture all log and debug events</td>
<td>Records all diagnostic messages each time you run a program, task, or query. You can view the diagnostic messages by clicking <strong>Messages</strong> in the lower right corner of your SAS Studio browser window. In the Messages window, click the <strong>Filter by</strong> drop-down list and select <strong>Debug</strong>. The debug messages are available in the Messages window only when you have selected this option.</td>
</tr>
</tbody>
</table>

### Setting the Code and Log Preferences

From the **Code and Log** page, you can specify these options for the code and log tabs.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable autocomplete</td>
<td>Turns on the autocomplete feature of the code editor. This feature can predict the next keyword that you want to enter before you actually type it completely. For more information, see “Using the Autocomplete Feature” on page 19.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Enable hint</strong></td>
<td>Displays the syntax help window when you position the mouse pointer over a valid SAS keyword in your program. If this option is not selected, then you can view the syntax help by right-clicking a keyword and selecting <strong>Syntax Help</strong>. This option is not selected by default.</td>
</tr>
<tr>
<td><strong>Tab width</strong></td>
<td>Displays the number of spaces that are inserted into your text when you insert a tab character. The default value is four spaces for each tab character.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>In Microsoft Internet Explorer and Apple Safari, spaces are used instead of Tab characters. If you are using those browsers, you must select the <strong>Substitute spaces for tabs</strong> check box in order for the value of the tab width to be used.</td>
</tr>
<tr>
<td><strong>Substitute spaces for tabs</strong></td>
<td>Inserts the number of spaces listed in the <strong>Tab width</strong> box instead of a single tab character. This option applies to both text that you enter in the code editor and text that you paste into the code editor.</td>
</tr>
<tr>
<td><strong>Enable color coding</strong></td>
<td>Displays the text in the code editor in different colors to help you identify different elements in the syntax.</td>
</tr>
<tr>
<td><strong>Show line numbers</strong></td>
<td>Displays line numbers in the leftmost column of the program and log windows.</td>
</tr>
<tr>
<td><strong>Font size</strong></td>
<td>Specifies the font size of the text in the code editor and log window.</td>
</tr>
</tbody>
</table>
Enable autosave

 Automatically creates auto-saved copies of each previously saved program file that you are editing so that you can recover the files if you lose your network connection or your browser closes unexpectedly.

*Note:* New program files are not auto-saved until you save them first.

The files are saved at the interval specified in the **Autosave Interval** option. The default interval is 30 seconds.

*Note:*

- The auto-saved file is deleted if you sign out of SAS Studio and click *Yes* when you are prompted about losing unsaved changes.

- If you lose your network connection or your browser closes unexpectedly and you have selected the **Open a new program/process flow tab** option in the Start Up preferences, the auto-saved file is saved as *filename.sas~*. You can open the auto-saved file in the code editor and save it.

- If you lose your network connection or your browser closes unexpectedly and you have selected the **Continue where you left off** option in the Start Up preferences, you are prompted to open the auto-saved file when you sign back on to SAS Studio.

For more information, see “Setting the Start Up Preferences” on page 133.

<table>
<thead>
<tr>
<th>Show generated code in the SAS log</th>
<th>Displays the ODS statements, %LET statements, and any other code that is automatically generated by SAS in the log file. This option applies to both SAS tasks and SAS program files.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream log updates while a procedure is running</td>
<td>Displays log updates as a procedure is processed. If you do not select this option, the log is displayed when the procedure has finished running. This option is selected by default and might have a slight effect on performance.</td>
</tr>
<tr>
<td>Automatically clear log</td>
<td>Clears the log each time you submit code. This option is selected by default. If you do not select this option, then you can click on the Program Editor toolbar to clear the log manually.</td>
</tr>
<tr>
<td>Append log</td>
<td>Appends new log information to the existing log. You can choose to append the logs for programs, tasks, or both programs and tasks.</td>
</tr>
</tbody>
</table>
# Setting the Results Preferences

From the **Results** page, you can specify these options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HTML output style</strong></td>
<td>Displays the style that is applied to results in HTML. To change the style that is applied to the results, select another style from the drop-down list.</td>
</tr>
<tr>
<td><strong>Generate HTML graphs as SVG</strong></td>
<td>Creates SVG graphs instead of PNG graphs in HTML output. SVG graphs maintain clarity when you zoom in and out.</td>
</tr>
<tr>
<td><strong>Display warning if results are larger than ( n ) MB</strong></td>
<td>Displays a warning message when you attempt to open a results file that is larger than ( n ) megabytes (MB). The default value is 3 MB.</td>
</tr>
<tr>
<td><strong>Enable accessible graph option</strong></td>
<td>Adds accessibility metadata to graphs that are created by ODS Graphics. Users with disabilities will access the accessibility metadata using SAS Graphics Accelerator. This option is available only if you are running SAS 9.4M4 or later. For more information, see the ODS HTML5 Statement in SAS Output Delivery System: User’s Guide. Note: The SAS code that is associated with this option is displayed in the log file only if you have selected the Show generated code in the SAS log option in the General preferences.</td>
</tr>
<tr>
<td><strong>Produce PDF output</strong></td>
<td>Generates results in PDF format. This option is selected by default.</td>
</tr>
<tr>
<td><strong>PDF output style</strong></td>
<td>Displays the style that is applied to results in PDF. To change the style that is applied to the results, select another style from the drop-down list.</td>
</tr>
<tr>
<td><strong>Generate the default table of contents</strong></td>
<td>Creates a table of contents in the PDF file.</td>
</tr>
</tbody>
</table>
### Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enable accessible PDF option</strong></td>
<td>Adds accessibility metadata to the PDF file that enables the file to be accessed by assistive technology such as a screen reader. When metadata is added, the file is often called a &quot;tagged PDF&quot; and follows the PDF/Universal Accessibility (PDF/UA) format. This option is available only if you are running SAS 9.4M4 or later. &lt;br&gt;&lt;br&gt;<strong>Note:</strong> The SAS code that is associated with this option is displayed in the log file only if you have selected the <strong>Show generated code in the SAS log</strong> option in the General preferences.</td>
</tr>
<tr>
<td><strong>Produce RTF output</strong></td>
<td>Generates results in RTF format. This option is selected by default.</td>
</tr>
<tr>
<td><strong>RTF output style</strong></td>
<td>Displays the style that is applied to results in RTF. To change the style that is applied to the results, select another style from the drop-down list.</td>
</tr>
<tr>
<td><strong>Automatically open generated output data</strong></td>
<td>Displays the output data that is created when you run a task or submit code. By default, only the first 30 columns are displayed.</td>
</tr>
</tbody>
</table>

**Note:** If you want to use a custom style, you must customize the SAS Studio output environment. For more information, see Appendix 4, “Customized Output Environment,” on page 159.

### Setting Table Preferences

From the Tables page, you can set the options for viewing tables in the SAS Studio workspace.

<table>
<thead>
<tr>
<th>Rows per page</th>
<th>Specifies the number of rows to display per page of data. The default value is <strong>100</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Columns displayed</strong></td>
<td>Specifies the maximum number of columns to be displayed in the table viewer. The default value is <strong>30</strong>. Increasing the number of columns that are displayed might affect your performance.</td>
</tr>
</tbody>
</table>

---

**Setting Table Preferences**

From the Tables page, you can set the options for viewing tables in the SAS Studio workspace.

<table>
<thead>
<tr>
<th>Rows per page</th>
<th>Specifies the number of rows to display per page of data. The default value is <strong>100</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Columns displayed</strong></td>
<td>Specifies the maximum number of columns to be displayed in the table viewer. The default value is <strong>30</strong>. Increasing the number of columns that are displayed might affect your performance.</td>
</tr>
</tbody>
</table>

---

**Setting Table Preferences**

From the Tables page, you can set the options for viewing tables in the SAS Studio workspace.

<table>
<thead>
<tr>
<th>Rows per page</th>
<th>Specifies the number of rows to display per page of data. The default value is <strong>100</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Columns displayed</strong></td>
<td>Specifies the maximum number of columns to be displayed in the table viewer. The default value is <strong>30</strong>. Increasing the number of columns that are displayed might affect your performance.</td>
</tr>
</tbody>
</table>
### SAS variable name policy

Enables you to specify one of the following sets of rules to apply to SAS variable names.

- **ANY** specifies that the variable names can begin with or contain any characters, including blanks, must contain at least one character, and cannot contain any null bytes. Variable names can contain mixed-case letters as well as special and multi-byte characters. Names can be up to 32 bytes in length. This option is selected by default. Leading blanks are preserved, but trailing blanks are ignored.

- **V7** specifies that the variable names must begin with a letter of the Latin alphabet (A–Z, a–z) or the underscore character. They cannot contain blanks or special characters except for the underscore and cannot be assigned the names of special SAS automatic variables or variable list names. Variable names can contain mixed-case letters and can be up to 32 bytes in length.

- **UPCASE** specifies that the variable name follows the same rules as V7, except that the variable name is uppercase, as in earlier versions of SAS.

### SAS member name policy

Enables you to specify one of the following sets of rules to apply to SAS data set names, SAS data view names, and item store names. Any changes you make to this option take effect when you start SAS Studio or reset your SAS session.

- **EXTEND** (Extended member names) specifies that names must contain at least one character (letters, numbers, valid special characters, and national characters), cannot begin with a blank or a period, and cannot include a null byte. Names can contain mixed-case letters and can be up to 32 bytes in length. Leading blanks are preserved, but trailing blanks are ignored. Special characters cannot include the / \ * ? " < > : - characters. This option is selected by default.

- **COMPATIBLE** (Basic member names) specifies that the names must begin with a letter of the Latin alphabet (A–Z, a–z) or the underscore character and cannot contain blanks or special characters except for the underscore. Names can contain mixed-case letters and can be up to 32 characters in length.
Setting Task Preferences

From the Tasks page, you can set the options for the generated SAS code and the task layout in the SAS Studio workspace.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trim all leading and trailing spaces in generated code</td>
<td>Removes any blank spaces that appear before or after the generated code.</td>
</tr>
<tr>
<td>Generate header comments for task code</td>
<td>Adds comments before the generated code for a SAS task.</td>
</tr>
<tr>
<td>Automatically format generated code</td>
<td>Automatically formats any code that is generated by a task and displayed in the code editor.</td>
</tr>
</tbody>
</table>

**View**

Specifies how to lay out the task options, task code, and task results in your workspace. You can choose from these options:

- **Split** displays the task settings, the code, and the results for the task.
- **Settings** displays only the options for the task in the workspace.
- **Code/Results** displays the SAS code, the log, and any results in the workspace.

**Show task code**

Specifies whether to display the SAS code for the task when you select the **Split** view or the **Code/Results** view.

**Show task log**

Specifies whether to display the log that is generated when you run the task. This option is available only if you select the **Split** view or the **Code/Results** view.

**Display task settings on right**

Displays the task option to the right of the SAS Studio workspace. By default, the task options are displayed to the left.

Accessing Repositories in SAS Studio

Administrators can create global repositories. These repositories are available to everyone at your site. Global repositories are automatically available from the Tasks and Utilities and Snippets sections when you first open SAS Studio.

If your administrator allows it, you can also set your preferences to accept additional repositories. For more information, see *SAS Studio: Administrator’s Guide*. 
## Setting Preferences for Background Job Submissions

From the **Background Jobs** page, you can set preferences for background job submissions.

<table>
<thead>
<tr>
<th>If the background job log or output file already exists</th>
<th>Specifies how to handle the background job submission if a log and output file already exist. You can choose from these options:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Delete existing log and output</strong> deletes the existing log and output files and replaces them with new files when you resubmit the program as a background job.</td>
</tr>
<tr>
<td></td>
<td><strong>Fail background job submission</strong> cancels the background job submission. You must delete the existing log and output files and resubmit the program.</td>
</tr>
<tr>
<td></td>
<td><strong>Prompt for fail or overwrite</strong> displays a message window to confirm that you want to delete or rename the existing log and output files before submitting the background job. If you select No, the background job submission is canceled. This is the default value.</td>
</tr>
<tr>
<td></td>
<td><strong>Number log/output</strong> saves all log and output files by creating a unique filename for each file. The log and output files are saved as <code>program-name(userid-YYYY-MM-DD HHMMSS)</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output and log destination</th>
<th>Specifies where to save the output and log files. You can choose from these options:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Same folder as .sas file</strong> saves the log and output files to the same folder as the .sas code file.</td>
</tr>
<tr>
<td></td>
<td><strong>Use folder</strong> enables you to specify a location in which to save the log and output files. Click <strong>Browse</strong> to search for a folder.</td>
</tr>
<tr>
<td></td>
<td><strong>Prompt for output and log file names</strong> prompts you to specify a location in which to save the log and output files. By default, the log and output are saved in the same location as the .sas code file. Click <strong>Browse</strong> to search for a folder and specify a filename.</td>
</tr>
</tbody>
</table>

You can change the length of time that the background job notification messages are displayed by using the **Display a message on arrival** option on the **General** page of the...
Preferences window. For more information, see “Setting General Preferences” on page 131.

Setting Preferences for Git Profiles

From the Git Profiles page, you can add, view, and delete Git profiles. For more information, see “Creating a Git Profile” on page 101.

Setting Preferences for Git Repositories

From the Git Repositories page, you can add, view, and delete Git repositories. For more information, see “Cloning a Repository” on page 103.
Opening SAS Enterprise Guide Projects in SAS Studio

Note: This functionality is available only if your site administrator has set the webdms.allowEGPOpen property to true. For more information, see “Configuration Properties for SAS Studio” in SAS Studio: Administrator’s Guide.

When you open a SAS Enterprise Guide project in SAS Studio, the process flows in the project are extracted and converted to process flows in SAS Studio. Any elements from the SAS Enterprise Guide project that are not supported by SAS Studio are either converted to different node types or are omitted from the process flow. SAS Studio creates a conversion report that lists the status of each node in the SAS Enterprise Guide process flow.
Appendix 2 • Converting SAS Enterprise Guide Projects to SAS Studio Process Flows

Here is a project in SAS Enterprise Guide.

Here is the converted process flow in SAS Studio.
Your Server Environments Must Match

In SAS Enterprise Guide, a profile defines the connection between SAS Enterprise Guide and a SAS Metadata Server. The SAS Metadata Server contains metadata definitions for objects such as workspace servers, libraries, and users. In SAS Enterprise Guide, only one profile can be active at a time. (If you do not select a profile, you can access only the SAS server installed on your machine. You cannot define libraries or share SAS resources with other SAS applications.)

If the active profile for the current SAS Enterprise Guide session does not match the profile for the project file that you are trying to open, you are prompted to change profiles. Because SAS Enterprise Guide is a desktop application, you can easily access both your local files as well as any SAS Workspace Servers that your SAS Enterprise Guide session is configured to use.

SAS Studio is a web interface that can have only one SAS server configuration at a time. You can access local files from SAS Studio only if you are using the Single-User edition.

- The Enterprise edition of SAS Studio can be configured to use multiple workspace servers that are defined in the SAS Metadata Server, but you can connect to only one workspace server at a time in SAS Studio.
- The Basic edition of SAS Studio can access only the single workspace server that it is configured to use and the file system available to that server.
- The Single-User edition of SAS Studio can access the local file system and the SAS installation on the desktop.

These differences are important to understand when you open a SAS Enterprise Guide project file in SAS Studio. If the environment for the SAS Enterprise Guide project and the environment for SAS Studio do not match, you cannot run the converted process flow in SAS Studio. In addition, you cannot access any nodes within the SAS Enterprise Guide project that are associated with an environment that is not accessible to SAS Studio. The conversion report provides warnings about the mismatches.

Open a SAS Enterprise Guide Project File

To open a SAS Enterprise Guide project in SAS Studio:


2. In the Server Files and Folders section, expand the appropriate folder and double-click the project that you want to open. You could also drag this project file to the SAS Studio workspace.

   Note: Your SAS Enterprise Guide project file must be in a location that can be accessed by SAS Studio. Before opening the project, you might need to move the file to this location.

3. Review any errors or warnings in the conversion report. Errors indicate that the process flow cannot execute as expected. Warnings indicate that the process flow might not execute as expected or might need some user modification to run.

4. To view the process flow in SAS Studio, click the Flow tab.
Understanding the Conversion Report

When you open the project file in SAS Studio, the conversion report is displayed. This report lists how each node of the SAS Enterprise Guide process flow was converted and notes any potential problems with the converted SAS Studio process flow.

The conversion report includes these sections:

<table>
<thead>
<tr>
<th>Section Heading</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>This section provides information about the SAS Enterprise Guide project from the project metadata. It might contain warnings about features that are not available in SAS Studio, such as the grid view and zoom features.</td>
</tr>
<tr>
<td>Environment</td>
<td>This section provides information about the SAS Enterprise Guide connection environment. The report displays an error message if the SAS server environment is not the same in SAS Studio.</td>
</tr>
<tr>
<td>Process Flow</td>
<td>This section provides a list of nodes that are included in the converted process flow and a list of nodes that are not included. The lists are in alphabetical order.</td>
</tr>
<tr>
<td>Node Summary</td>
<td></td>
</tr>
<tr>
<td>Conditions</td>
<td>This section provides a list of conditions from the SAS Enterprise Guide project. Conditional processing is not supported in SAS Studio.</td>
</tr>
</tbody>
</table>
**Supported Nodes**

Assuming that the connection environments are the same for the SAS Enterprise Guide project file and your SAS Studio session, the nodes in the SAS Enterprise Guide process flow should convert as follows:

**Program**

Program nodes are converted directly to SAS Program nodes in SAS Studio. If the contents of the program node are stored in a file, the file must be available in order for the contents of the file to be read into the converted program node.

**SAS Studio Tasks**

SAS Studio tasks in SAS Enterprise Guide are converted directly to SAS Studio tasks in SAS Studio. Any options that are specified in SAS Enterprise Guide are included in the converted task in SAS Studio. You can improve the conversion process by updating your project to use SAS Studio tasks where possible rather than SAS Enterprise Guide tasks. For more information, see “Accessing SAS Studio tasks” in the SAS Enterprise Guide Help.

**Data**

Data nodes in SAS Enterprise Guide are converted to program nodes in SAS Studio. SAS Studio does not currently support data nodes. The converted program node contains SAS code that includes the following information:

- Comments about the data location.
- SAS code that samples the contents of the data. The generated code depends on the type of data in the data node.
  - The first 50 lines of any directly readable text file, including CSV files, fixed-length text files, HTML files, and tab-delimited text files are displayed in the program output.
  - Data nodes that represent Microsoft Excel files do not sample the data. The nodes include only comments about the original Excel file.
  - SAS programs that represent SAS data sets create a view of the rows in the data set. You cannot edit the table or perform any other functions in this view.
SAS Enterprise Guide Tasks

The code that is generated by a SAS Enterprise Guide task is converted to a program node in SAS Studio. This code sometimes includes references to macro variables. For more information about the macro variables that are available in SAS Studio, see “Using Macro Variables” on page 26.

Queries

The code that is generated by a SAS Enterprise Guide query is converted to a program node in SAS Studio. This code sometimes includes a reference to the `_eg_conditional_dropds` macro. This macro is also available in SAS Studio.

Import Data

The Import Data node in SAS Enterprise Guide does not convert directly to an Import Data task node in SAS Studio.

When you import a file in SAS Enterprise Guide, three nodes are created in the SAS Enterprise Guide process flow:

- the input data
- Import Data
- the generated output SAS data set
Here is how these nodes appear in the SAS Studio process flow:

<table>
<thead>
<tr>
<th>SAS Enterprise Guide Node</th>
<th>SAS Studio Node</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>input data node</td>
<td>program node that samples the file contents</td>
<td>Data nodes that represent Excel files do not sample the data. The nodes include only comments about the original Excel file.</td>
</tr>
</tbody>
</table>
| Import Data node          | program node that contains the code that is used to create the output SAS data set | Some options in the SAS Enterprise Guide Import Data wizard are not supported in SAS Studio:  
  • SAS Studio does not support importing data from HTML files.  
  • The Use SAS/ACCESS Interface option is ignored. The code that is generated by SAS Studio always uses PROC IMPORT to read an Excel file and then a DATA step to format the data appropriately. All other data types are imported using a DATA step.  
  • SAS Studio does not remove characters that cause transmission errors from the automatically generated code. |
| output SAS data set       | program node that samples the output data set | None. |
Because the import functionality is different in SAS Enterprise Guide and SAS Studio, you might have to update your import code to make it work exactly the same in SAS Studio.

**Export File**

The Export File node in SAS Enterprise Guide does not convert directly to a node in SAS Studio.

When you export a file in SAS Enterprise Guide, three nodes are created in the SAS Enterprise Guide process flow:

- the input data
- Export File
- the output data
Here is how these nodes are converted in SAS Studio:

<table>
<thead>
<tr>
<th>SAS Enterprise Guide Node</th>
<th>SAS Studio Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>input data node</td>
<td>program node that displays the contents of the SAS data set</td>
</tr>
<tr>
<td>Export File node</td>
<td>program node that uses the metadata that is stored in the SAS Enterprise Guide project to generate code that can export the input data set</td>
</tr>
<tr>
<td>output data node</td>
<td>program node that samples the output data</td>
</tr>
</tbody>
</table>

*Note:* The **Use labels for column names** option in SAS Enterprise Guide is not supported in SAS Studio.
Unsupported Nodes

These node types are not currently supported in a SAS Studio process flow:

- Notes
  
  *Note:* The contents of any Note nodes are written to the Conversion Report in SAS Studio.

- Stored processes
- OLAP cubes
- Code (last submitted code)
- Logs
- Project logs
- ODS results
- SAS Web Report Studio (WRS) reports
- Send email as a step in a project
- Publish elements
SAS Enterprise Guide Project Items That Are Not Supported

These items in a SAS Enterprise Guide project are not currently supported in SAS Studio:

- Project log
- Decision Manager elements
- Data Exploration elements
- Scheduling
- Conditions
- Ordered lists
- Explore Data list
- Git source control

In addition, the following attributes of a SAS Enterprise Guide process flow are not currently supported in a SAS Studio process flow:

- background color
- process flow zoom

  Note: In SAS Studio, use the zoom functionality in your browser.

- grid view
- submit to grid

Prompts

SAS Studio process flows do not currently support prompts. When you convert a SAS Enterprise Guide process flow that contains prompts, code is added to the converted program or task nodes that use prompts. To view the added code, open the conversion report. The added code includes these elements:

- Comments about the values that are expected for the macro variables that are created by the prompt, such as types, minimums, and maximums.
• Macro variable definitions that the prompts would have created.
• Default or blank default values for the macro variables based on the prompt definition.
• Deletion of the macro variables at the end of the program unless the prompt definition includes the **Use prompt value throughout project** option. (You can select this check box in the Prompt Manager in SAS Enterprise Guide.)
In this example, the **Use prompt value throughout project** option was not selected, so the `%SYMDEL` statement deletes the macro variables.

If you want to run your prompt-dependent code against different prompt values, you must manually change the values in the generated macro variables. If a prompt is used for more than one node, the values of the macro variables must be manually changed in each node to simulate the prompt. Editing the values can be a lot of work if there are many nodes that depend on a single prompt.

If a SAS Enterprise Guide task node has a prompt defined, the converted SAS Studio task node includes the generated macro code. If you want to change the values in the converted task node, you must edit the task’s CTM file. If you want to change these prompt values with a user interface, you can create a SAS Studio task node that contains input controls for each of the macro variables that the prompt would have created. For more information and examples, see [SAS Studio Tasks Resembling SAS Enterprise Guide Prompts](#).

### Limitations

SAS Studio process flows contain a subset of all the functionality of a SAS Enterprise Guide project.

Before you convert a SAS Enterprise Guide project to a SAS Studio process flow, note these limitations:

- You can convert only SAS Enterprise Guide projects that are saved with SAS Enterprise Guide 7.1 or later.
• You cannot open password-protected SAS Enterprise Guide projects in SAS Studio. You must remove the password protection before you convert the file.

• The ActiveX graphics output format in SAS Enterprise Guide does not look the same in SAS Studio. SAS Studio does not support ActiveX for ODS output.

• When you open a process flow for the first time in SAS Studio, the nodes are arranged using the same functionality of the **Arrange** button.

• If multiple process flows are defined in a SAS Enterprise Guide project, they appear as subflows when the project is converted to a SAS Studio process flow.

• SAS Enterprise Guide can be configured to generate different result types, including SAS Report, HTML, PDF, RTF, text output, PowerPoint, and Excel. These settings do not persist with a SAS Enterprise Guide project file, and there are no appearance or advanced settings related to those results. In addition, there are no nodes for these result types in the SAS Studio process flows that are extracted from SAS Enterprise Guide project files. Messages in the conversion report indicate when those nodes have been omitted.

You can configure SAS Studio to generate HTML, PDF, and RTF output for SAS programs, and you can configure the styles for this output. These options are available in SAS Studio by selecting **Preferences** from the main SAS Studio menu. Use the **Results** tab to set your preferred results locations and style settings.

• SAS Enterprise Guide has options for Graph Format and Graph Format for Built-in Graph Tasks. These options include ActiveX, Java, GIF, JPEG, ActiveX image, Java image, SAS EMF, and PNG. These options do not persist with the SAS Enterprise Guide project file.

SAS Studio uses HTML5 for graphics. In SAS Studio, you have the option of generating HTML graphs as SVG. You can change this setting by selecting **Preferences** from the main SAS Studio menu. Click the **Results** tab and select **Generate HTML graphs as SVG**.

These setting differences can have a very noticeable effect on the look of your converted graphs.

• SAS Enterprise Guide can be configured to insert custom code at the beginning and end of programs, tasks, and queries. SAS Studio does not include this feature. However, if you open a SAS Enterprise Guide project that contains programs, tasks, or queries with custom code in SAS Studio, SAS Studio inserts the custom code in the converted programs, tasks, and queries. Notes are added to the conversion report for each inserted custom code segment before and after programs, tasks, and queries.

• The **Use labels for column names** option in SAS Enterprise Guide Export File nodes is not supported in SAS Studio.
# Text Encoding Options and Language Mappings

The following table lists some of the text encoding options and the languages they are associated with. For more information about specifying a text encoding option, see “Setting General Preferences” on page 131.

<table>
<thead>
<tr>
<th>Text Encoding Option</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows-1250</td>
<td>(Central European languages): Polish, Czech, Slovak, Hungarian, Slovenian, Serbian Latin, Croatian, Bosnian, Romanian, Albanian</td>
</tr>
<tr>
<td>Windows-1251</td>
<td>(Cyrillic languages): Russian, Byelorussian, Bulgarian, Serbian Cyrillic, Macedonian, Ukrainian</td>
</tr>
<tr>
<td>Windows-1252</td>
<td>(Western European languages): Afrikaans, Basque, Catalan, Valencian, Welsh, Danish, German, English, Spanish, Basque, Finnish, Faroese, French, Western Frisian, Irish, Galician, Indonesian, Icelandic, Italian, Inuktitut, Luxembourgish, Malay, Norwegian Bokmål, Dutch, Norwegian Nynorsk, Portuguese, Quechua, Romansh, Northern Sami, Swedish, Swahili, Tswana, Xhosa, Zulu</td>
</tr>
<tr>
<td>Windows-1253</td>
<td>Greek</td>
</tr>
<tr>
<td>Windows-1254</td>
<td>Turkish</td>
</tr>
<tr>
<td>Text Encoding Option</td>
<td>Language</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Windows-1255</td>
<td>Hebrew</td>
</tr>
<tr>
<td>Windows-1256</td>
<td>Arabic</td>
</tr>
<tr>
<td>Windows-1257</td>
<td>(Baltic languages): Estonian, Latvian, Lithuanian</td>
</tr>
<tr>
<td>Windows-1258</td>
<td>Vietnamese</td>
</tr>
</tbody>
</table>
Appendix 4

Customized Output Environment

Overview

You must customize the SAS Studio output environment to perform any of these tasks:

- generate output for other output destinations
- send your results to another location
- use a custom style for your output
- use an image format other than the default
- create a drill-down graph
- create an animated GIF or SVG image

To customize the SAS Studio output environment, first disable the default output environment in order to conserve system resources. Next, establish your own output environment, and then execute the SAS statements that are required to generate your output. Use ODS statements, ODS procedures, or ODS options in your SAS program to define the environment that you need.

As a best practice, if your SAS program requires a customized output environment in SAS Studio, your program should always perform these steps:

1. Create a file reference for your ODS output. You can use the &_SASWS_ macro variable that is defined in SAS Studio to reference your home directory as shown in the following statement:

   filename odsout "&_SASWS_/charts";
If you want to store your image files in a separate directory, create a second file reference for your image files as shown in the following statement:

```plaintext
filename ods1out "&_SASWS_/charts/images";
```

*Note:* The directories that you specify must already exist, and you must have Write access to the directories.

2. To conserve system resources, disable the default output environment by using the following statement:

```plaintext
ods _all_ close;
```

3. Open the desired ODS destination. Use the PATH= option to specify the file reference that you created for your ODS output. If you created a separate file reference for your image files, use the GPATH= option to specify the image output file reference. Here is an example:

```plaintext
ods html path=odsout gpath=ods1out file="saleschart.html";
```

4. Execute the SAS statements that are required to generate your output.

5. Close your ODS destination.

When you disable the default SAS Studio output environment, results are no longer displayed on the **Results** tab for the duration of your program. The results are generated only by the ODS destination that you open.

---

## Generate Output for Other Output Destinations

If you need to generate output other than the default HTML5, PDF, or RTF output, you must open your own ODS destination. Examples of output destinations include HTML, PowerPoint, and LISTING. After you disable the default output environment, use an ODS statement to open your own output destination. Here is an example:

```plaintext
filename odsout "&_SASWS_/charts";
ods _all_ close;
ods powerpoint path=odsout file="filename";
```

To access the dictionary of ODS statements, see *SAS Output Delivery System: User’s Guide*.

---

## Send Your Results to Another Location

When you execute a program in SAS Studio, you can download the output from the **Results** tab to your local machine. If you want to send your output directly to another location, you must open your own ODS destination. By default, output files that are generated by the ODS destinations that you open are written to your home directory.

In SAS Studio Single-User deployments, the output is written to your home directory on your local machine.

In SAS Studio Basic and in SAS Studio Mid-Tier deployments, the output is written to your home directory on the remote SAS server. Your home directory appears in the navigation pane under **Files and Folders** or **Server Files and Folders**.
If you want to send the results to a specific location, use a FILENAME statement to define a file reference to the desired location. You can use the &_SASWS_ macro variable to reference your home directory. After you create the file reference, use the PATH=option in your ODS statement. Here is an example:

```sas
filename odsout "&_SASWS_/charts";
ods _all_ close;
ods html path=odsout file="sales.htm";
```

In this case, file sales.htm and any image files that are generated are written to subdirectory **charts** in your home directory.

### Use a Custom Style for Your Output

When you need to use a custom ODS style such as a corporate style for your results in SAS Studio, you must open your own ODS destination. You cannot specify a custom style for the default results. Use the STYLE= option in your ODS statement to specify your custom style. Here is an example:

```sas
filename odsout ="&_SASWS_/charts";
ods _all_ close;
ods html path=odsout file="filename.htm" style=style-name;
```

To create a custom style, use the ODS TEMPLATE procedure, CSSStyles, or the STYLE= option. For more information, see *SAS Output Delivery System: User’s Guide*.

### Use an Image Format Other Than the Default

When you need to use an image format other than the default, you must specify the desired output format, and then open your own ODS destination. To specify the image format:

- If you are using SAS/GRAPH to create your graphs, specify the DEVICE= option in an OPTIONS or GOPTIONS statement. For more information, see *SAS/GRAPH: Reference*.
- If you are using ODS Graphics to create your graphs, specify the OUTPUTFMT= option in an ODS GRAPHICS statement. For more information, see *SAS Output Delivery System: User’s Guide*.

### Create a Drill-down Graph

When you need to create a drill-down graph in SAS Studio, you must open your own ODS destination. Drill-down graphs provide a convenient means for users to explore complex data. In a drill-down graph, certain elements of the graph contain active links. When a user clicks a linked element, the linked resource appears in a new browser window by default.

For more information, see the following documents:

- If you are using SAS/GRAPH to create the graph, see *SAS/GRAPH: Reference*.
Create an Animated GIF or SVG Image

When you need to create an animated graph in SAS Studio, you must open your own ODS destination. An animated graph displays a series of charts automatically when the graph is viewed in a web browser or other viewer that supports animation. The animation plays as a sequence of graphs in a slide-show fashion with a delay between each graph. The sequence can play only one time, loop a fixed number of times and then stop, or loop indefinitely.

For more information, see the following documents:

• If you are using SAS/GRAPH to create the graph, see *SAS/GRAPH: Reference*.
• If you are using the Graph Template Language to create the graph, see *SAS Graph Template Language: User’s Guide*. 
# Appendix 5

## Keyboard Shortcuts

Keyboard Shortcuts in SAS Studio Workspace ........................................ 163
Keyboard Shortcuts for the Code Editor .................................................. 165

---

### Keyboard Shortcuts in SAS Studio Workspace

This table lists the keyboard shortcuts for the SAS Studio workspace. Additional keyboard shortcuts are available from “Keyboard Shortcuts for the Code Editor” on page 165.

<table>
<thead>
<tr>
<th>Action</th>
<th>Keyboard Shortcut for Microsoft Windows</th>
<th>Keyboard Shortcut for Mac OS X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zoom in.</td>
<td>Ctrl+plus sign</td>
<td>Cmd+plus sign</td>
</tr>
<tr>
<td>Note: When you zoom into SAS Studio, no scroll bars are displayed for the browser window.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zoom out.</td>
<td>Ctrl+minus sign</td>
<td>Cmd+minus sign</td>
</tr>
<tr>
<td>Reset the zoom state.</td>
<td>Ctrl+0</td>
<td>Cmd+0</td>
</tr>
<tr>
<td>Maximize the view of the currently displayed tab or exit the maximized view.</td>
<td>Alt+F11</td>
<td></td>
</tr>
<tr>
<td>Open a pop-up menu in the navigation pane.</td>
<td>Select an item, and press Ctrl+Shift+F10.</td>
<td>Select an item, and press Fn +Shift+F10.</td>
</tr>
<tr>
<td>Note: If you use Ctrl+Shift+F10 to display the pop-up menu, then it is always displayed in the top left corner of the user interface control that you are using.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Keyboard Shortcut for Microsoft Windows</td>
<td>Keyboard Shortcut for Mac OS X</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Create a new SAS program.</td>
<td>F4</td>
<td>Fn+F4</td>
</tr>
</tbody>
</table>
| Save the SAS program. | Ensure that the Code tab for a SAS program is displayed, and press Ctrl+S.  
*Note:* This shortcut does not work for the Code tab that displays a task's XML code. | Ensure that the Code tab for a SAS program is displayed, and press Cmd+S.  
*Note:* This shortcut does not work for the Code tab that displays a task's XML code. |
| Run the code. | Ensure that a Code tab is displayed, and press F3.  
*Note:* In JAWS, F3 is the keyboard shortcut for the find functionality. Therefore to run SAS code, you first must press Insert+3 and then press F3. Alternatively, turn off the JAWS virtual PC cursor by pressing Insert+Z. | Ensure that a Code tab is displayed, and press Fn+F3. |
| Reset the SAS session. | F9 | Fn+F9 |
| Print the SAS program, log, or results. | Ctrl+P  
*Note:* This shortcut does not work for the Log and Results tabs when you use Internet Explorer. This shortcut requires Internet Explorer 10 or later. | Cmd+P to open the content in a new tab. Then press Cmd+P again to print the content. |
| Close the selected primary tab. | Ensure that the focus is on the tab label (press Alt+3 if necessary), and then press Delete.  
*Note:* You cannot close the Code, Log, and Results tabs. | Ensure that the focus is on the tab label (press Option+3 if necessary), and then press Delete.  
*Note:* If your keyboard does not contain a key for deleting forward, then press Fn+Delete. |
| Move the focus to the Server Files and Folders section label. | Alt+1  
*Note:* This shortcut requires Internet Explorer 10 or later. | Option+1 |
### Keyboard Shortcuts for the Code Editor

This table contains keyboard shortcuts for the code editor in SAS Studio. Additional keyboard shortcuts are available from “Keyboard Shortcuts in SAS Studio Workspace” on page 163.

<table>
<thead>
<tr>
<th>Action</th>
<th>Keyboard Shortcut for Microsoft Windows</th>
<th>Keyboard Shortcut for Mac OS X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move the focus to the label of the currently displayed secondary tab.</td>
<td>Alt+2</td>
<td>Option+2</td>
</tr>
<tr>
<td>In the SAS Programmer perspective, the secondary tabs are Code, Log, or Results.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>In the Visual Programmer perspective, the secondary tabs are Flow, Results, and Properties.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move the focus to the label of the currently displayed primary tab and then navigate among all open tabs in the work area.</td>
<td>Alt+3</td>
<td>Option+3</td>
</tr>
<tr>
<td>Move the focus to the body of the Code tab on the currently displayed primary tab.</td>
<td>Alt+4</td>
<td>Option+4</td>
</tr>
<tr>
<td>Move the focus to the selected item in the Log tab on the currently displayed primary tab.</td>
<td>Alt+5</td>
<td>Option+5</td>
</tr>
<tr>
<td>Open a pop-up menu in the code editor.</td>
<td>Shift+F10</td>
<td>Fn+Shift+F10</td>
</tr>
<tr>
<td>Create a new snippet from SAS code.</td>
<td>In the code editor, select the code and press Shift+Alt+A.</td>
<td>In the code editor, select the code and press Cmd+ Option +A.</td>
</tr>
<tr>
<td>Note: The snippet is saved to your My Snippets folder.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Keyboard Shortcut for Microsoft Windows</td>
<td>Keyboard Shortcut for Mac OS X</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>----------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Insert an existing snippet.</td>
<td>Alt+I</td>
<td>Cmd+Option+I</td>
</tr>
<tr>
<td>Note: Only snippets saved to your My Snippets folder are available.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add or remove comment tags for SAS code in the code editor.</td>
<td>Ctrl+/</td>
<td>Cmd+/</td>
</tr>
<tr>
<td>Note: This shortcut might not work if your keyboard does not have the forward slash (/) character on the same key as the question mark (?) character. Use Ctrl+Shift+C instead.</td>
<td>Note: This shortcut might not work if your keyboard does not have the forward slash (/) character on the same key as the question mark (?) character. Use Control+Shift +C instead.</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td><strong>Keyboard Shortcut for Microsoft Windows</strong></td>
<td><strong>Keyboard Shortcut for Mac OS X</strong></td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>In the code editor, display the currently applicable list of SAS keywords (autocomplete feature).</td>
<td>Ctrl+Spacebar (This shortcut also navigates to the next list if more than one is available.)</td>
<td>Control+Spacebar (This shortcut also navigates to the next list if more than one is available.)</td>
</tr>
<tr>
<td></td>
<td>Ctrl+Shift+Spacebar (This shortcut also navigates to the previous list if more than one is available.)</td>
<td>Control+Shift+Spacebar (This shortcut also navigates to the previous list if more than one is available.)</td>
</tr>
<tr>
<td></td>
<td>Note: These shortcuts do not work when JAWS is on.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: If you use Ctrl+Spacebar to switch the input method editor (IME) in Windows, complete these steps to use a different shortcut for switching the IME:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• First, in the region and language options of the Control Panel, select the option for changing your keyboard or input method.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Next, select the option or button for changing your keyboard.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Then, in the advanced key settings section, select the row that contains the Ctrl+Spacebar key sequence and change it to use a different shortcut.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Finally, restart any application that needs to use the new shortcut. (You might need to also restart your computer.)</td>
<td></td>
</tr>
<tr>
<td>Navigate between an autocomplete list and its associated Help pop-up window in the code editor.</td>
<td>Tab, Shift+Tab</td>
<td>Tab, Shift+Tab</td>
</tr>
<tr>
<td>Select all text in the code editor.</td>
<td>Ctrl+A</td>
<td>Cmd+A</td>
</tr>
<tr>
<td>Copy text in the code editor.</td>
<td>Ctrl+C</td>
<td>Cmd+C</td>
</tr>
<tr>
<td>Cut text in the code editor.</td>
<td>Ctrl+X</td>
<td>Cmd+X</td>
</tr>
<tr>
<td>Action</td>
<td>Keyboard Shortcut for Microsoft Windows</td>
<td>Keyboard Shortcut for Mac OS X</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>----------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Paste text in the code editor.</td>
<td>Ctrl+V</td>
<td>Cmd+V</td>
</tr>
<tr>
<td>Undo an action in the code editor.</td>
<td>Ctrl+Z</td>
<td>Cmd+Z</td>
</tr>
<tr>
<td>Redo an action in the code editor.</td>
<td>Ctrl+Y</td>
<td>Cmd+Shift+Z</td>
</tr>
<tr>
<td>Find and replace text in the code editor.</td>
<td>Ensure that the focus is in the body of the code editor, and then press Ctrl+F.</td>
<td>Ensure that the focus is in the body of the code editor, and then press Cmd+F.</td>
</tr>
<tr>
<td>Move the cursor to the beginning of the word on the left of the cursor.</td>
<td>Ctrl+left arrow</td>
<td>Option+left arrow</td>
</tr>
<tr>
<td>Move the cursor to the beginning of the word on the right of the cursor.</td>
<td>Ctrl+right arrow</td>
<td></td>
</tr>
<tr>
<td>Move the cursor to the end of the word on the right of the cursor.</td>
<td>Option+right arrow</td>
<td></td>
</tr>
<tr>
<td>Move the cursor to the beginning of the previous paragraph.</td>
<td>Option+up arrow</td>
<td></td>
</tr>
<tr>
<td>Move the cursor to the beginning of the next paragraph.</td>
<td>Option+down arrow</td>
<td></td>
</tr>
<tr>
<td>Scrolls the editor (the cursor does not move).</td>
<td>Ctrl+down arrow, Ctrl+up arrow</td>
<td></td>
</tr>
<tr>
<td>Scrolls a page down or up (moves cursor).</td>
<td>Page Down, Page Up</td>
<td>Page Down, Page Up</td>
</tr>
<tr>
<td>Move the cursor to the end of the line.</td>
<td>End</td>
<td>Cmd+right arrow</td>
</tr>
<tr>
<td>Move the cursor to the beginning of the line.</td>
<td>Home</td>
<td>Cmd+left arrow</td>
</tr>
<tr>
<td>Move to the bottom of the currently displayed section of the navigation pane.</td>
<td>End</td>
<td>Fn+right arrow</td>
</tr>
<tr>
<td>Action</td>
<td>Keyboard Shortcut for Microsoft Windows</td>
<td>Keyboard Shortcut for Mac OS X</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Move to the top of the currently displayed section of the navigation pane.</td>
<td>Home</td>
<td>Fn+left arrow</td>
</tr>
<tr>
<td>Move the cursor to the top of the document.</td>
<td>Ctrl+Home</td>
<td>Cmd+up arrow</td>
</tr>
<tr>
<td>Move the cursor to the bottom of the document.</td>
<td>Ctrl+End</td>
<td>Cmd+down arrow</td>
</tr>
<tr>
<td>Select (highlight) text to the left of the cursor character-by-character.</td>
<td>Shift+left arrow</td>
<td>Shift+left arrow</td>
</tr>
<tr>
<td>Select (highlight) text to the right of the cursor character-by-character.</td>
<td>Shift+right arrow</td>
<td>Shift+right arrow</td>
</tr>
<tr>
<td>Select (highlight) text from the cursor to the same location on the previous line.</td>
<td>Shift+up arrow</td>
<td>Shift+up arrow</td>
</tr>
<tr>
<td>Select (highlight) text from the cursor to the next line.</td>
<td>Shift+down arrow</td>
<td>Shift+down arrow</td>
</tr>
<tr>
<td>Delete the word on the left of the cursor.</td>
<td>Ctrl+Backspace</td>
<td>Option+Delete (backward)</td>
</tr>
<tr>
<td>Note: If your keyboard does not contain a key for deleting backward, then press Fn +Delete.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete the word on the right of the cursor.</td>
<td>Ctrl+Delete</td>
<td>Option+Delete (forward)</td>
</tr>
<tr>
<td>Note: If your keyboard does not contain a key for deleting forward, then press Fn +Delete.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select (highlight) the word on the left of the cursor.</td>
<td>Ctrl+Shift+left arrow</td>
<td>Option+Shift+left arrow</td>
</tr>
<tr>
<td>Select (highlight) the word on the right of the cursor.</td>
<td>Ctrl+Shift+right arrow</td>
<td>Option+Shift+right arrow</td>
</tr>
<tr>
<td>Select (highlight) text from the cursor to the beginning of the document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select (highlight) text from the cursor to the end of the document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Keyboard Shortcut for Microsoft Windows</td>
<td>Keyboard Shortcut for Mac OS X</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>----------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Select a column or vertical block of text.</td>
<td>Alt+left mouse button. Then drag the mouse pointer to select the text.</td>
<td>Option+left mouse button. Then drag the mouse pointer to select the text.</td>
</tr>
</tbody>
</table>
| Change case of selected text.              | Ctrl+Shift+U cycles among initial capital, uppercase, and lowercase.  
                                          | Ctrl+U converts the selected text to uppercase.                      | Cmd+Shift+U cycles among initial capital, uppercase, and lowercase.  
                                          | Ctrl+L converts the selected text to lowercase.                     | Cmd+U converts the selected text to uppercase.                      |
Recommended Reading

- *An Introduction to SAS University Edition*
- *Biostatistics by Example Using SAS Studio*
- *Elementary Statistics Using SAS*
- *Essential Statistics Using SAS University Edition*
- *Getting Started with Programming in SAS Studio*
- *Learning SAS by Example: A Programmer’s Guide*
- *The Little SAS Book: A Primer*
- *SAS Statistics by Example*

For a complete list of SAS publications, go to [sas.com/store/books](http://sas.com/store/books). If you have questions about which titles you need, please contact a SAS Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-0025
Fax: 1-919-677-4444
Email: sasbook@sas.com
Web address: [sas.com/store/books](http://sas.com/store/books)
Index

Special Characters
  _CLIENTAPP  26
  _CLIENTAPPVERSION  26
  _SASPROGRAMFILE  26

A
  accessible graph  136
  accessible PDF  137
  autocomplete  19, 133
  autoexec file  13
  autosaving
    enabling  135

B
  background jobs
    messages  132
  background submissions
    cancelling  140
    creating unique log and output
      filenames  140
    deleting log and output  140
    prompting for fail or overwrite  140
    prompting for location to save log and output  140
    saving log and output to code folder
      location  140
    saving log and output to specified location  140
  brackets
    matching  22
  branches
    creating in Git  111
    merging in Git  112
    understanding in Git  110

C
  changing your SAS workspace server  14
  cloning a Git repository  103
  code
    editing from a task  24
    formatting SAS code  27
  selecting a column  23
  showing in the SAS log  135
  trimming blank spaces  139
  code editor
    about  15
    autocomplete  19, 133
    color coding  134
    customizing  40, 133
    enabling autosave  135
    specifying font size  134
    substituting spaces for tabs  134
    syntax help window  134
    tab width  134
    code snippets  30
    commit history
      viewing in Git  108
      committing changes in Git  105
      custom tasks  128

D
  data
    exporting  92
    extended attributes  75
    filtering  80
    importing  83
    opening  77
    process flows  63
    sorting  80
    table viewer  75
  DBMS file, importing  90
  delimited file, importing  87
  differences between commits in Git  108

E
  editing columns  23
  editor preferences  40, 133
  encoding
    specifying  132
  error messages
    show details  131
  Excel worksheet, importing  84
  exporting data  92
**F**  
File Shortcuts section of the navigation pane 7  
filter  
  changing the relationship 52  
  creating 50  
filtering data 80  
folder shortcuts 3  
formatting SAS code 27  
FTP folder shortcuts 3  

**L**  
libraries  
  using to add table and column names 23  
Libraries section of the navigation pane 6  
log  
  specifying font size 134  
  stream updates 135  
logs  
  displaying line numbers 134  

**G**  
general preferences 131  
Git  
  cloning a repository 103  
  commit history 108  
  committing changes 105  
  creating a branch 111  
  creating a profile 101  
  deleting a repository 112  
  merging a branch 112  
  pulling and pushing files 109  
  resetting local repository 109  
  staging changes 105  
  working with branches 110  
Git integration  
  about 101  
  sample workflow scenario 113  
Git integration in SAS Studio 101  
grouping query output 54  

**H**  
hints  
  code editor 134  
HTML output 136  

**I**  
importing  
  DBMS file 90  
  delimited file 87  
  Excel worksheet 84  
importing data 83  
interactive mode 28, 132  

**J**  
joins  
  about 43  
  creating 43  
  modifying 46  
types 45  

**M**  
macro variables 26  
  _CLIENTAPP 26  
  _CLIENTAPPVERSION 26  
  _SASPROGRAMFILE 26  
matching parentheses 22  
maximizing the work area 9  
messages  
  details in the log 133  
  displaying in SAS Studio 132  
My Tasks folder 127  

**N**  
names  
  SAS variables 138  
navigation pane 3  
  customizing 8  
File Shortcuts 7  
Libraries 6  
Server Files and Folders 3  
Snippets 6  
Tasks 5  

**O**  
ODS (SAS Output Delivery System) 97  
ODS Graphics Designer 97  
ODS Graphics Editor 99  
output data  
  displaying automatically 137  

**P**  
parentheses  
  matching 22  
PDF output 136  
  enable accessible PDF 137  
table of contents 136  
perspectives 12  
preferences  
  editor 40, 133  
  general 131  
  results 136
saving results 57
selecting columns for output 46
sorting output 53
summary functions 48
types of joins 45

R
repository
cloning in Git 103
deleting in Git 112
opening local in Git 103
resetting your local Git repository 109
results
accessible HTML graph option 136
changing styles 136
displaying output data 137
downloading 95
downloading generated data 95
HTML output 136
PDF output 136
PDF output style 136
RTF output 137
sending to another user 96
viewing 93
RTF output 137

S
SAS ODS Graphics 97
SAS Output Delivery System (ODS) 97
SAS program package 25
SAS program summary 26
SAS Programmer perspective 12
SAS workspace server 14
searching in SAS Studio 11
selecting a column of text 23
server 14
Server Files and Folders section of the
navigation pane 3
snippets
about 30
creating 39
inserting 40
Snippets section of the navigation pane 6
sorting data 80
staging changes in Git
committing changes 105
resetting file changes 105
showing file differences 105
stream log updates 135
submission history 27
syntax help 21
syntax help window 134

query
adding tables 42
changing filter relationship 52
creating 42
creating a filter 50
creating a join 43
eliminating duplicate rows 54
joining 54
joining 43
modifying a join 46
running 58

Index 175
T
  tab characters
    number of spaces 134
  tab width
    code editor 134
table viewer 75
  specifying width of columns 132
tables
  displaying the generated code 79
  extended attributes 75
  opening 77
  resizing column widths 77
tabs
  rearranging in the workspace 9
task code
  adding comments 129, 139
  formatting 129, 139
tasks
  about 123
  adding comments before generated code 139
  creating 128
  customizing the layout 139
  editing 127
  editing the generated code 24
  formatting generated code 139
  running 125
  showing SAS code 139
  showing the log 139
  Tasks and Utilities section of the navigation pane 5

V
  Visual Programmer perspective 12

W
  warnings
    opening large files 136
  work area 8
    customizing 9
    rearranging tabs 9
  workspace server 14

X
  XML file, creating 3
  XML templates 128
Gain Greater Insight into Your SAS® Software with SAS Books.

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