



SAS[®] Studio 3.6: User's Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS® Studio 3.6: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Studio 3.6: User's Guide

Copyright © 2016, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

June 2017

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

3.6-P1:webeditorug

Contents

<i>About This Book</i>	v
<i>What's New in SAS Studio 3.6</i>	vii
Chapter 1 • Introduction to SAS Studio	1
About SAS Studio	1
Using SAS Studio	2
Chapter 2 • Working with Programs	17
About the Code Editor	17
Opening and Creating Programs	18
Working with Code Snippets	31
Customizing the Code Editor	38
Chapter 3 • Working with Queries	39
What Is a Query?	39
Creating a Query	40
Understanding Joins	41
Selecting Data	44
Filtering Data	48
Managing Output	51
Chapter 4 • Working with Process Flows	57
Understanding Process Flows	57
Add Data to the Process Flow	61
Adding a SAS Program to the Process Flow	62
Adding a Query to a Process Flow	67
Add a Task to a Process Flow	68
Understanding Subflows	69
Linking Nodes in a Process Flow	71
Generating Code from a Process Flow	71
Running a Process Flow	71
Save a Process Flow	72
Chapter 5 • Working with Data	73
About the Table Viewer	73
Opening and Viewing Data	75
Viewing the Code That Is Used to Create a Table	77
Filtering and Sorting Data	78
Importing Data	81
Exporting Data	90
Chapter 6 • Working with Results	91
Viewing Results	91
Default SAS Studio Output	91
Sending Your Results to Another User	94
About the SAS Output Delivery System	95
About SAS ODS Statistical Graphics	95

Chapter 7 • Understanding Tasks in SAS Studio	99
What Is a Task?	99
How to Run a Task	100
Save a Task and Its Option Settings	102
Edit a Predefined Task	102
Create a New Task	103
Customizing the Task Code and the Task Layout in the Workspace	104
Appendix 1 • Customizing SAS Studio	105
About Setting Your Preferences	105
Setting General Preferences	105
Setting the Start Up Preferences	107
Setting the Editor Preferences	108
Setting the Result Preferences	110
Setting Task Preferences	111
Access Repositories in SAS Studio	112
Setting Preferences for Background Job Submissions	112
Appendix 2 • Text Encoding Options and Language Mappings	115
About the Text Encoding to Language Mappings	115
Text Encoding Options and Language Mappings	115
Appendix 3 • Customized Output Environment	117
Overview	117
Generate Output for Other Output Destinations	118
Send Your Results to Another Location	118
Use a Custom Style for Your Output	119
Use an Image Format Other Than the Default	119
Create a Drill-down Graph	119
Create an Animated GIF or SVG Image	120
Appendix 4 • Keyboard Shortcuts	121
Keyboard Shortcuts in SAS Studio Workspace	121
Keyboard Shortcuts for the Code Editor	123
Recommended Reading	129
Index	131

About This Book

Audience

This book is designed for all users of SAS Studio. SAS Studio was initially released with the first maintenance release for SAS 9.4. SAS Studio 3.6 is the latest release.

What's New in SAS Studio 3.6

Overview

SAS Studio 3.6 includes these new features and enhancements:

- ability to create a SAS program from a process flow. For more information, see [“Creating a SAS Program from a Process Flow” on page vii](#).
- new code snippets that help you connect to SAS Viya. For more information, see [“New Code Snippets” on page viii](#).
- enhancements to the background submit feature. For more information, see [“New Background Submit Features” on page viii](#).
- new preferences that enable you to control whether items in the navigation pane are automatically refreshed. For more information, see [“Preferences for Refreshing the Navigation Pane” on page viii](#).
- new Start Up preferences that determine whether SAS Studio opens in the state that you closed it. For more information, see [“New Start Up Preferences” on page viii](#).
- ability to generate HTML graphs in the SVG format. For more information, see [“New HTML Graph Format” on page viii](#).
- ability to match parentheses, square brackets, and braces. For more information, see [“New Parentheses Matching Feature” on page viii](#).
- many new tasks for power and sample size analysis, cluster analysis, and network optimization. New Viya Machine Learning tasks enable you to run SAS code in a SAS Viya environment. For more information, see [“SAS Studio Tasks” on page ix](#).

Creating a SAS Program from a Process Flow

You can generate 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. For more information, see [“Generating Code from a Process Flow” on page 71](#).

New Code Snippets

There is a new category of Viya Cloud Analytic Services code snippets to help you connect to SAS Viya and work with CAS tables. For more information, see [“Why Use Code Snippets?”](#) on page 31.

New Background Submit Features

You can now specify the location of the log and output files. You can also specify what action to take if an output or log file already exists. For more information, see [“Setting Preferences for Background Job Submissions”](#) on page 112.

Preferences for Refreshing the Navigation Pane

You can now specify whether SAS Studio automatically refreshes the libraries and the files and folders after you run a program, task, or query. For more information, see [“Setting General Preferences”](#) on page 105.

New Start Up Preferences

You can specify whether SAS Studio attempts to restore the tabs that were open in your prior session or open with a new program or process flow tab. For more information, see [“Setting the Start Up Preferences”](#) on page 107.

New HTML Graph Format

You can choose to generate HTML graphs in the SVG format instead of the PNG format. For more information, see [“Setting the Result Preferences”](#) on page 110.

New Parentheses Matching Feature

You can match nested parentheses, square brackets, and braces within a program. For more information, see [“Matching Parentheses”](#) on page 24.

SAS Studio Tasks

New Tasks

The tasks that are shipped with SAS Studio are documented in *SAS Studio: Task Reference Guide*. All of the tasks that are available with SAS Studio are documented. However, the tasks that are available at your site depend on whether you license and install other SAS products.

- Power and Sample Size
 - The One-Way ANOVA task calculates power and sample size analyses for one degree of freedom contrasts and the overall F test. For more information, see “One-Way ANOVA” in *SAS Studio: Task Reference Guide*.
 - The Logistic Regression task calculates power and sample size analyses for the likelihood ratio chi-square test of a single predictor in binary logistic regression, possibly in the presence of one or more covariates that might be correlated with the tested predictor. For more information, see “Logistic Regression” in *SAS Studio: Task Reference Guide*.
 - The Survival Rank Test task calculates power, sample size, or number of events for two-sample survival rank tests. For more information, see “Survival Rank Test” in *SAS Studio: Task Reference Guide*.
 - The Wilcoxon Test task calculates power and sample size analyses for the Wilcoxon-Mann-Whitney test for two independent groups. This test is also called the Wilcoxon rank-sum test, Mann-Whitney-Wilcoxon test, and Mann-Whitney U test. For more information, see “Wilcoxon Test” in *SAS Studio: Task Reference Guide*.
- Cluster Analysis
 - The Compute Similarities and Distances task computes various measures of distance, dissimilarity, or similarity between observations in an input SAS data set. For more information, see “Compute Similarities and Distances” in *SAS Studio: Task Reference Guide*.
 - The Cluster Variables task finds clusters of variables to use in additional clustering or to select non-redundant variables in further clustering. For more information, see “Cluster Variables” in *SAS Studio: Task Reference Guide*.
 - The K-Means Clustering task enables you to cluster interval or ratio data using the k -means algorithm. This task is a good starting point for cluster analysis. For more information, see “K-Means Clustering” in *SAS Studio: Task Reference Guide*.
 - The Cluster Observations task enables you to perform hierarchical or k -means cluster analysis. The Cluster Observations task can be used either with coordinate data or distance data. The distance data can be created using the Compute Similarities and Distances task. For more information, see “Cluster Observations” in *SAS Studio: Task Reference Guide*.
 - The Estimate Cluster Covariances task uses the Art, Gnanadesikan, and Kettenring method to estimate within-cluster covariances. This approach might

result in more spherical clusters when the resulting canonical variables are clustered. For more information, see [“Estimate Within-Cluster Covariances”](#) in *SAS Studio: Task Reference Guide*.

- Network Optimization
 - The Biconnected Components task finds all biconnected components. For more information, see [“Biconnected Components”](#) in *SAS Studio: Task Reference Guide*.
 - The Connected Components task finds all connected components of a graph. For more information, see [“Connected Components”](#) in *SAS Studio: Task Reference Guide*.
 - The Cycle Detection task finds the elementary cycles of an input graph. An elementary cycle is a path in which the start node and the end node are the same and no node appears more than once in the sequence. For more information, see [“Cycle Detection”](#) in *SAS Studio: Task Reference Guide*.
 - The Linear Assignment task assigns any abstract object from one group to some abstract object from a second group at minimal costs. For more information, see [“Linear Assignment”](#) in *SAS Studio: Task Reference Guide*.
 - The Maximal Cliques task finds the maximal cliques of a graph. A clique is an induced subgraph such that every node in that subgraph is connected to every other node. A maximal clique is a clique that is not a subset of the nodes of any larger clique. For more information, see [“Maximal Cliques”](#) in *SAS Studio: Task Reference Guide*.
 - The Minimum Cut task finds a minimum cut of an undirected graph that has the smallest link metric. A cut is a partition of the nodes of a graph into two disjoint subsets. For more information, see [“Minimum Cut”](#) in *SAS Studio: Task Reference Guide*.
 - The Minimum Spanning Tree task finds the spanning tree, among all possible spanning trees, that has the minimum link cost. For more information, see [“Minimum Spanning Tree”](#) in *SAS Studio: Task Reference Guide*.
 - The Minimum Cost Network Flow task sends flow over a network at minimal cost. For more information, see [“Minimum Cost Network Flow”](#) in *SAS Studio: Task Reference Guide*.
 - The Shortest Path task calculates paths between sets of nodes on the input graph with the lowest total link weight. For more information, see [“Shortest Path”](#) in *SAS Studio: Task Reference Guide*.
 - The Transitive Closure task calculates the transitive closure of a graph. For more information, see [“Transitive Closure”](#) in *SAS Studio: Task Reference Guide*.
 - The Traveling Salesman Problem task finds a minimum cost tour in a graph. A tour of a graph is a sequence of nodes where the start node and the end node are the same. Every node in the graph is visited once. In solving this problem, the goal is to find a tour that minimizes the sum of the costs of the links in the tour. For more information, see [“Traveling Salesman Problem”](#) in *SAS Studio: Task Reference Guide*.
- Viya Machine Learning
 - The Connect to Viya task enables you to connect your SAS Studio session to a SAS Viya environment. For more information, see [“Connect to Viya”](#) in *SAS Studio: Task Reference Guide*.

- The Factorization Machine task combines the advantages of smart machine vision (SMV) with factorization models. For more information, see [“Factorization Machine”](#) in *SAS Studio: Task Reference Guide*.
- The Forest task produces an ensemble of tree-based statistical models called decision trees for interval or nominal targets. It uses the Random Forest approach. For more information, see [“Forest”](#) in *SAS Studio: Task Reference Guide*.
- The Gradient Boosting task produces an ensemble of tree-based statistical models called decision trees for interval or nominal targets. It uses the gradient boosting approach. For more information, see [“Gradient Boosting”](#) in *SAS Studio: Task Reference Guide*.
- The Neural Network task uses a large set of adaptive weights to accurately approximate complex and nonlinear relationships between inputs and targets. For more information, see [“Neural Network”](#) in *SAS Studio: Task Reference Guide*.
- The Support Vector Machine task performs classification analysis for binary targets by using a support vector machine, which is a supervised machine learning method. For more information, see [“Support Vector Machine”](#) in *SAS Studio: Task Reference Guide*.

Enhancements to the Common Task Model

Here are the enhancements to the Common Task Model for SAS Studio 3.6:

- The new `passwordtext` control enables you to prompt for a password in your custom tasks.
- For the predefined `$CTMUtil` variable, the new `doublequotestring` method encloses a string in double quotation marks.
- For the `datasource` object in Velocity, the new `getDistinctValues` function returns an array of the distinct values for a given column name for the current data source.
- To use the `getWhereClause` method, you must specify `where = “true”` in the `DataSource` element. Any filter that is added to a data source can affect any distinct controls (such as the `getDistinctCount` and `getDistinctValues` methods) that are associated with the same data source.

For more information about the common task model and writing your own custom tasks for SAS Studio, see *SAS Studio 3.6 Developer's Guide for Writing Custom Tasks*.

Chapter 1

Introduction to SAS Studio

About SAS Studio	1
Using SAS Studio	2
About Using SAS Studio	2
Using the Navigation Pane	3
Using the Work Area	8
Rearranging the Tabs in the Work Area	9
Searching in SAS Studio	11
Using the Messages Window	12
Understanding Perspectives	13
Editing the Autoexec File	14
Changing Your SAS Workspace Server	15

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 System 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.

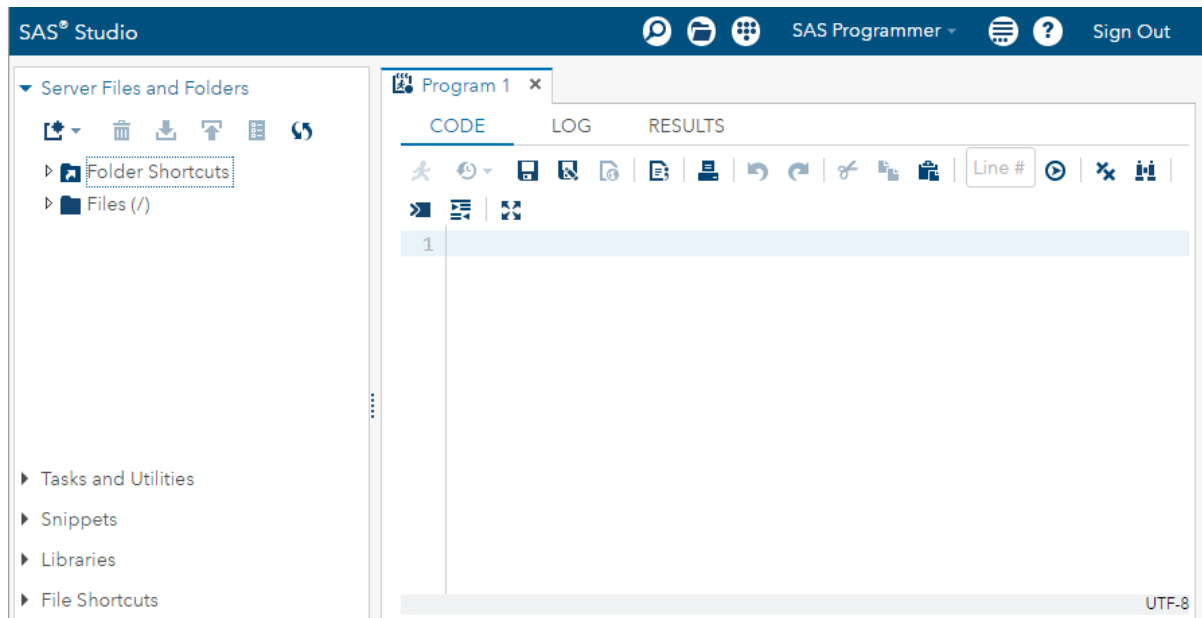
SAS Studio includes two different perspectives: the SAS Programmer perspective and the Visual Programmer perspective. A *perspective* is a predetermined set of features that is customized to meet the needs of a specific user type. By selecting a specific perspective, you can narrow the choices that are 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. For more information, see [“Understanding Perspectives” on page 13](#).

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 all five sections of the navigation pane.

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 the following locations:

- your SAS server
- any remote FTP server on which you have an account


Note: SAS Studio supports only FTP servers that use a directory listing similar to UNIX.

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.

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 7, “Understanding Tasks in SAS Studio,” on page 99](#).

To edit a task that you have created, select the task from the My Tasks folder and click . 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 102](#).

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 17](#).





To edit a snippet that you have created, select the snippet from the My Snippets folder and click . 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.


Icon	Type of Column
	Character
	Numeric
	Date
	Datetime

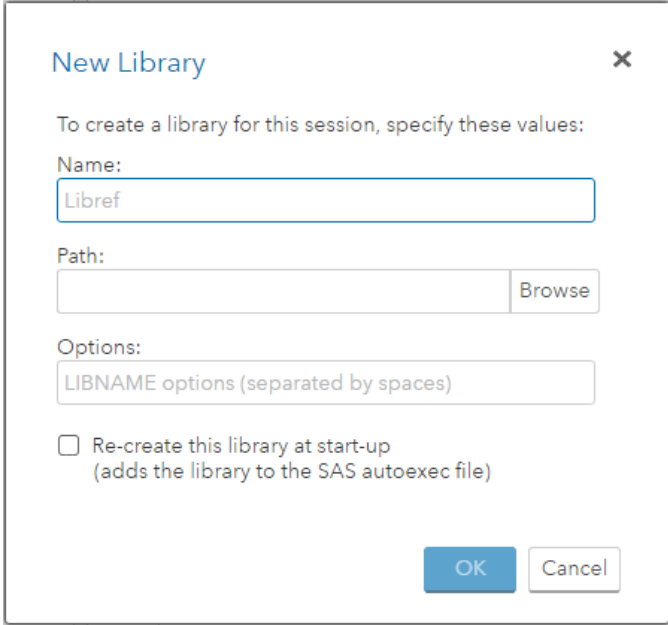
You can drag tables and columns from the **Libraries** section to a program, and SAS Studio adds code for the dragged items to your program. For more information, see [“Opening and Creating Programs” on page 18](#).


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.




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, all 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.

The screenshot displays the SAS Studio interface. On the left, the **DATA** pane shows the dataset `SASHELP.CLASS` with the following filters:

- WHERE CLAUSE FILTER:** None
- ROLES:**
 - Category variable:** Age
 - Response variable:** Height
 - Group variable:** Age
 - URL variable:** Column
 - BY variable:** Column
- DIRECTION:** None
- GROUP LAYOUT:** None
- STATISTICS:** None

On the right, the **CODE** pane shows the following SAS code:

```

11 * Generated on web client 'http://sasstudio36
12 *
13 */
14
15 /*--Set output size--*/
16 ods graphics / reset imagemap;
17
18 /*--SGPLOT proc statement--*/
19 proc sgplot data=SASHELP.CLASS;
20     /*--TITLE and FOOTNOTE--*/
21     title H=16pt "Class Bar Chart";
22     footnote2 j=1 "Class Bar Chart";
23
24     /*--Bar chart settings--*/
25     vbar Age / response=Height group=Age group
26         dataskin=Crisp name='Bar';
27
28     /*--Category Axis--*/
29     xaxis discreteorder=data;
30
31     /*--Response Axis--*/
32     yaxis grid;
33
34     /*--Legend Settings--*/
35     keylegend / location=Inside across=1;
36 run;
37
38 ods graphics / reset;
39 title;
40


```


The status bar at the bottom right indicates "Line 15, Column 24" and "UTF-8".

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.

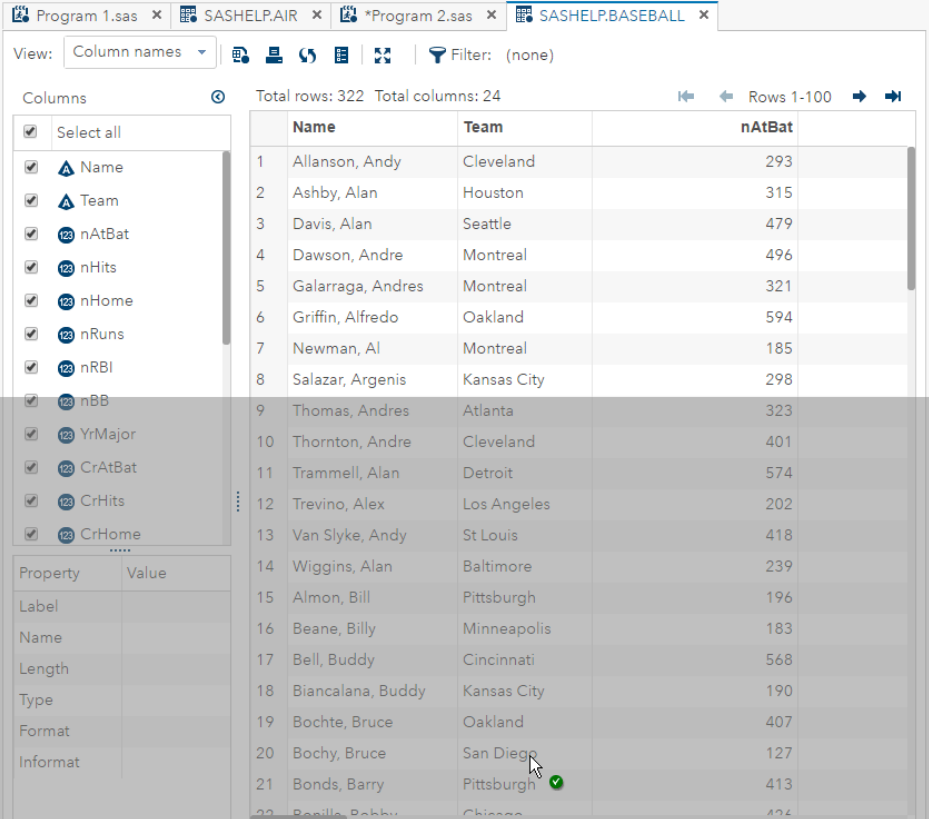
Rearranging the Tabs in the Work Area

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.



The screenshot shows the SAS Studio interface with a data table. The table has 24 columns and 322 rows. The columns are: Name, Team, nAtBat, nHits, nHome, nRuns, nRBI, nBB, YrMajor, CrAtBat, CrHits, and CrHome. The table is currently displaying rows 1 through 22. A mouse cursor is hovering over the 'Pittsburgh' cell in the 'Team' column of row 21, which has a green checkmark icon next to it.

	Name	Team	nAtBat
1	Allanson, Andy	Cleveland	293
2	Ashby, Alan	Houston	315
3	Davis, Alan	Seattle	479
4	Dawson, Andre	Montreal	496
5	Galarraga, Andres	Montreal	321
6	Griffin, Alfredo	Oakland	594
7	Newman, Al	Montreal	185
8	Salazar, Argenis	Kansas City	298
9	Thomas, Andres	Atlanta	323
10	Thornton, Andre	Cleveland	401
11	Trammell, Alan	Detroit	574
12	Trevino, Alex	Los Angeles	202
13	Van Slyke, Andy	St Louis	418
14	Wiggins, Alan	Baltimore	239
15	Almon, Bill	Pittsburgh	196
16	Beane, Billy	Minneapolis	183
17	Bell, Buddy	Cincinnati	568
18	Biancalana, Buddy	Kansas City	190
19	Bochte, Bruce	Oakland	407
20	Bochy, Bruce	San Diego	127
21	Bonds, Barry	Pittsburgh 	413
22	Bonilla, Bobby	Chicago	424

The screenshot shows the SAS Studio interface. The top window, titled "Program 2.sas", has three tabs: CODE, LOG, and RESULTS. The CODE tab is active, showing the following SAS code:

```
1 proc print data=sashelp.baseball;
2 run;
3
```

The RESULTS tab displays a table of baseball statistics. The table has the following columns: Obs, Name, Team, nAtBat, nHits, nHome, nRuns, nRBI, and nBB. The data is as follows:

Obs	Name	Team	nAtBat	nHits	nHome	nRuns	nRBI	nBB
1	Allanson, Andy	Cleveland	293	66	1	30	29	14
2	Ashby, Alan	Houston	315	81	7	24	38	39
3	Davis, Alan	Seattle	479	130	18	66	72	76
4	Dawson, Andre	Montreal	496	141	20	65	78	37
5	Galarraga, Andres	Montreal	321	87	10	39	42	30
6	Griffin, Alfredo	Oakland	594	169	4	74	51	35
7	Newman, Al	Montreal	185	37	1	23	8	21


Below the main window, the SASHELP.BASEBALL dataset is open. The "View" dropdown is set to "Column names". The "Columns" list shows the following columns selected: Name, Team, nAtBat, and nHits. The "Property" and "Value" table is empty. The main data table shows the following columns: Name, Team, and nAtBat. The data is as follows:

	Name	Team	nAtBat
1	Allanson, Andy	Cleveland	293
2	Ashby, Alan	Houston	315
3	Davis, Alan	Seattle	479
4	Dawson, Andre	Montreal	496
5	Galarraga, Andres	Montreal	321
6	Griffin, Alfredo	Oakland	594
7	Newman, Al	Montreal	185
8	Salazar, Argenis	Kansas City	298

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.

The screenshot shows the SAS Studio search interface. The search term is 'class'. The search results are displayed in a table with three columns: Library, Table, and Column. The results are filtered to show only 'CLASS' objects from the SASHELP library.

Library	Table	Column
SASHELP	ASSOCWA	PREDCLAS Label:Predecessor Object Class
SASHELP	ASSOCWA	SUCCCLAS Label:Successor Object Class
SASHELP	BDVITEM	CLASS
SASHELP	BDVLD3	CLASS
SASHELP	BDVMETH	CLASS
SASHELP	CLASS	
SASHELP	Description: Student Data	
SASHELP	CLASSFIT	
SASHELP	Description: Predicted Weights with C	
SASHELP	DMTOOLS	CLASS
SASHELP	EMTOOL	class
SASHELP	GEOEXS	MTFCC Label:MAF/TIGER Feature Class Code

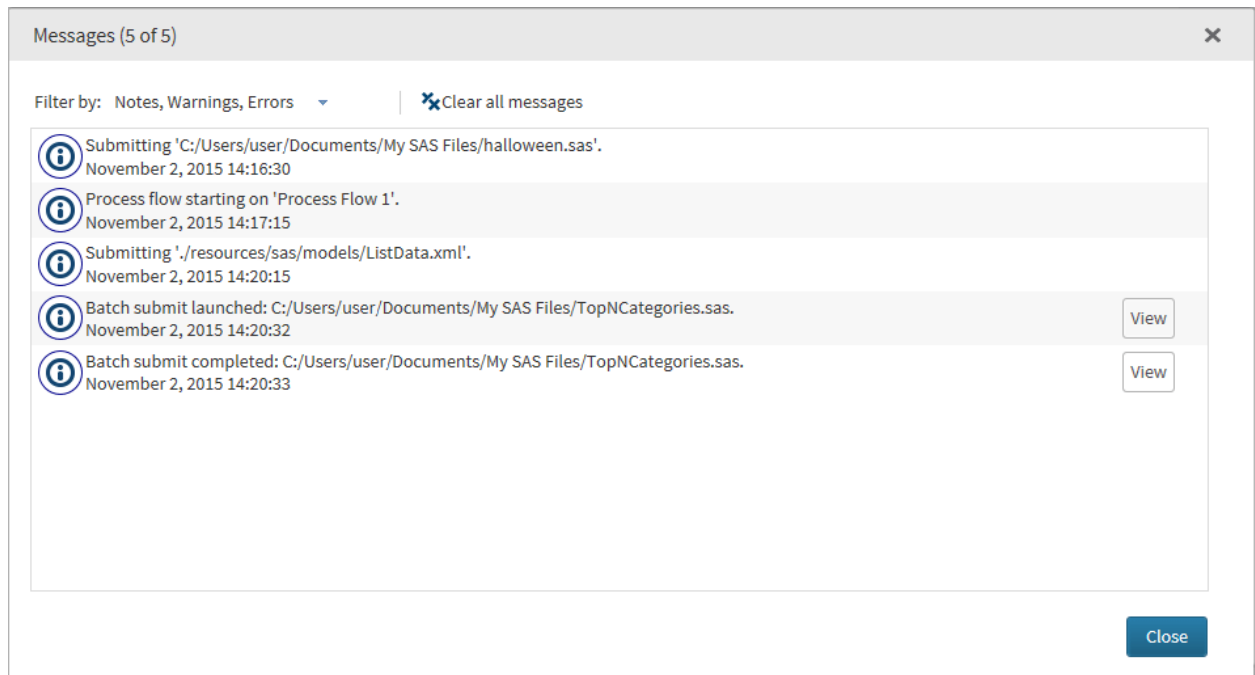
You can open an item from your results by double-clicking it. To return to your search results list, click  again.

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 Batch Submission Status window. For more information, see [“Running a Program as a Background Job” on page 19](#). 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 105.](#)

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 all five sections of the navigation pane so that you can easily manage all of your program files and code snippets.

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 four sections in the navigation pane: Folders, Tasks, Snippets, and Libraries.


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:

Element in SAS Studio	Available in SAS Programmer?	Available in Visual Programmer?
Navigation Pane sections		
• Server Files and Folders	Yes	Yes
• Tasks and Utilities	Yes	Yes
• Snippets	Yes	Yes
• Libraries	Yes	Yes
• File Shortcuts	Yes	Yes, but not displayed by default
SAS Programs	Yes	Yes
Queries	Yes	Yes
Process Flows	No	Yes

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.

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	17
Opening and Creating Programs	18
Opening a Program	18
Creating a Program	18
Running a Program	18
Using the Background Submit Feature	19
Using the Autocomplete Feature	21
Using the Syntax Help	23
Matching Parentheses	24
Selecting Columns of Text to Edit	25
Adding Table Names and Column Names	25
Editing the Code from a Task	26
Creating a SAS Program Package	26
Creating a Program Summary	27
Using Macro Variables	27
Using Your Submission History	28
Automatically Formatting Your SAS Code	28
Working in Interactive Mode	29
Working with Code Snippets	31
Why Use Code Snippets?	31
Create a Code Snippet	37
How to Insert a Code Snippet	37
Customizing the Code Editor	38

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.


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 in a new tab in the work area.

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 opens in a new tab in the work area.

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

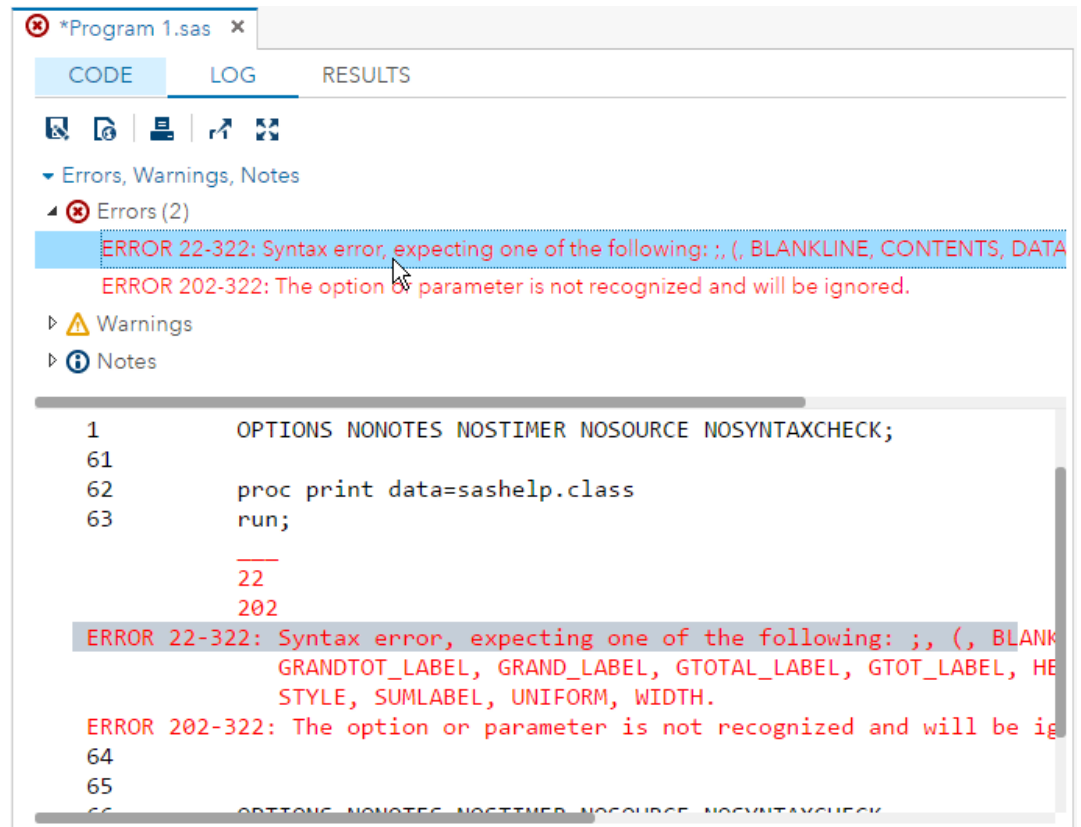
After you save your program, you can use the **Enable autosave** option to automatically create a copy of your new program as well as each previously saved program file that you are editing. You can use the auto-saved copies to recover the files if you lose your network connection or your browser closes unexpectedly. For more information, see [“Setting the Editor Preferences” on page 108](#).


Note: The **Enable autosave** option is available only for programs that you have saved at least once.

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 on a message, SAS Studio highlights it for you in the log so that you can see exactly where the message occurs in the log.



Note: Because you are working in a server environment, do not include the ENDSAS statement in your SAS programs. If you run a program that contains ENDSAS, reset your SAS session by clicking  and selecting **Reset SAS Session**.

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

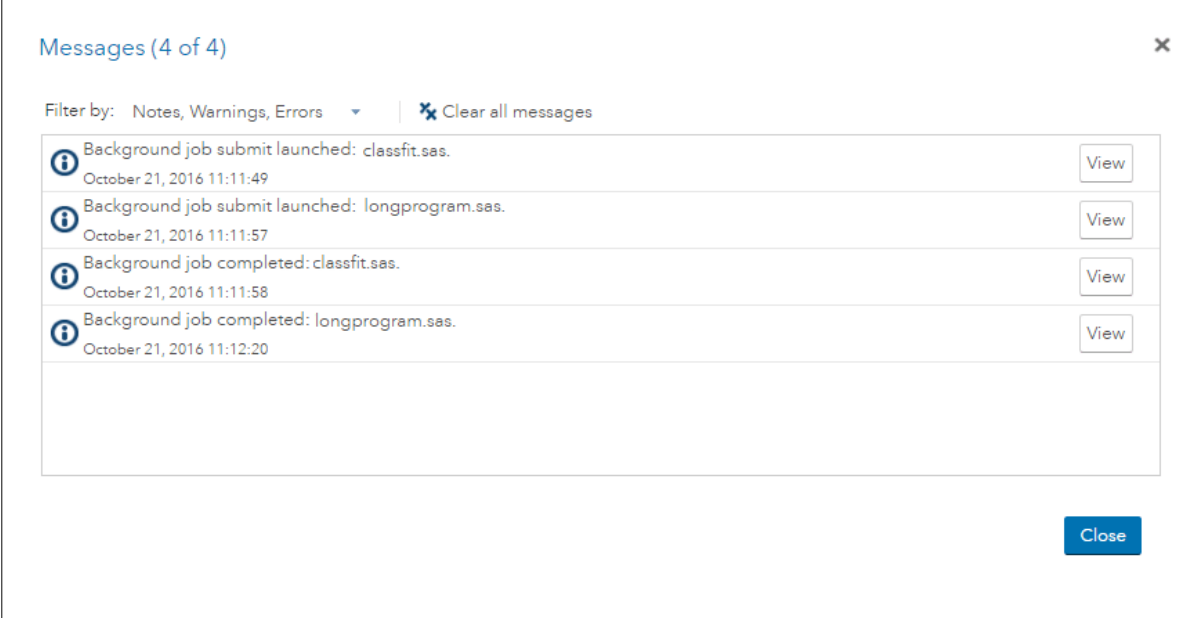
To run a program as a background job, right-click the program in the navigation pane and select **Background Submit**. Before the program is run, the background process changes the current working directory to the directory in which the program is located.

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 of 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**.







Messages (4 of 4) ✕

Filter by: Notes, Warnings, Errors ✕ Clear all messages

	Background job submit launched: classfit.sas. October 21, 2016 11:11:49	View
	Background job submit launched: longprogram.sas. October 21, 2016 11:11:57	View
	Background job completed: classfit.sas. October 21, 2016 11:11:58	View
	Background job completed: longprogram.sas. October 21, 2016 11:12:20	View

Close


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

-  Display the background job properties, including file pathnames, start time, end time, and status.
-  Open the program log in the SAS Studio work area.
-  Open the program results in a separate window or download the results file.
-  Cancel 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 112.

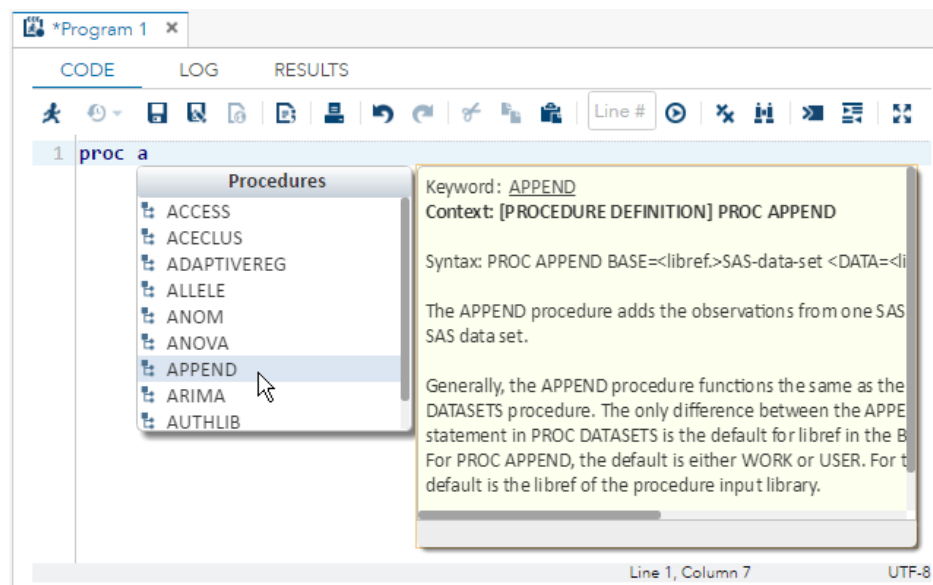
Using the Autocomplete Feature

About the Autocomplete Feature

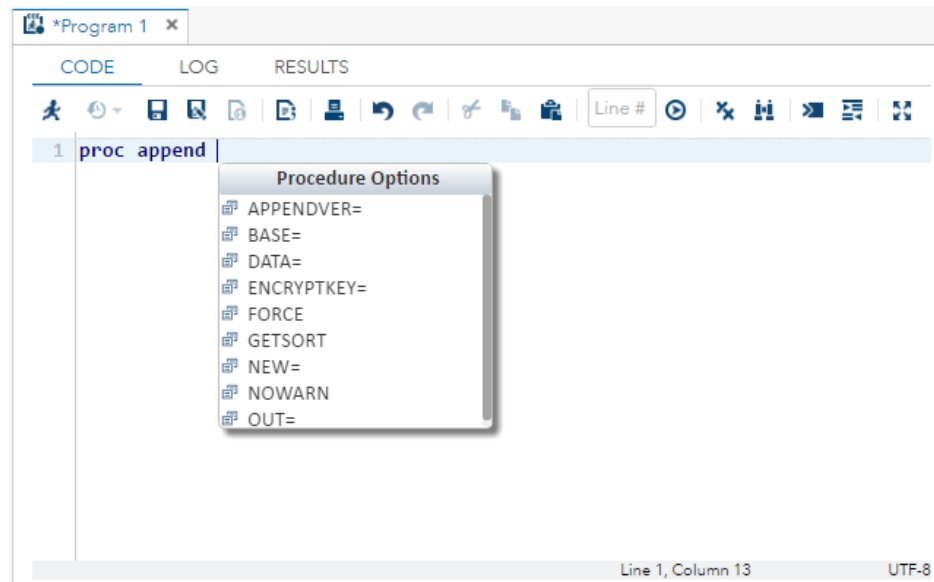
The autocomplete, or code completion, feature in the code editor can predict the next word that you want to enter before you actually enter it completely. The autocomplete feature can complete keywords that are associated with SAS procedures, statements, macros, functions, CALL routines, formats, informats, 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.

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.

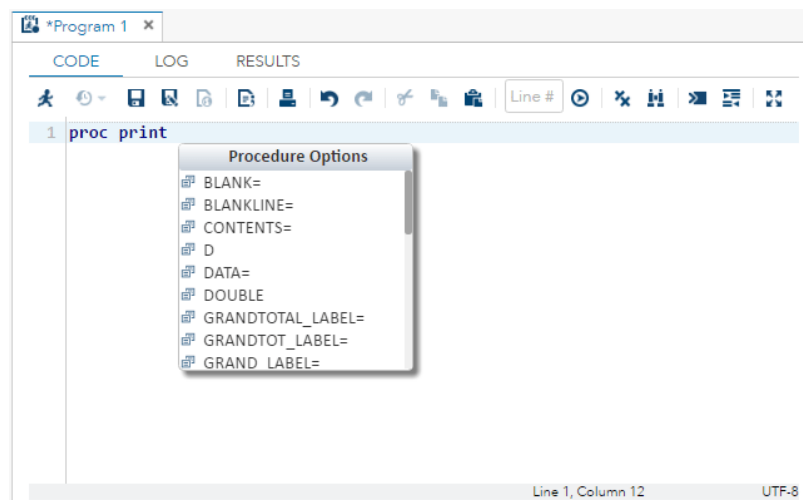


How to Use the Autocomplete Feature

To use the autocomplete feature:

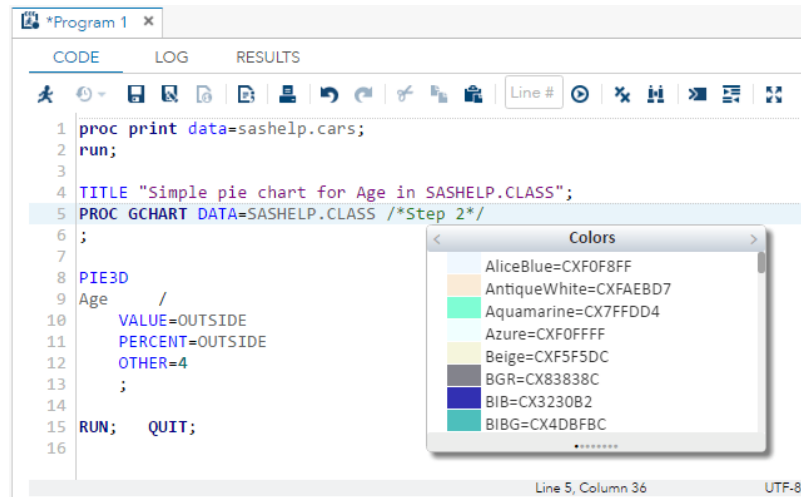
1. How you open the autocomplete list depends on the keyword that you want to add.
 - If you want to add a global statement, DATA step statement, CALL routine, procedure, macro statement, or automatic macro variable, enter the first one or more letters of the keyword that you want to use.

A window opens 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 38.



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.

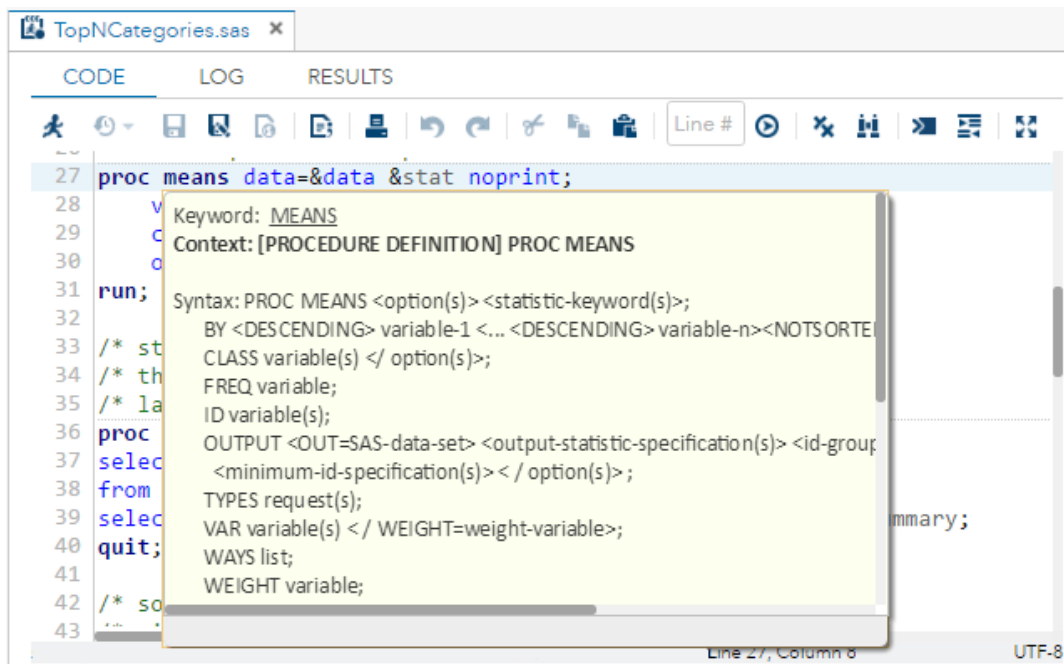
Using the Syntax Help

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

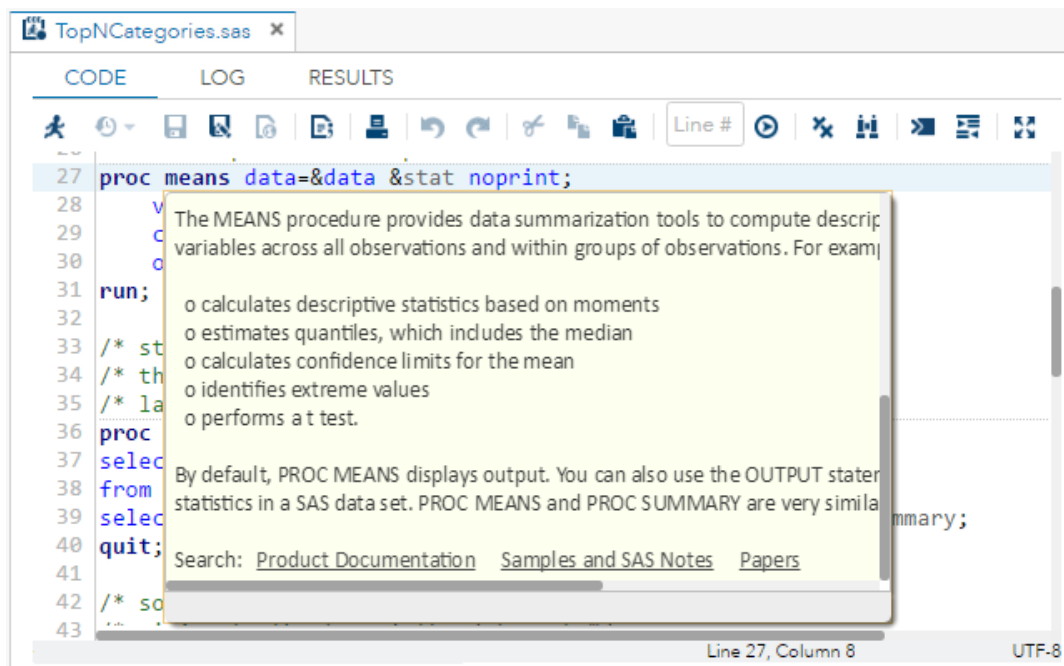
- Right-click a keyword in your program and select **Syntax Help**.
- Start entering a valid SAS keyword, and then click a suggested keyword in the autocomplete window.
- Position the mouse pointer over a valid SAS keyword in your program. This works only if you have selected the **Enable hint** option in the Editor preferences. For more information, see “Customizing the Code Editor” on page 38.

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*y + x*x));

```

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

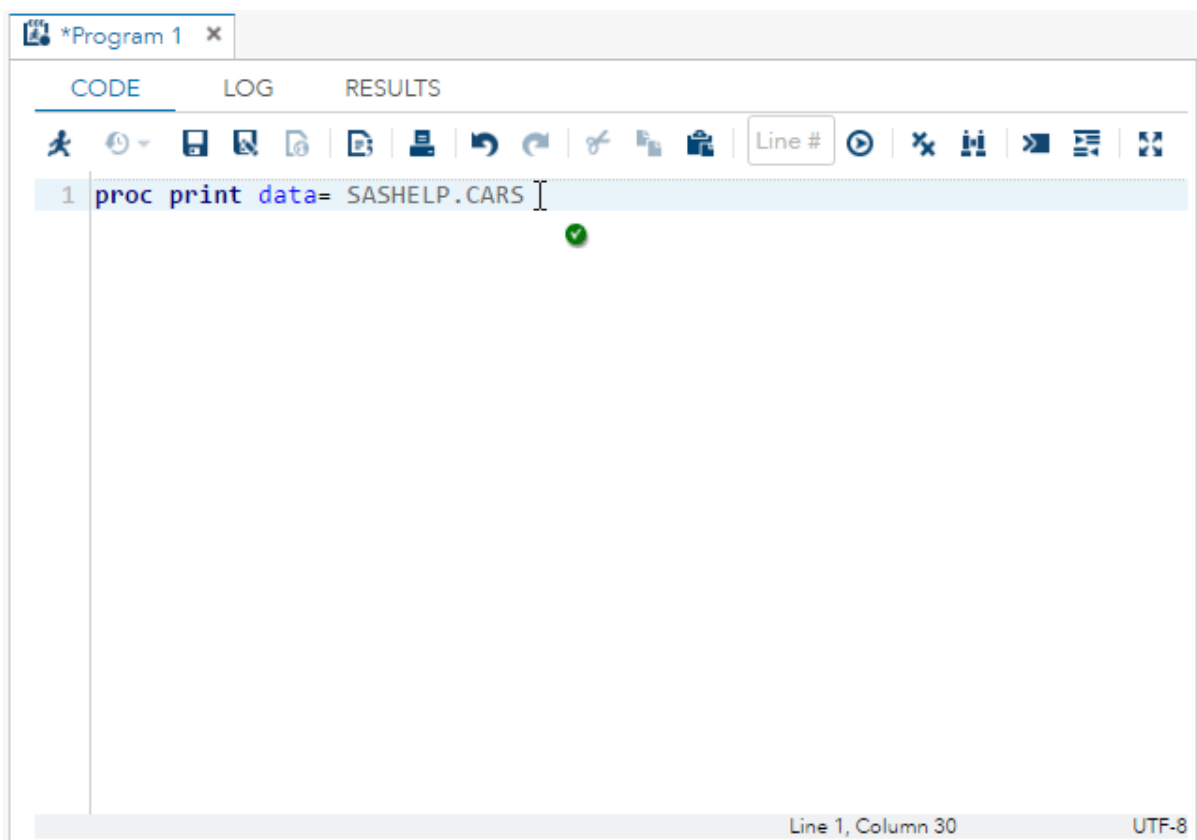
You do not have to select entire horizontal lines of text. You can select columns or vertical blocks of text.

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.

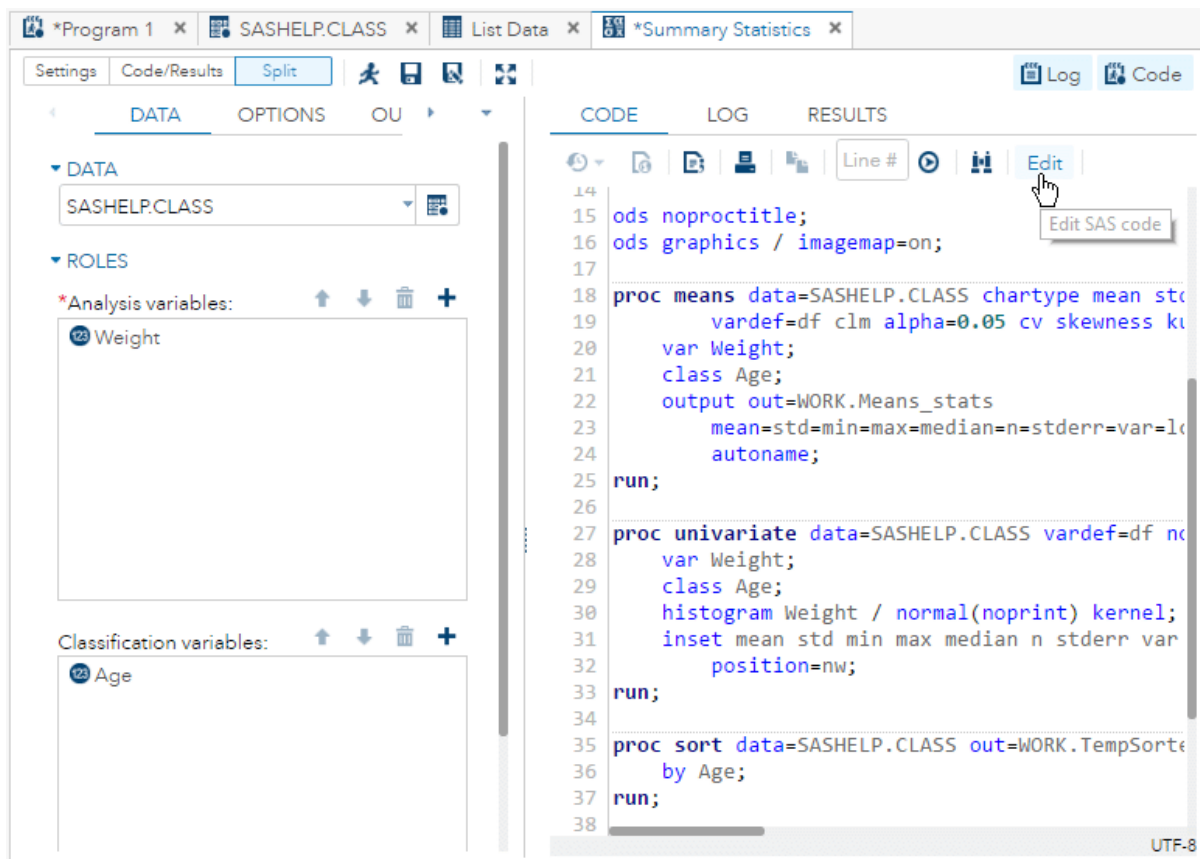
To edit a program generated by a task:

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  and select

Preferences. Click **Tasks**, and then 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 "&".

For example, you can use macros in a footnote like the following:

```
Generated with &_CLIENTAPP &_CLIENTAPPVERSION
```

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*.


Macro Variable	Description
_CLIENTAPP	name of the application.
_CLIENTAPPVERSION	application version number.

Macro Variable	Description
<code>_SASPROGRAMFILE</code>	the full path and filename 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.
<code>SYSPROCESSNAME</code>	name of the current SAS process.
<code>SYSPROCESSMODE</code>	current SAS session run mode or server type name.
<code>SYSVLONG4</code>	SAS software release number, maintenance level, and four-digit year.

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  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 .

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

```
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;
```

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 the first maintenance release for SAS 9.4 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 14.



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  and select

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.

Snippet Name	Description
Catalogs	
Edit a SOURCE Entry	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.
List Catalog Entries	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.
List Catalogs	creates and displays a SAS table listing all of the categories in a specified library. By default, the table is saved as Work.catalogs.
Print GRSEG Entry	prints the GRSEG entry to the open destinations specified by the SAS Output Delivery System (ODS output).
Viya Cloud Analytic Services	
<i>Note:</i> These snippets are available only if your site licenses SAS/CONNECT. To run these snippets, you must establish a connection to a SAS Viya environment.	
Connect to Viya	establishes a connection to SAS Viya and defines macros for importing SAS ODS results from SAS Viya.
Create CAS Engine Libref	creates a CAS library for the specified hostname and port.
Upload CAS Table	uploads a data set from SAS 9.4 to SAS Viya.
List CAS Session Information	lists all the CAS sessions and session properties.
List caslibs	lists all the caslibs in the current CAS session.
List CAS Tables	lists the CAS tables in the specified caslib.
List CAS Table Contents	lists the contents of the specified CAS table.
Run Machine Learning Procedure	runs a Machine Learning procedure with the Sashelp.Cars CAS table.
Download CAS Table	downloads the specified CAS table to a SAS 9.4 data set.

Snippet Name	Description
Disconnect from Viya	disconnects from SAS Viya.
Data	
DS2 Code	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 allows SQL preprocessing of input tables, which effectively combines the power of the two languages. For more information, see <i>SAS DS2 Language Reference</i> .
DS2 Package	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 <i>SAS DS2 Language Reference</i> .
DS2 Thread	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 <i>SAS DS2 Language Reference</i> .
Generate CSV File	exports SAS data as a comma-separated text file.
Generate PowerPoint Slide	streams Microsoft PowerPoint output to your web browser.
Generate XML File	exports SAS data as an XML file that you can view in your web browser.
Import CSV File	imports a comma-separated file and write the output to a SAS data set.
Import XLSX File	imports a Microsoft XLSX file and write the output to a SAS data set.
Simulate Linear Regression Data	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.

Snippet Name	Description
Simulate One-Way ANOVA Data	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.
Descriptive	
Custom ODS Output	provides a template for creating HTML, PDF, and RTF output by using the SAS Output Delivery System. For more information, see <i>SAS Output Delivery System: User's Guide</i> .
PROC SQL	provides a template for writing SQL queries. For more information, see <i>SAS SQL Procedure User's Guide</i> .
Graph	
<i>Note:</i> For more information about the SGPLOT, SGPANEL, and SGSCATTER procedures, see <i>SAS ODS Graphics: Procedures Guide</i> .	
Bar Panel	uses the VBAR statement in the SGPANEL procedure and enables you to create multiple bar charts.
Box Panel	uses the VBOX statement in the SGPANEL procedure and enables you to create multiple box plots.
Comparative Scatter Plot	uses the COMPARE statement in the SGSCATTER procedure. This code snippet creates a comparative panel of scatter plots with shared axes.
Dot Plot	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.
Fit Plot	uses the REG statement in the SGPLOT procedure. This code snippet produces a regression plot with a quadratic fit and includes confidence limits.
HBar Plot	uses the HBAR statement in the SGPLOT procedure. This code snippet creates a horizontal bar chart that summarizes the values of a category variable.
HighLow Plot	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.
Histogram Plot	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.

Snippet Name	Description
Scatter Plot Matrix	uses the MATRIX statement in the SGSCATTER procedure. This code snippet creates a scatter plot matrix.
VBox Plot	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.
IML	
<i>Note:</i> These snippets are available only if your site licenses SAS/IML.	
Find Roots of Nonlinear Equation	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.
Fit by using Maximum Likelihood	uses maximum likelihood estimation to estimate parameters for the normal density estimate.
Generate a Bootstrap Distribution	uses the IML procedure to create and analyze a bootstrap distribution of the sample mean.
Integrate a Function	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.
Simulate Multivariate Normal Data	simulates data from a multivariate normal distribution with a specified mean and covariance.
Macro	
<i>Note:</i> For more information about SAS macros, see <i>SAS Macro Language: Reference</i> .	
SAS Macro	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. For more information, see <i>SAS Macro Language: Reference</i> .


Snippet Name	Description
SAS Macro Variables	<p>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.</p> <p>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.</p>
SAS Macro Do Statement	<p>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.</p> <p>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.</p> <p><i>Note:</i> SAS also provides a %DO iterative statement, which is different from the code that is generated by this snippet. For more information, see <i>SAS Macro Language: Reference</i>.</p>
SAS Macro If Statement	<p>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.</p>
SAS Macro Parameters	<p>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.</p>

Snippet Name	Description
SAS Macro Quoting	<p data-bbox="788 235 1396 327">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.</p> <ul data-bbox="788 338 1396 1564" style="list-style-type: none"> <li data-bbox="788 338 1396 520">• 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 %: ' " () <li data-bbox="788 632 1396 814">• 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: & % <li data-bbox="788 825 1396 1008">• 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 <li data-bbox="788 1018 1396 1201">• 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 <li data-bbox="788 1211 1396 1266">• The %QSCAN function searches for a word and masks special characters and mnemonic operators. <li data-bbox="788 1276 1396 1331">• The %QSUBSTR function produces a substring and masks special characters and mnemonic operators. <li data-bbox="788 1341 1396 1434">• The %QUPCASE function converts a value to uppercase and returns a result that masks special characters and mnemonic operators. <li data-bbox="788 1444 1396 1564">• The %UNQUOTE function unmask a value during macro execution so that any special characters and mnemonic operators are interpreted as macro language elements instead of text. <p data-bbox="788 1575 1396 1656">For more information about macro compilation and macro execution, see <i>SAS Macro Language: Reference</i>.</p>

Snippet Name	Description
SAS Macro Char Functions	<p>provides several examples of these SAS macros that work with character values:</p> <ul style="list-style-type: none"> • 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. <p>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.</p> <ul style="list-style-type: none"> • 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.

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 . 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**.

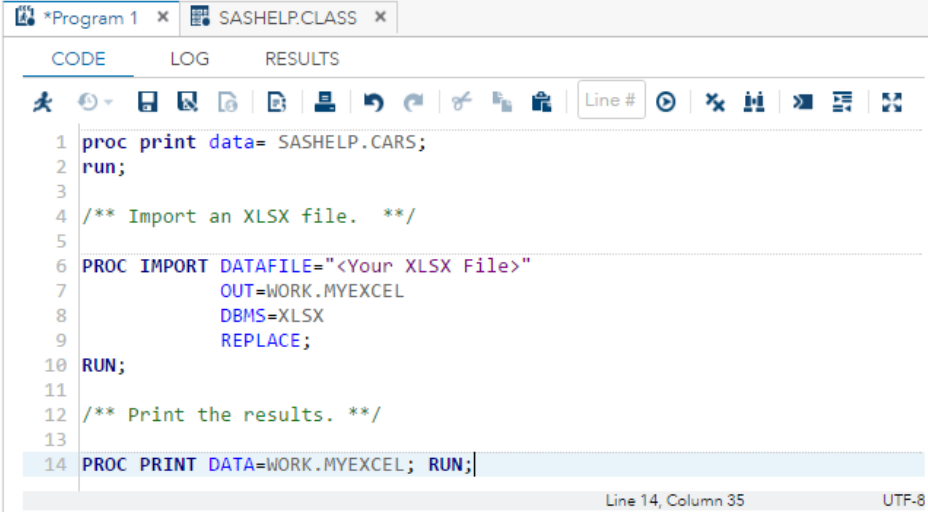
This snippet is now available from the **My Snippets** folder.

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 move 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.



The screenshot shows the SAS code editor interface. The title bar indicates two open windows: '*Program 1' and 'SASHELP.CLASS'. The editor has tabs for 'CODE', 'LOG', and 'RESULTS'. A toolbar at the top contains various icons for file operations and editing. The code is as follows:

```

1 proc print data= SASHELP.CARS;
2 run;
3
4 /** Import an XLSX file. **/
5
6 PROC IMPORT DATAFILE="<Your XLSX File>"
7             OUT=WORK.MYEXCEL
8             DBMS=XLSX
9             REPLACE;
10 RUN;
11
12 /** Print the results. **/
13
14 PROC PRINT DATA=WORK.MYEXCEL; RUN;

```

The status bar at the bottom right shows 'Line 14, Column 35' and 'UTF-8'.

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 **Editor**.

For more information, see “[Setting the Editor Preferences](#)” on page 108.

Chapter 3

Working with Queries

What Is a Query?	39
Creating a Query	40
Creating a New Query	40
Adding Tables to a Query	40
Understanding Joins	41
Joining Tables	41
Creating a Join	41
Understanding the Types of Joins	43
Modifying an Existing Join	44
Selecting Data	44
Specifying Columns in the Output	44
Using Summary Functions	46
Filtering Data	48
Creating a Filter	48
Changing the Relationship between Filters	50
Managing Output	51
Sorting Your Output	51
Eliminating Duplicate Rows in Output	52
Grouping Your Output	52
Saving Your Results	55
Running a Query	56

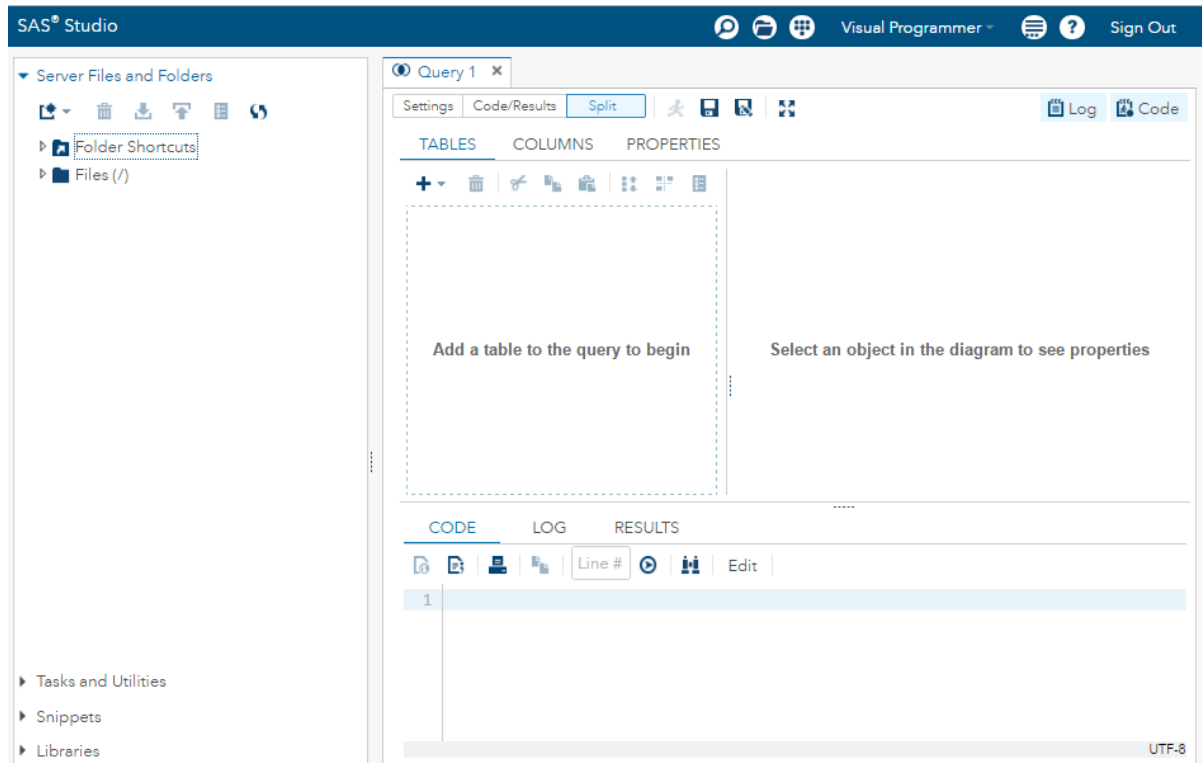
What Is a Query?

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 columns that have the same name and type. If no matches for column name and type are found, then you can specify the join criteria.

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:

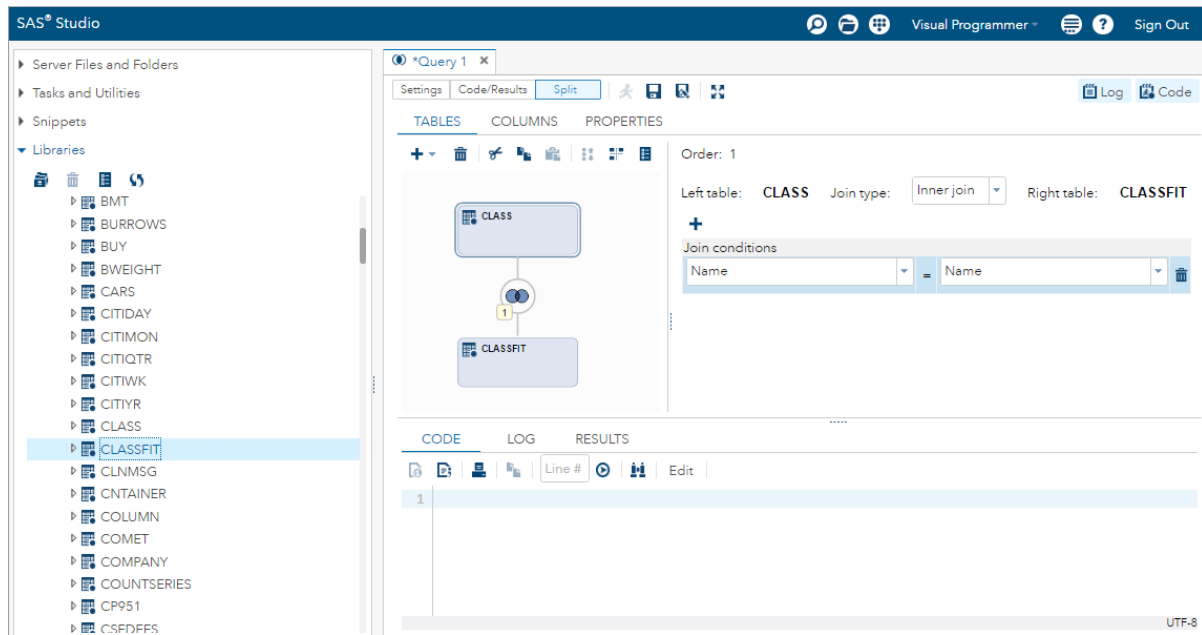
From the **Libraries** of the navigation pane, drag the table that you want to add to the query to the **Tables** tab. Next, drop that table on top of the first table in the query to join the two.

The screenshot shows the SAS Studio interface. On the left, the 'Libraries' pane lists various tables, with 'CLASSFIT' highlighted. The main workspace shows a query window with a 'TABLES' tab. A diagram indicates that the 'CLASSFIT' table is being joined to the 'CLASS' table. A table properties window is open, displaying the following columns:

Name	Type	Length	Format	Informat	Label
Name	Char	8			
Sex	Char	1			
Age	Numeric	8			
Height	Numeric	8			
Weight	Numeric	8			

The 'CODE' tab at the bottom shows a single line of code with a line number '1'.

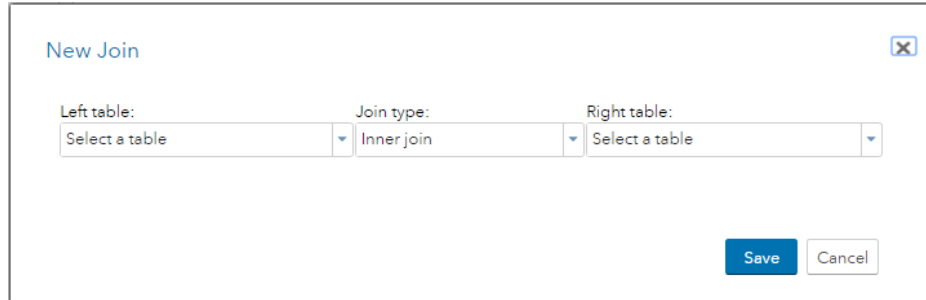
The Join window displays the join criteria. In the following example, the Classfit table is automatically joined to the Class table by using the Name column in both tables.



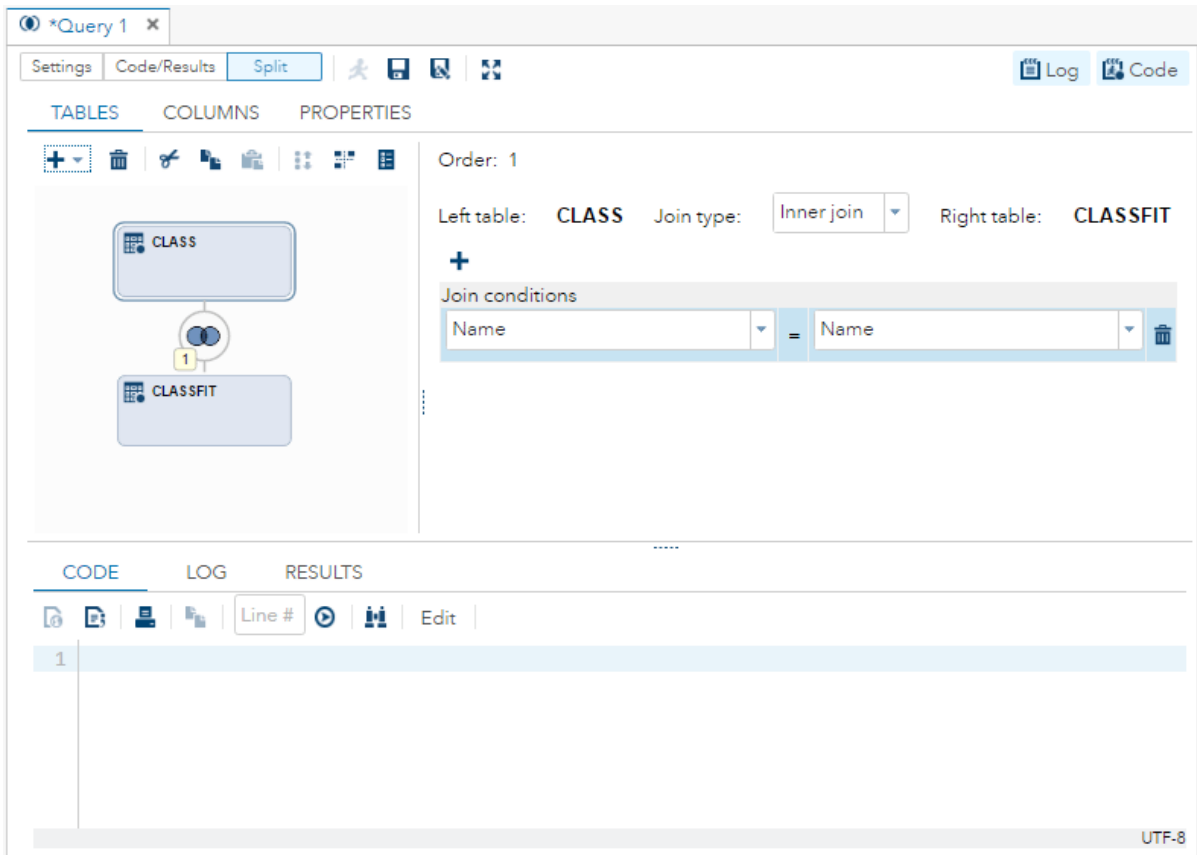
If a join cannot be created automatically, you can specify the join condition manually.

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.







- 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.


SAS Studio Join Type	Join Icon	Description
Inner Join		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.
Left Join		The output rows include all rows from the first table and the rows from the second table in which the joining criterion is met.

SAS Studio Join Type	Join Icon	Description
Right Join		The output rows include all rows from the second table and the rows from the first table in which the joining criterion is met.
Full Join		The output rows include all matching and nonmatching rows from both tables.

Modifying an Existing Join

You can modify an existing join by selecting a different type of join or by changing the columns that are used in the join condition. You can also add and remove join conditions or remove the entire join.

To modify a join:

1. On the **Tables** tab of the query window, click the join indicator that you want to modify. The join is displayed in the **Join** area.
2. To change the type of join, select a new type from the **Join type** drop-down list.
3. To add a new join condition, click **+** and specify the columns to use in the join. To remove a join condition, click  next to the appropriate condition.

To delete the entire join, right-click the join indicator and select **Delete**.

Selecting Data

Specifying Columns in the Output

By default, no columns are included in the output. You must specify the columns that you want to appear in the output table. You can also specify an alias to use in place of the column name in the output table.

The order in which the columns are listed on the **Select** is the order in which they appear in the output table.

To select columns for the output table:

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 output data by dragging them from the columns list to the **Select** tab. You can also click **+** on the **Select** tab toolbar and select one or more columns from the Choose Column window.

The screenshot shows the SAS Query Builder interface for a query named '*Query 1'. The 'COLUMNS' tab is active, displaying a list of columns from two tables: CLASS and CLASSFIT. The 'CLASSFIT.predict' column is selected. Below the column list, the 'SELECT' tab is active, showing a table with columns: Table, Source Column, Column Name, and Summary. The table contains three rows: CLASS.Name, CLASS.Age, and CLASSFIT.predict. The 'Column Name' for CLASSFIT.predict is 'predict'. The 'CODE' tab at the bottom shows the following SQL code:

```

4
5 PROC SQL;
6 CREATE TABLE WORK.QUERY
7 AS
8 SELECT CLASS.Name, CLASS.Age, CLASSFIT.predict
9 FROM SASHELP.CLASS CLASS
10 INNER JOIN SASHELP.CLASSFIT CLASSFIT
11 ON

```

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.

The screenshot shows the SAS Query Builder interface for a query named '*Query 1'. The 'COLUMNS' tab is selected, showing a tree view of columns from 'CLASS' and 'CLASSFIT' tables. The 'CLASS' table has columns: Name, Sex, Age, Height, Weight. The 'CLASSFIT' table has columns: Name, Sex, Age, Height, Weight, predict. The 'predict' column is selected. Below the tree view, the 'SELECT' tab is active, showing a table with columns: Table, Source Column, Column Name, and Summary. The 'Summary' column has a dropdown menu. The 'CODE' tab is also visible, showing the following SQL code:

```

4
5 PROC SQL;
6 CREATE TABLE WORK.QUERY
7 AS
8 SELECT CLASS.Name, CLASS.Age, CLASSFIT.predict
9 FROM SASHELP.CLASS CLASS
10 INNER JOIN SASHELP.CLASSFIT CLASSFIT
11 ON

```

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:

The screenshot shows the SAS Studio interface for a query named '*Query 1'. The 'COLUMNS' tab is active, displaying a table definition for 'CLASS'. The table has columns: Name, Sex, Age, Height, and Weight. The 'Age' column is selected, and its summary is set to 'AVG'. The 'RESULTS' tab is also visible, showing the following SQL code:

```
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
```

Additional details from the interface include a 'View: Column names' dropdown, a 'Select distinct rows only' checkbox, and a 'Log' button. The bottom right corner of the window displays 'UTF-8'.

By default, the query displays the results in the **Output Data** tab and generates an output table in the Work library:

The screenshot shows the SAS Studio Query Editor interface. The top toolbar includes 'Settings', 'Code/Results', 'Split', and 'Log/Code' buttons. The 'COLUMN NAMES' tab is active, displaying a table with the following data:

Table	Source Column	Column Name	Summary
CLASS	Age	Age	AVG

The 'OUTPUT DATA' tab is selected, showing a table with one row:

Age
13.315789474

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 52](#).

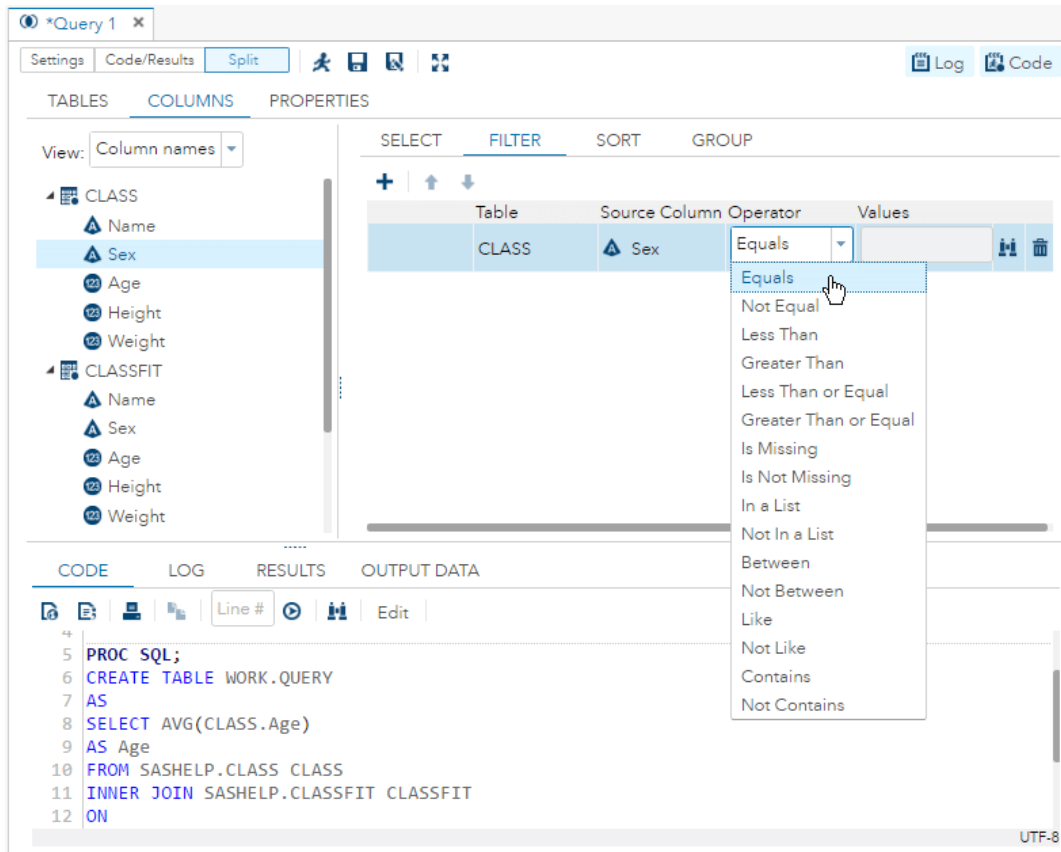
Filtering Data



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 **+** on the **Filter** tab toolbar and select one or more columns from the Choose Column window.

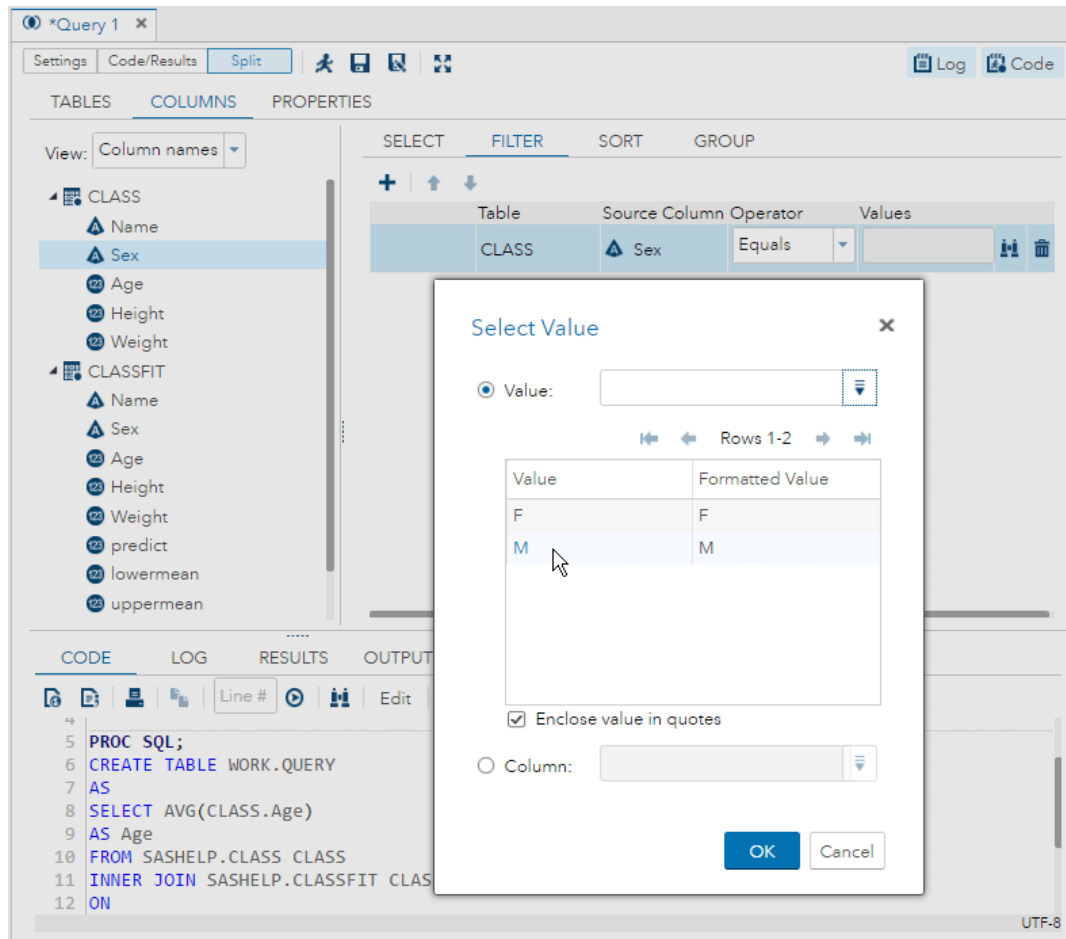
3. Select a comparison operator from the **Operator** drop-down list. The default value is **Equals**.



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.

The screenshot shows the SAS Query Builder interface. The 'FILTER' tab is active, and a filter rule is being added. The rule is: Table: CLASS, Source Column: Age, Operator: Greater Than, Values: 12. The filter rule is being added to an existing rule: Table: CLASS, Source Column: Sex, Operator: Equals, Values: M. The filter rule is being added using the 'AND' operator.

The SQL code at the bottom of the window is:

```

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
12 ON

```

Managing Output

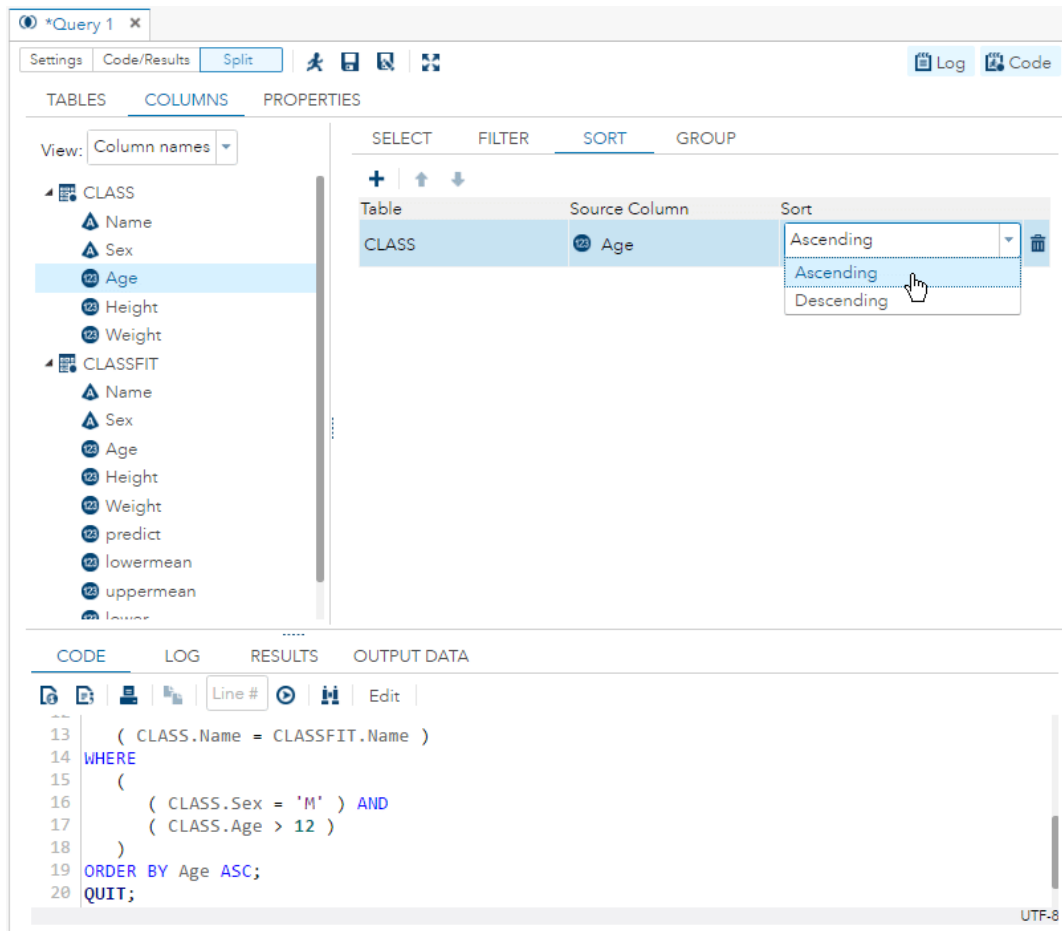
Sorting Your Output

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.

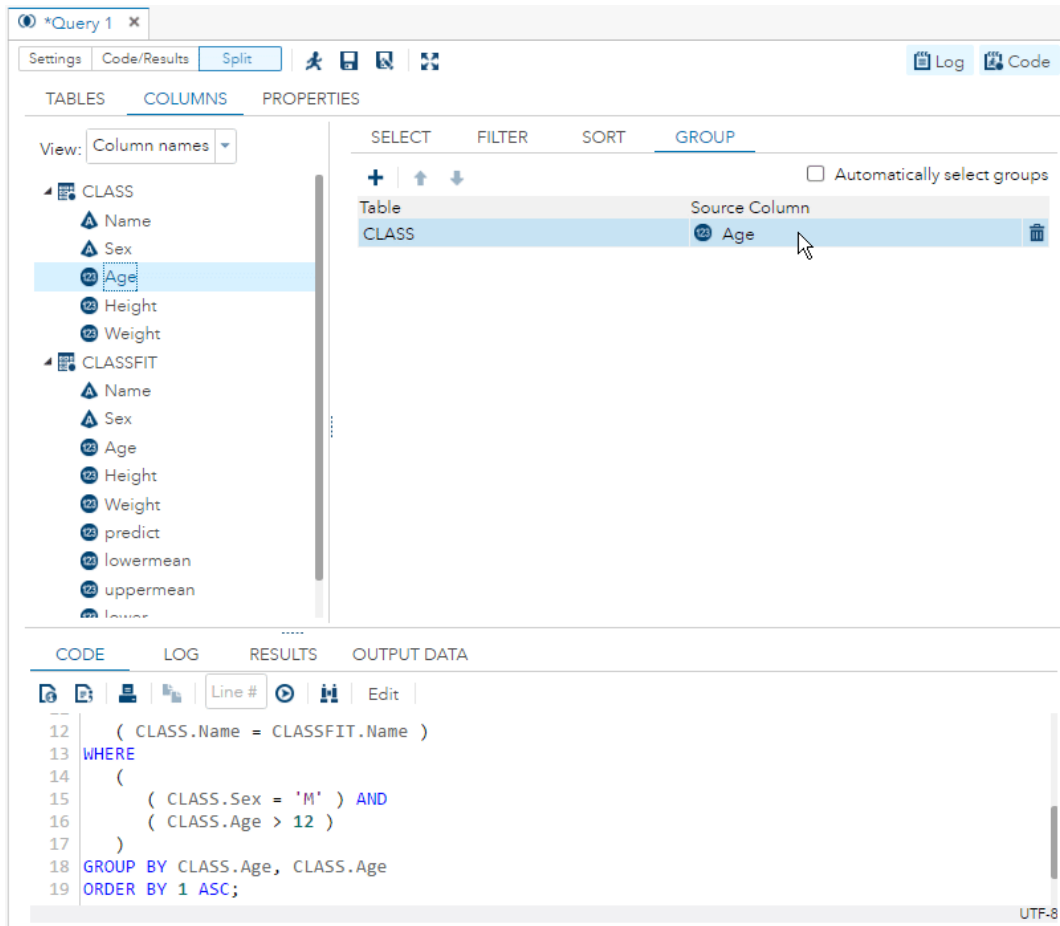
By default, the **Automatically select groups** option is selected on the **Group** tab. When this option is selected and you have performed a summary function on a column, your

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:

The screenshot shows the SAS Query Builder interface for a query named '*Query 1'. The interface is divided into several sections:

- Navigation and Settings:** Includes tabs for 'Settings', 'Code/Results', and 'Split'. There are also 'Log' and 'Code' buttons in the top right.
- Table Selection:** A tree view on the left shows tables 'CLASS' and 'CLASSFIT' with their respective columns. 'CLASS' columns include Name, Sex, Age, Height, and Weight. 'CLASSFIT' columns include Name, Sex, Age, Height, Weight, predict, lowermean, and uppermean.
- Query Configuration:** A central pane shows the 'SELECT' tab with options for 'FILTER', 'SORT', and 'GROUP'. A table lists the selected columns:

Table	Source Column	Column Name	Summary
CLASS	Age	Age	
CLASS	Weight	Weight	AVG
- Code Editor:** The bottom section shows the generated SQL code:


```

4
5 PROC SQL;
6 CREATE TABLE WORK.QUERY
7 AS
8 SELECT CLASS.Age, AVG(CLASS.Weight)
9 AS Weight
10 FROM SASHELP.CLASS CLASS
11 INNER JOIN SASHELP.CLASSFIT CLASSFIT
12 ON
      
```

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:

The screenshot shows the SAS Query Builder interface. The 'OUTPUT DATA' tab is active, displaying a table with 6 rows and 2 columns: 'Age' and 'Weight'. The 'Columns' list on the left includes 'Select all', 'Age', and 'Weight'. The 'Table' dropdown is set to 'WORK.QUERY' and the 'View' dropdown is set to 'Column names'. The 'Filter' is set to '(none)'. The 'Output Data' table shows the following data:

	Age	Weight
1	11	67.75
2	12	94.4
3	13	88.666666667
4	14	101.875
5	15	117.375
6	16	150

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 55](#).

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	57
What Is a Process Flow?	57
Creating Process Flows	58
What Are Ports?	58
Understanding the Status of Each Node	59
Customizing a Process Flow	61
Viewing the Properties of a Process Flow	61
Add Data to the Process Flow	61
Adding a SAS Program to the Process Flow	62
Create a New SAS Program	62
Add an Existing SAS Program	64
Add a Snippet	64
Adding a Query to a Process Flow	67
Create a New Query	67
Add an Existing Query to a Process Flow	67
Add a Task to a Process Flow	68
Understanding Subflows	69
What Is a Subflow?	69
Create a New Subflow	70
Add a Subflow to an Existing Process Flow	71
Linking Nodes in a Process Flow	71
Generating Code from a Process Flow	71
Running a Process Flow	71
Save a Process Flow	72

Understanding Process Flows

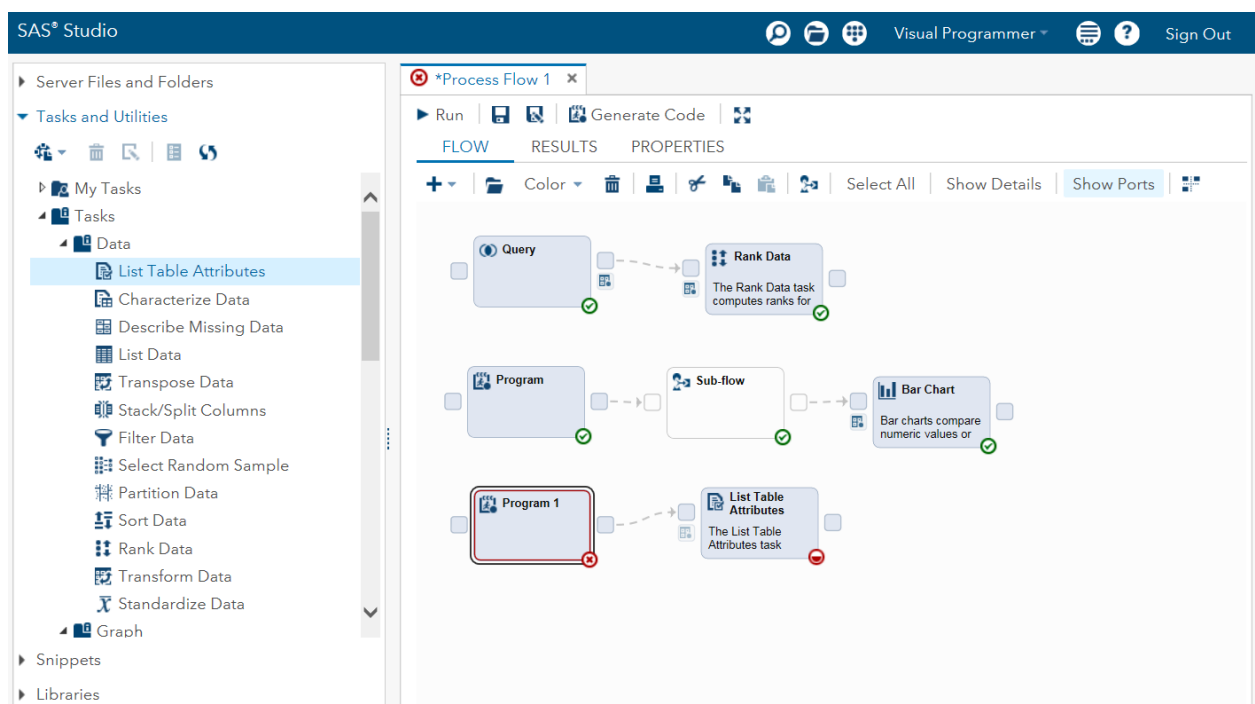
What Is a Process Flow?

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 13](#).


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  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 . 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 71](#).


data ports

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

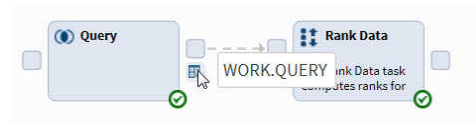
When you run a task, you must specify an input data source. You specify the input data source in the task interface. From the process flow, you can determine the name of the input data source by positioning your mouse pointer over the input data port. To view the data source on a separate tab, double-click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

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.

The screenshot shows the SAS Studio interface. On the left, the 'Tasks and Utilities' pane is expanded to 'Data', with 'List Table Attributes' selected. The main window displays the 'RESULTS' tab for 'Process Flow 1'. The results are as follows:


Name	Status	Elapsed	Output
Program	Success	0:0:0:0.287	
Query	Success	0:0:0:0.202	WORK.QUERY
Sub-flow	Success		
Rank Data	Success	0:0:0:0.202	
Bar Chart	Success	0:0:0:1.055	

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

Icon	Description
	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.
	specifies that the node is in the queue to run.
	specifies that the code for the node was submitted successfully. No warnings or errors were returned.
	specifies that the code for the node generated a warning. Review the log for that node for more information.
	specifies that the code for the node generated an error. Review the log for that node for more information.

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

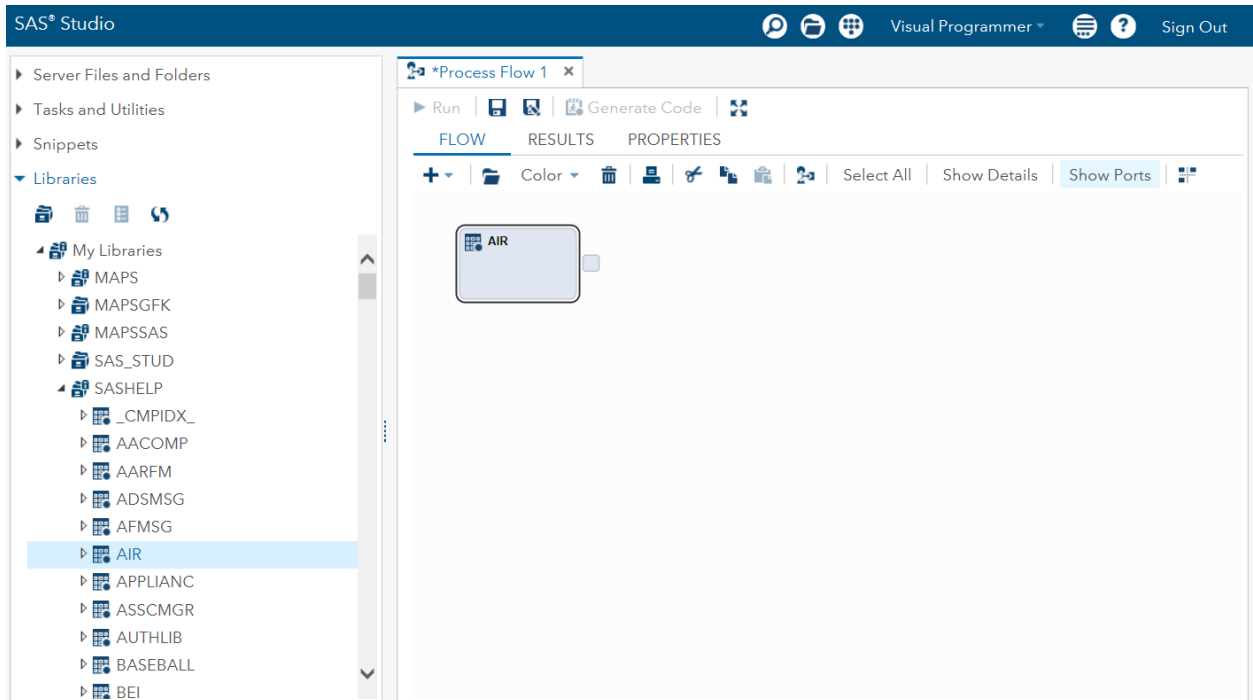
To view the properties of the current process flow, click the **Properties** tab. From the properties, you can specify the priority of execution of the nodes. The process flow runs the nodes in the order in which the nodes are added to the process flow. If node 1 is dependent on another node 2, node 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.

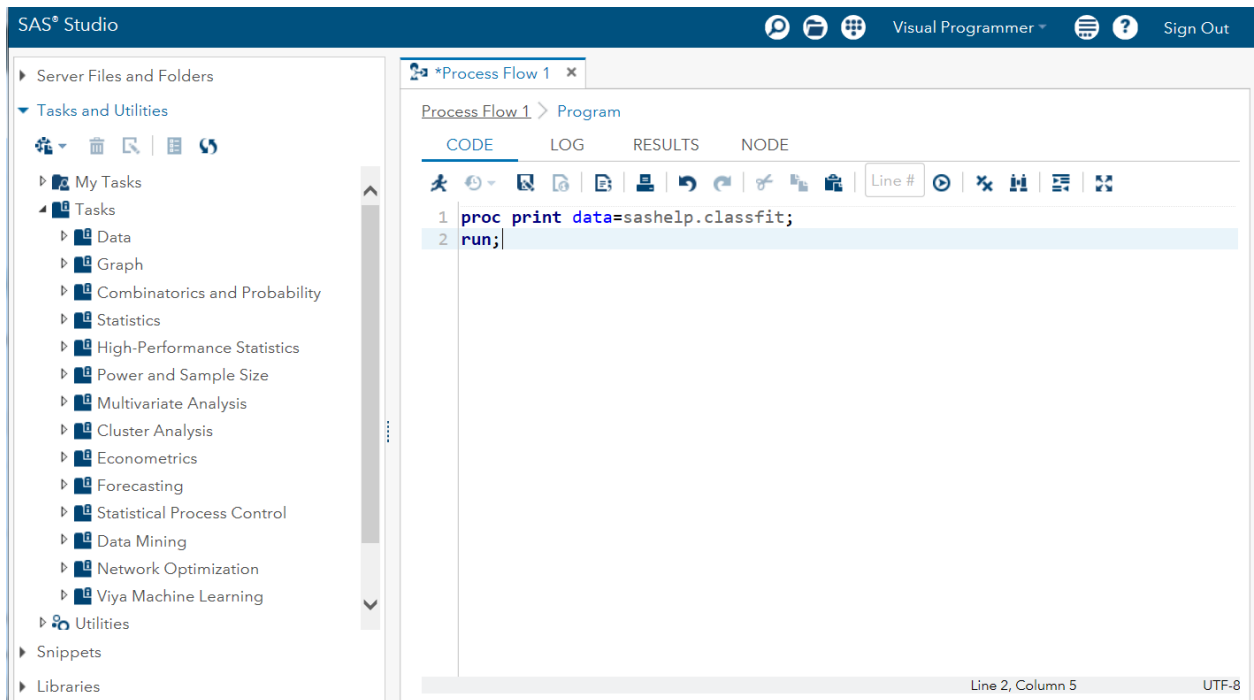


Adding a SAS Program to the Process Flow

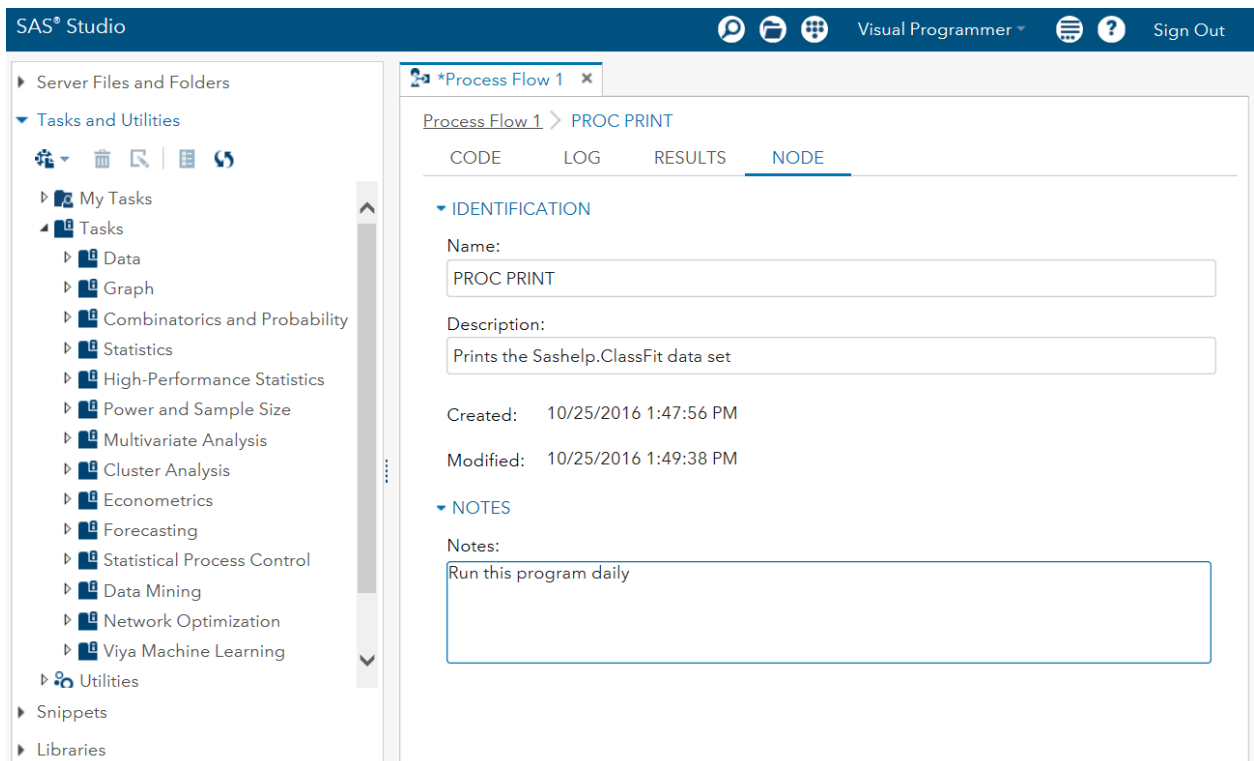
Create a New SAS Program

To add a new SAS program to a process flow:

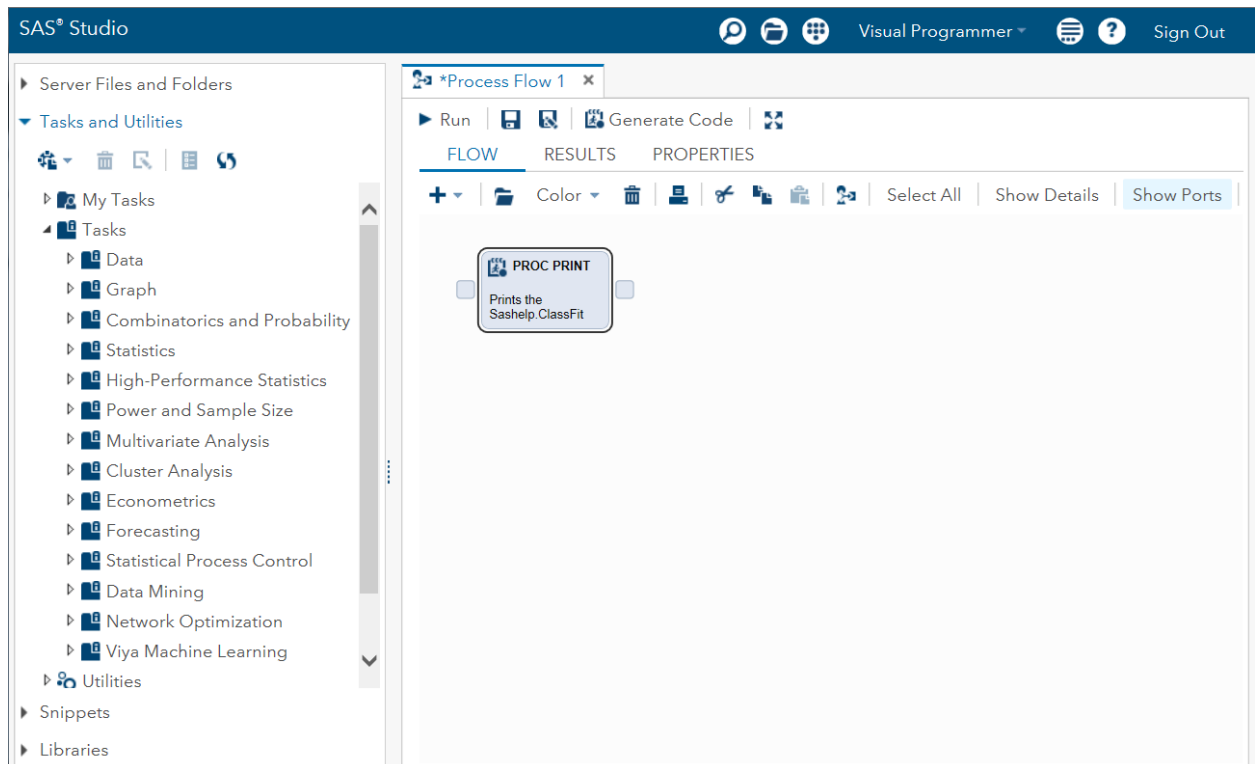
1. Click **+** and select **SAS program**. A node for the SAS program is added to the process flow.
2. Select the node and click . (You can also select the node and press Enter.) The code editor appears.
3. Enter the code for your program.



- 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.)

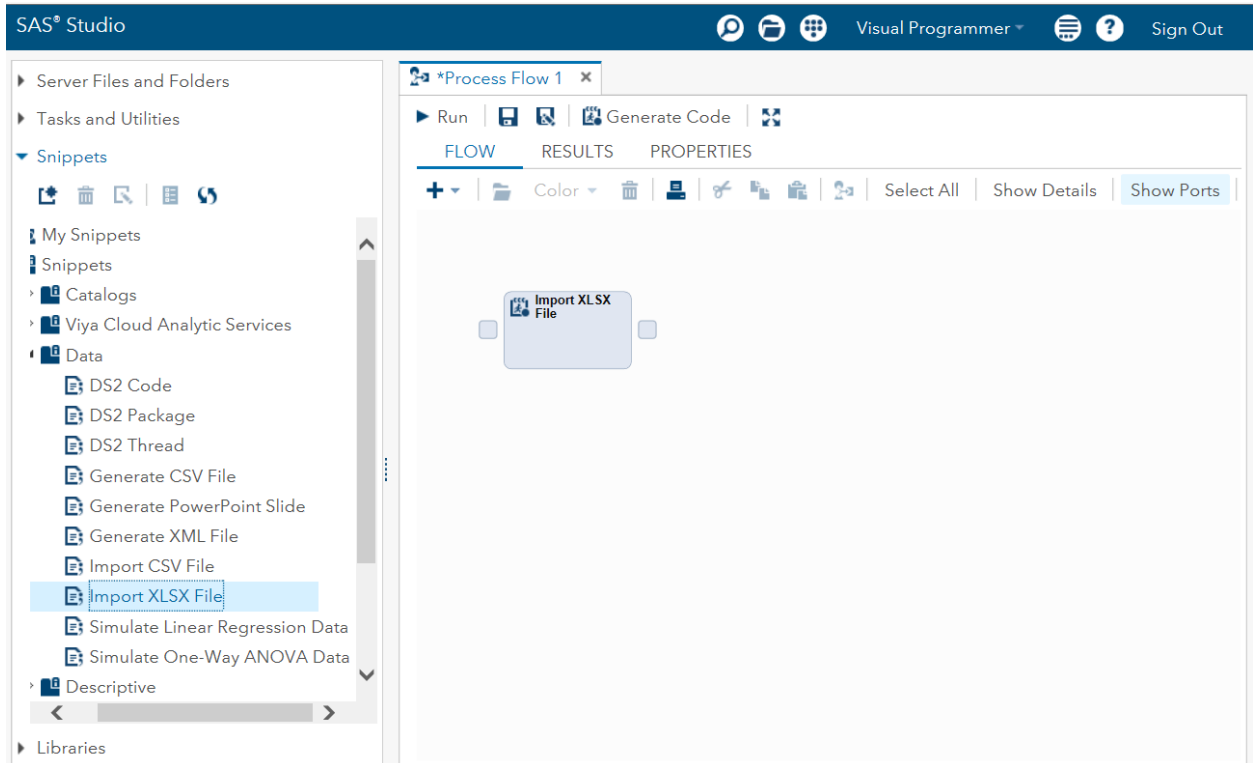
Add a Snippet

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

To add a snippet to a process flow:

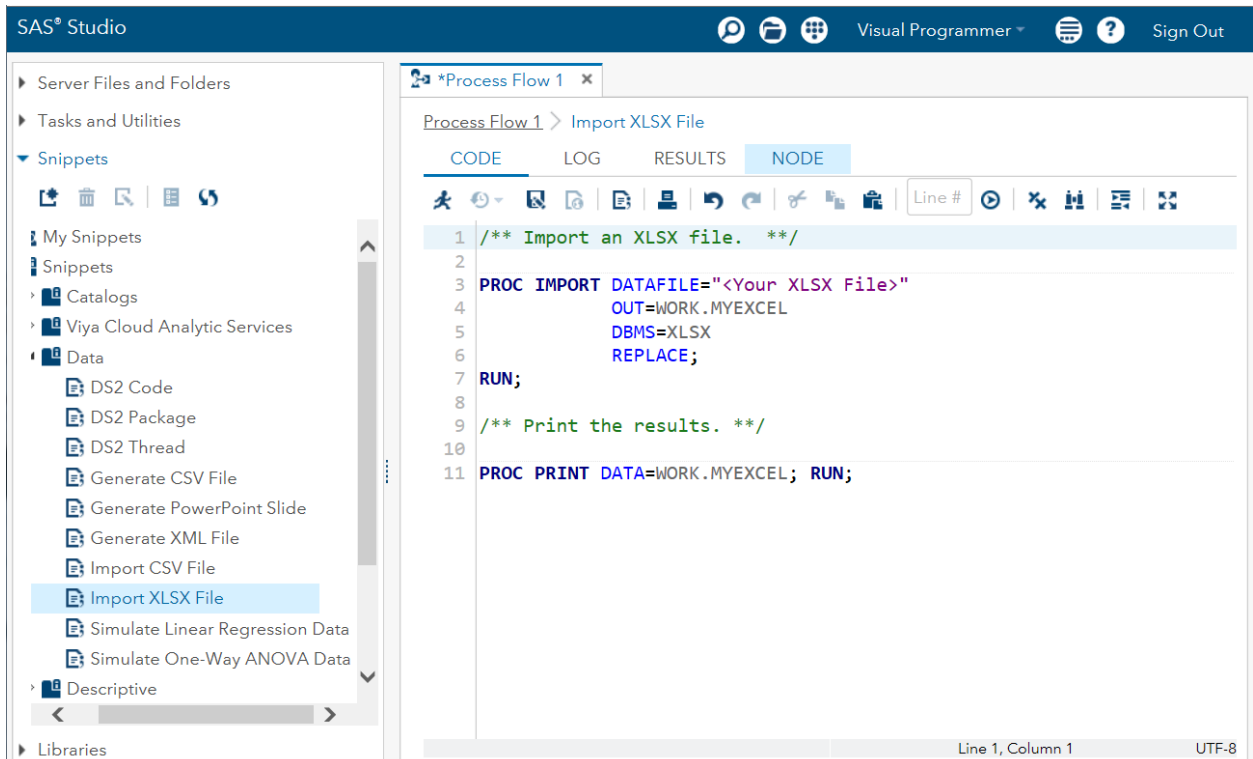
1. In the navigation pane, select **Snippets**.
2. In the **Snippets** section, select the snippet that you want to add, and then drag it to the process flow.

In this example, the Import XLSX file snippet is added to the process flow.

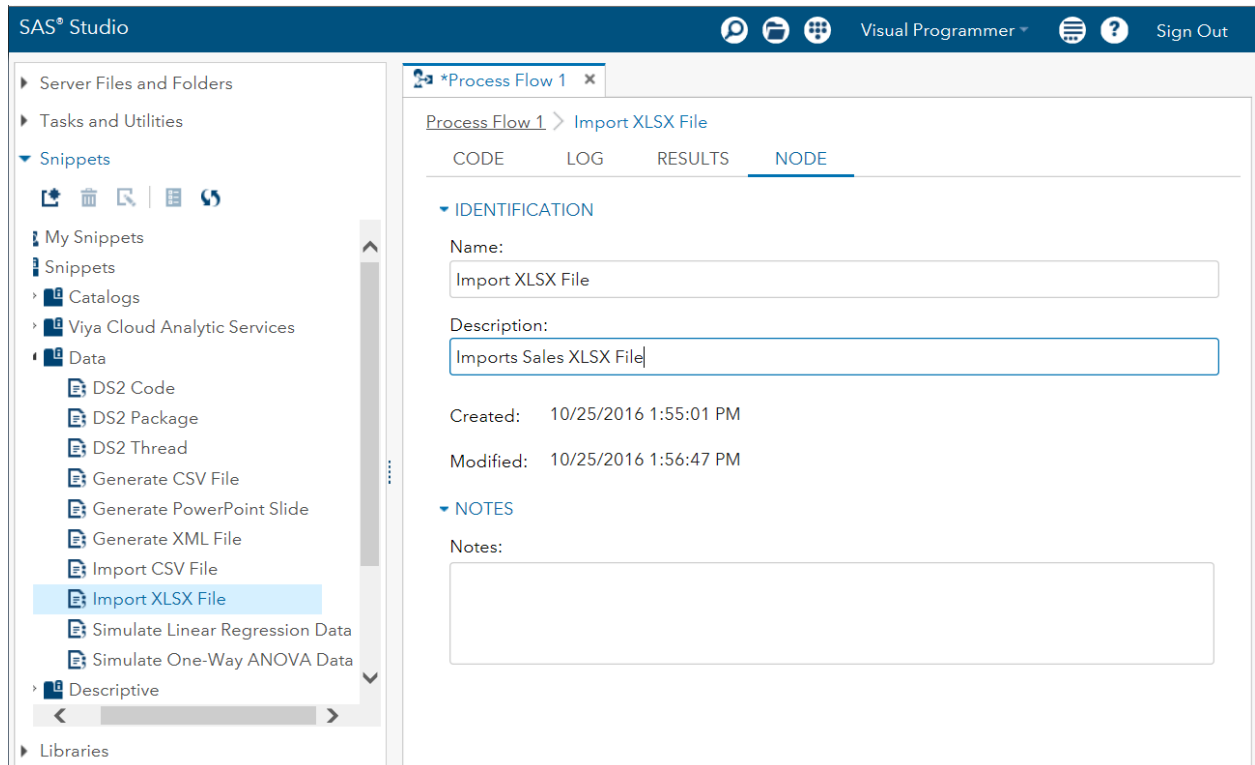


- To view the code for the snippet, select the snippet node and click . (You can also select the node and press Enter.) Edit the code to meet your needs.

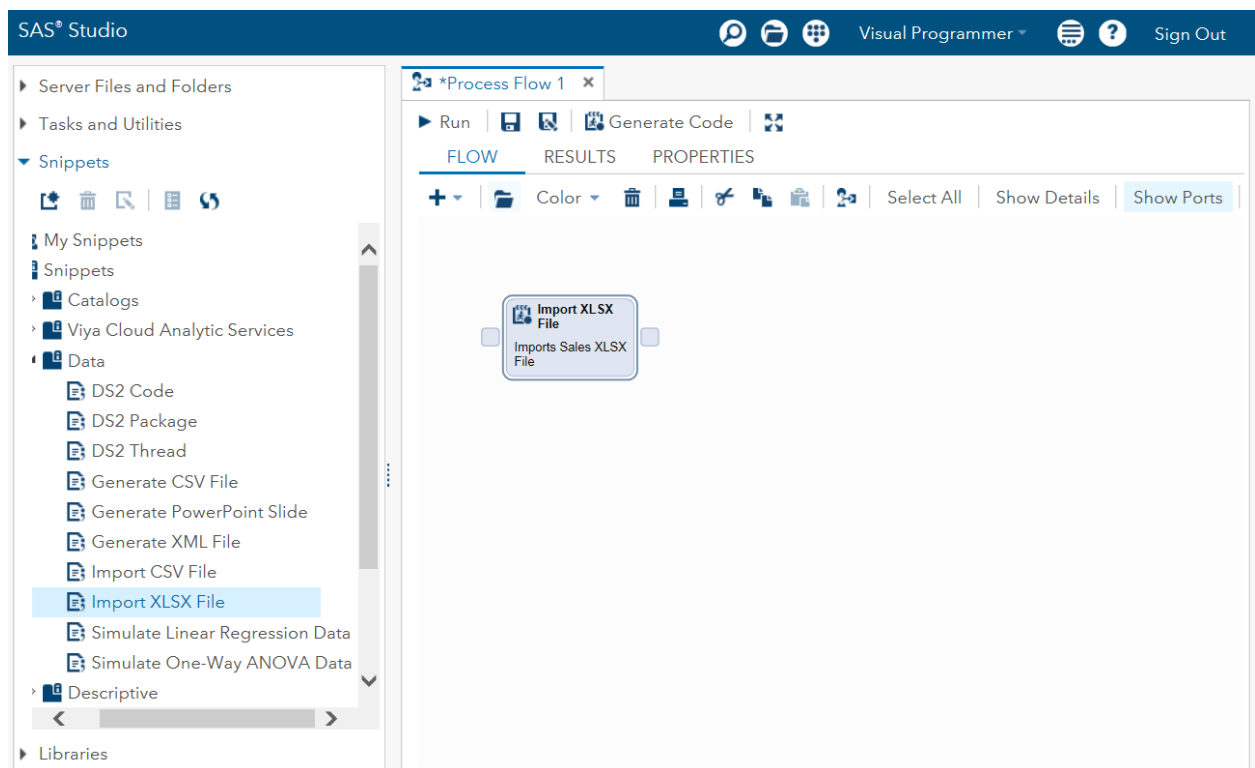
Here is the code for the Import XLSX file snippet. In this example, specify in the code the location of the XLSX file that you want to import.



- 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.



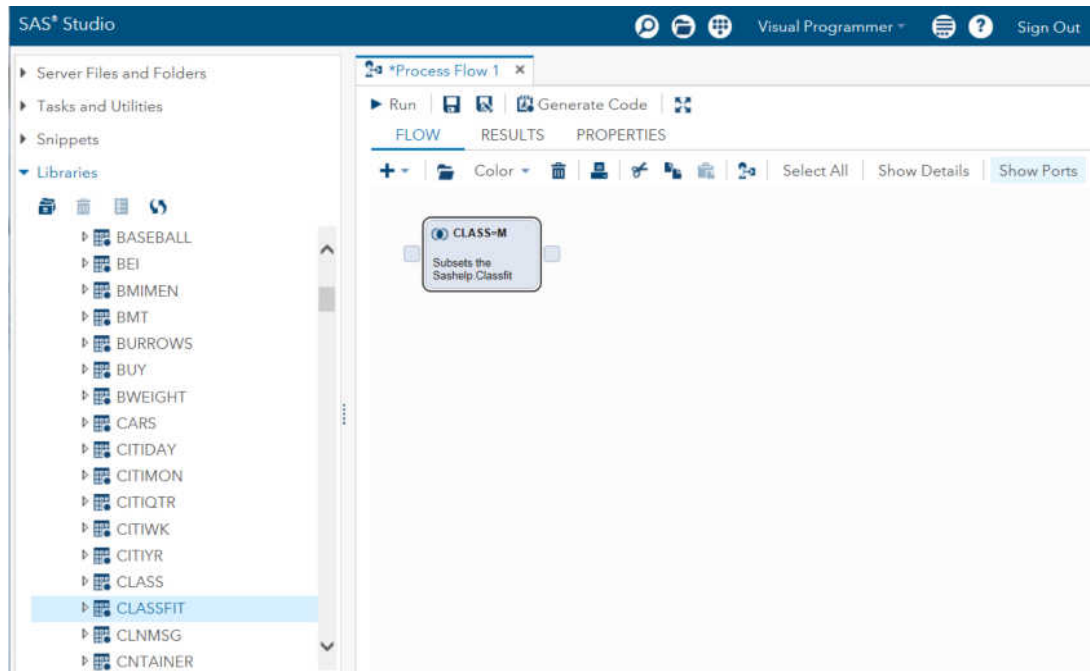
Adding a Query to a Process Flow

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 40.
4. To define the type of results for the query, use the **Properties** tab. For more information, see [“Saving Your Results”](#) on page 55.
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.

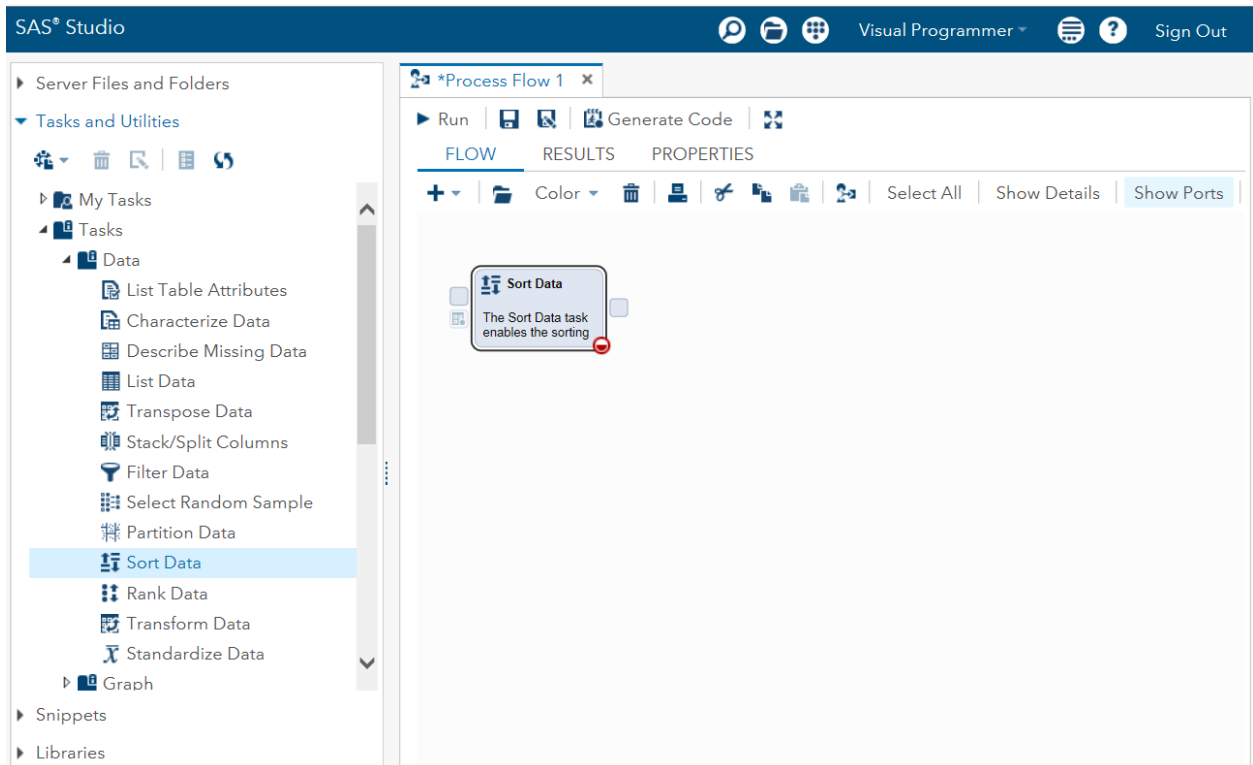
3. Select the query that you want to add, and then drag it to the process flow. (A green check mark indicates that you can add this file to the process flow.)


Add a Task to a Process Flow

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

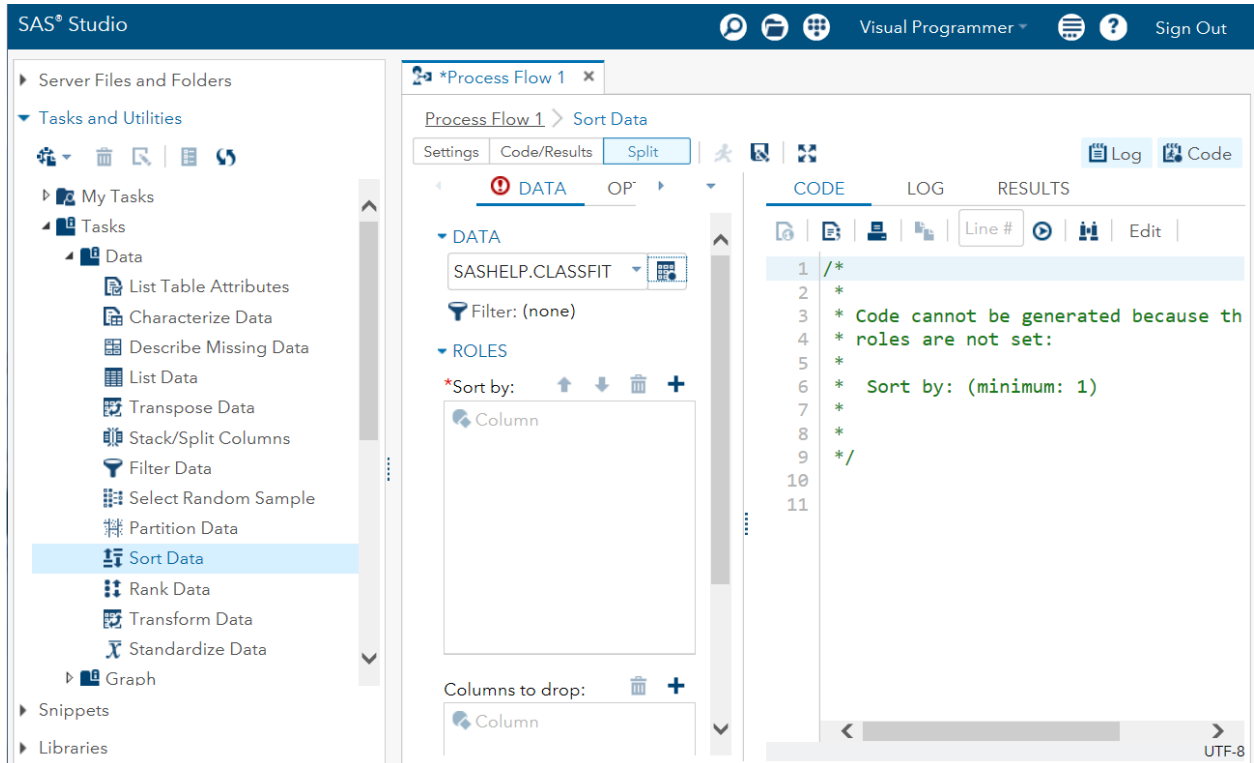
1. In the navigation pane, click the **Tasks and Utilities** section.
2. Select the task that you want to add, and then drag it to the process flow.

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



3. Select the task node and click . To run the task, you must specify values for any required options.

Here is the user interface for the Sort Data task.



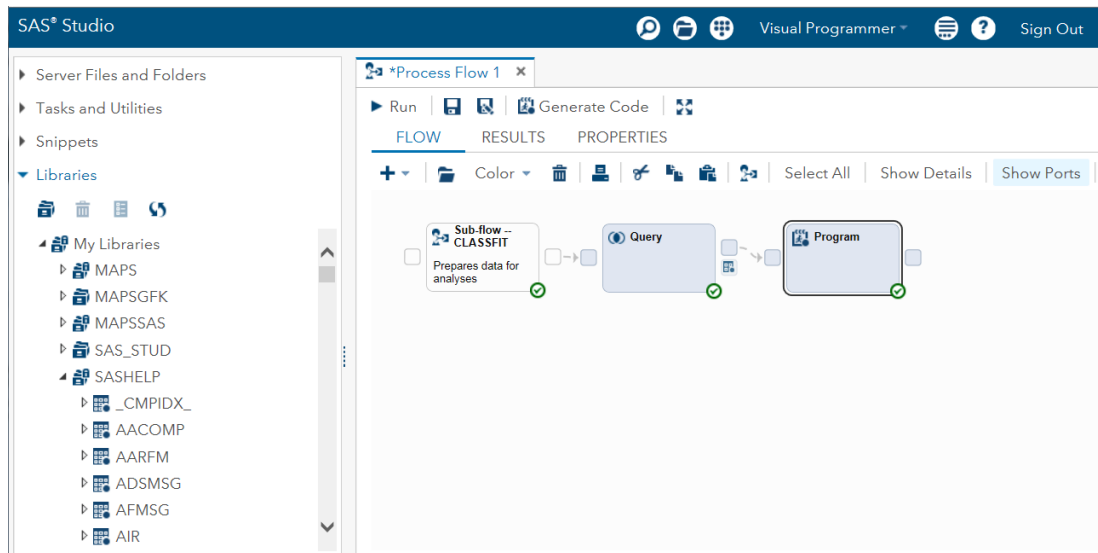
When you run the task, SAS Studio uses the values that you specified for the task options.

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.

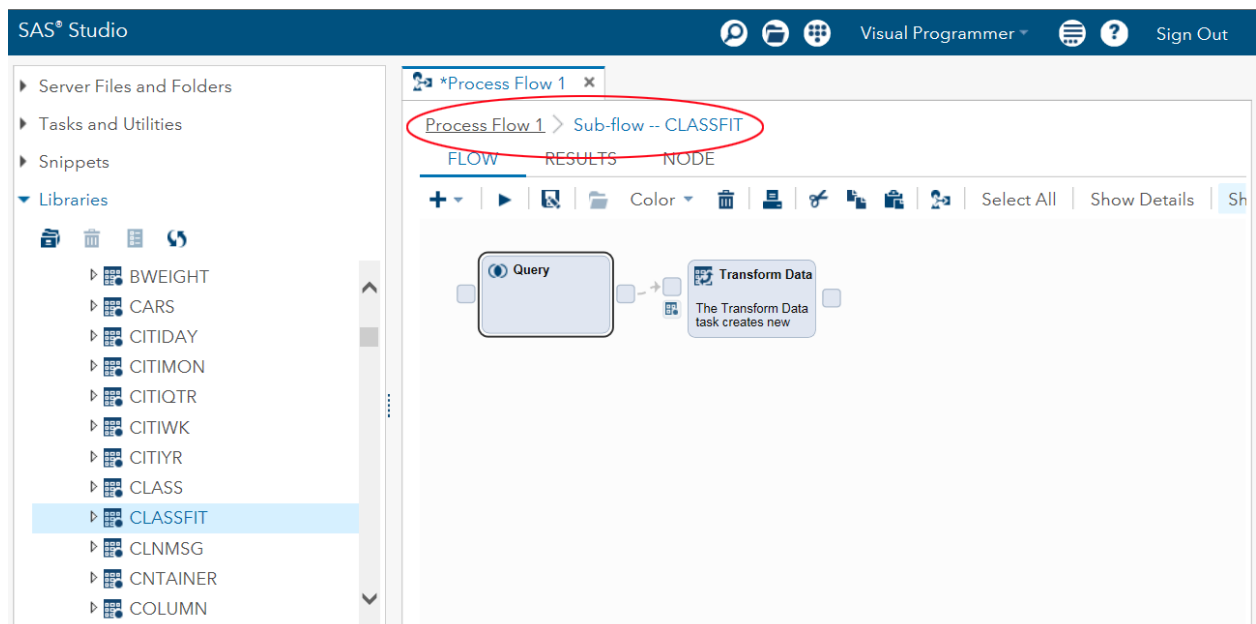


Note: SAS Studio cannot run a process flow that contains an empty subflow. An error message appears instead.

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	73
Opening and Viewing Data	75
Viewing the Code That Is Used to Create a Table	77
Filtering and Sorting Data	78
Importing Data	81
About Importing Data to SAS Studio	81
Import an Excel Worksheet	82
Import a Delimited File	85
Import a DBMS File	88
Importing Data in a Process Flow	89
Save the Import Task	89
Exporting Data	90

About the Table Viewer


When you open a table in SAS Studio, you use the table viewer.

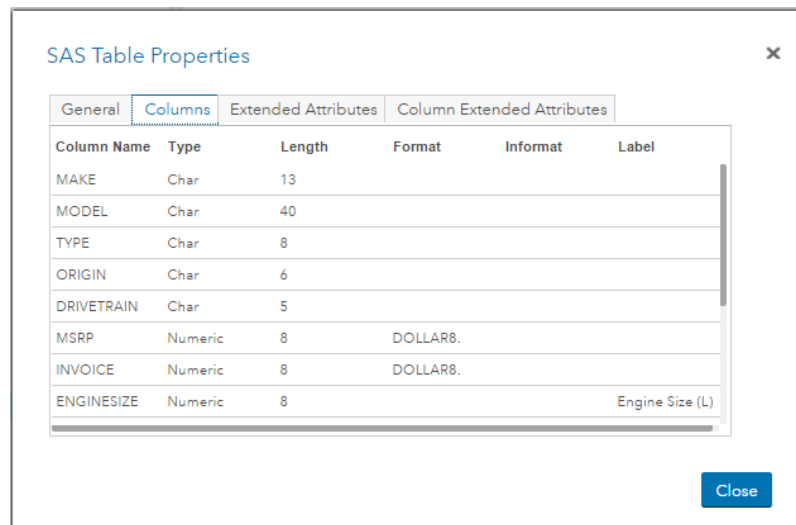
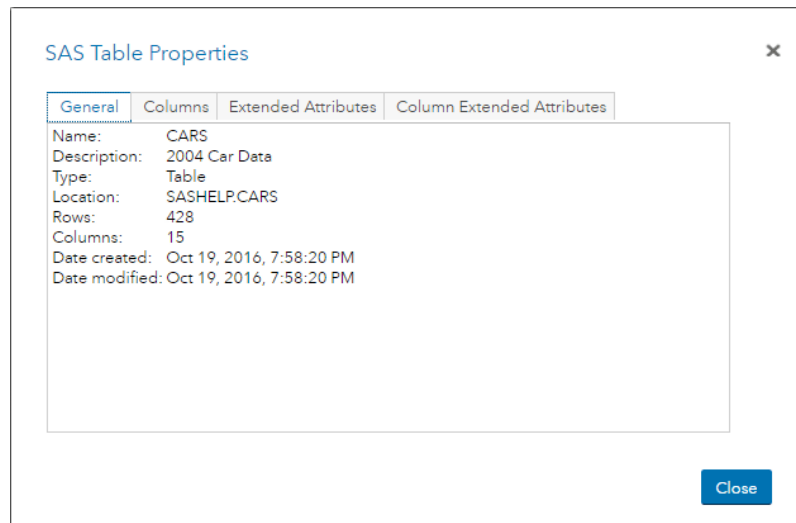
The screenshot shows the SASHELP.CARS table viewer. The 'View' dropdown is set to 'Column names'. The 'Columns' pane on the left lists various columns with checkboxes. The 'Property Value' table below it shows details for the 'Model' column. The main table displays the first 20 rows of data.

	Type	Origin	DriveTrain	MSRP	Invoice
1	SUV	Asia	All	\$36,945	\$33,337
2	Sedan	Asia	Front	\$23,820	\$21,761
3	Sedan	Asia	Front	\$26,990	\$24,647
4	Sedan	Asia	Front	\$33,195	\$30,299
5	Sedan	Asia	Front	\$43,755	\$39,014
6	Sedan	Asia	Front	\$46,100	\$41,100
7	Sports	Asia	Rear	\$89,765	\$79,978
8	Sedan	Europe	Front	\$25,940	\$23,508
9	Sedan	Europe	Front	\$35,940	\$32,506
10	Sedan	Europe	Front	\$31,840	\$28,846
11	Sedan	Europe	All	\$33,430	\$30,366
12	Sedan	Europe	All	\$34,480	\$31,388
13	Sedan	Europe	Front	\$36,640	\$33,129
14	Sedan	Europe	All	\$39,640	\$35,992
15	Sedan	Europe	Front	\$42,490	\$38,325
16	Sedan	Europe	All	\$44,240	\$40,075
17	Sedan	Europe	All	\$42,840	\$38,840
18	Sedan	Europe	All	\$49,690	\$44,936
19	Sedan	Europe	All	\$69,190	\$64,740
20	Sedan	Europe	All	\$48,040	\$43,556

Property	Value
Label	Model
Name	Model
Length	40
Type	Char
Format	
Informat	

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, all of the columns in the table are displayed. You can use the Columns area to specify which columns you want to include in the table viewer. 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.

The screenshot shows the SAS Studio interface for a table named 'SASHELP.CARS'. The 'View' dropdown menu is open, showing options: 'Column names' (selected), 'Column labels', and 'Column names'. Below the menu is a 'Columns' list with checkboxes for 'Make', 'Model', 'Type', 'Origin', 'DriveTrain', and 'MSRP'. A 'Property Value' table is also visible, showing 'Label: Model', 'Name: Model', and 'Length: 40'. The main table viewer shows 11 rows of data with columns: Type, Origin, DriveTrain, MSRP, and Invoice. The table has 428 total rows and 15 total columns.

	Type	Origin	DriveTrain	MSRP	Invoice
1	SUV	Asia	All	\$36,945	\$33,337
2	Sedan	Asia	Front	\$23,820	\$21,761
3	Sedan	Asia	Front	\$26,990	\$24,647
4	Sedan	Asia	Front	\$33,195	\$30,299
5	Sedan	Asia	Front	\$43,755	\$39,014
6	Sedan	Asia	Front	\$46,100	\$41,100
7	Sports	Asia	Rear	\$89,765	\$79,978
8	Sedan	Europe	Front	\$25,940	\$23,508
9	Sedan	Europe	Front	\$35,940	\$32,506
10	Sedan	Europe	Front	\$31,840	\$28,846
11	Sedan	Europe	All	\$33,430	\$30,366

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 105](#).

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.

SASHELP.CARS

View: Column names Filter: (none)

Columns: Total rows: 428 Total columns: 15 Rows 1-100

	Type	Origin	DriveTrain	Invoice	MSRP
1	SUV	Asia	All	\$33,337	\$36,945
2	Sedan	Asia	Front	\$21,761	\$23,820
3	Sedan	Asia	Front	\$24,647	\$26,990
4	Sedan	Asia	Front	\$30,299	\$33,195
5	Sedan	Asia	Front	\$39,014	\$43,755
6	Sedan	Asia	Front	\$41,100	\$46,100
7	Sports	Asia	Rear	\$79,978	\$89,765
8	Sedan	Europe	Front	\$23,508	\$25,940
9	Sedan	Europe	Front	\$32,506	\$35,940
10	Sedan	Europe	Front	\$28,846	\$31,840
11	Sedan	Europe	All	\$30,366	\$33,430

Property Value

Label	Model
Name	Model
Length	40

SASHELP.CARS

View: Column names Filter: (none)


Columns: Total rows: 428 Total columns: 15 Rows 1-100

	Type	Origin	DriveTrain	Invoice	MSRP
1	SUV	Asia	All	\$33,337	\$36,945
2	Sedan	Asia	Front	\$21,761	\$23,820
3	Sedan	Asia	Front	\$24,647	\$26,990
4	Sedan	Asia	Front	\$30,299	\$33,195
5	Sedan	Asia	Front	\$39,014	\$43,755
6	Sedan	Asia	Front	\$41,100	\$46,100
7	Sports	Asia	Rear	\$79,978	\$89,765
8	Sedan	Europe	Front	\$23,508	\$25,940
9	Sedan	Europe	Front	\$32,506	\$35,940
10	Sedan	Europe	Front	\$28,846	\$31,840
11	Sedan	Europe	All	\$30,366	\$33,430
12	Sedan	Europe	All	\$31,388	\$34,480
13	Sedan	Europe	Front	\$33,129	\$36,640
14	Sedan	Europe	All	\$35,992	\$39,640
15	Sedan	Europe	Front	\$38,325	\$42,490
16	Sedan	Europe	All	\$40,075	\$44,240

Property Value

Label	Model
Name	Model
Length	40
Type	Char
Format	

Viewing the Code That Is Used to Create a Table

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.

	Make	Type	Origin	DriveTrain	Invoice	MSRP
1	Acura	Sedan	Asia			\$43,755
2	Acura	Sedan	Asia			\$46,100
3	Chrysler	Sedan	USA			\$29,865
4	Chrysler	Sedan	USA			\$33,295
5	BMW	Sedan	Europe			\$30,795
6	BMW	Sedan	Europe	Rear	\$34,800	\$37,995
7	BMW	Sedan	Europe	Rear	\$26,155	\$28,495
8	BMW	Sedan	Europe	All	\$27,745	\$30,245
9	BMW	Wagon	Europe	All	\$30,110	\$32,845
10	BMW	Sedan	Europe	Rear	\$33,890	\$36,995
11	BMW	Sedan	Europe	Rear	\$40,530	\$44,295
12	BMW	Sedan	Europe	Rear	\$32,525	\$35,495
13	BMW	Sedan	Europe	All	\$34,115	\$37,245
14	Nissan	Sports	Asia	Rear	\$25,203	\$26,910
15	Nissan	Sports	Asia	Rear	\$31,845	\$34,390
16	Toyota	SUV	Asia	Front	\$24,801	\$27,710

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 **+**.

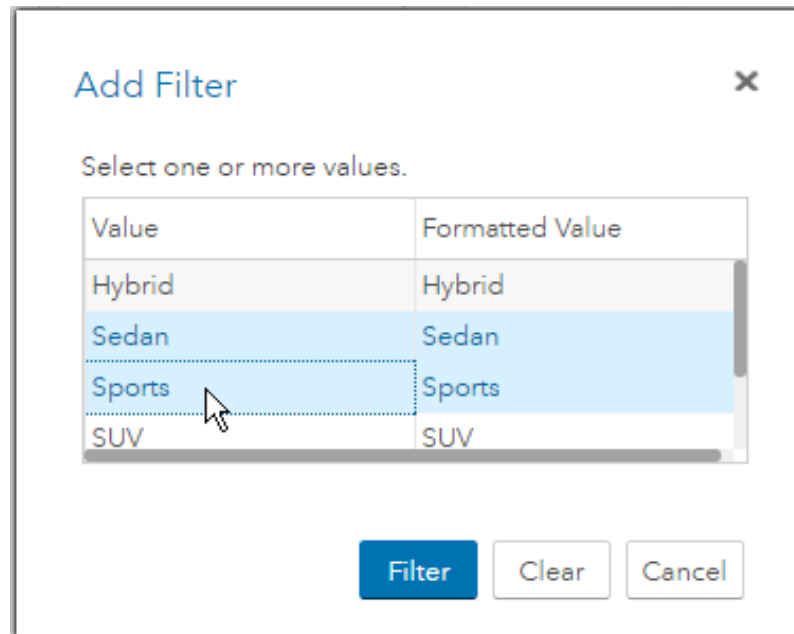
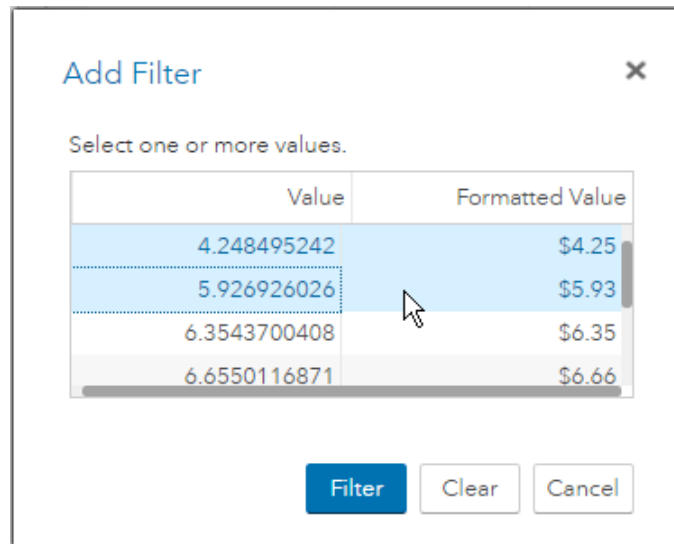
The screenshot shows the SASHELP.CARS dataset with 428 rows and 15 columns. An 'Add Filter' dialog box is open for the MSRP column. The dialog prompts the user to 'Specify the criteria for "MSRP"'. A dropdown menu shows comparison operators: \geq , $=$, \leq , $>$, $<$, and $<$. The \leq operator is selected. The criteria field contains '34,000'. The background table shows the following data:

Make	Type	Origin	DriveTrain	Invoice	MSRP	
				\$39,014	\$43,755	
				\$41,100	\$46,100	
				\$27,797	\$29,865	
				\$30,884	\$33,295	
				\$28,245	\$30,795	
				\$34,800	\$37,995	
				\$26,155	\$28,495	
				\$27,745	\$30,245	
				\$30,110	\$32,845	
				\$33,890	\$36,995	
				\$40,530	\$44,295	
12	BMW	Sedan	Europe	Rear	\$32,525	\$35,495
13	BMW	Sedan	Europe	All	\$34,115	\$37,245
14	Nissan	Sports	Asia	Rear	\$25,203	\$26,910
15	Nissan	Sports	Asia	Rear	\$31,845	\$34,390
16	Toyota	SUV	Asia	Front	\$24,801	\$27,710

The Add Filter window for a character column is case sensitive and searches for values that contain the text that you enter.

The screenshot shows the 'Add Filter' dialog box for a character column. The dialog prompts the user to 'Value contains:' followed by an empty text input field. Below the input field, it says '(Case sensitive)'. At the bottom, there are three buttons: 'Filter', 'Clear', and 'Cancel'.

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:

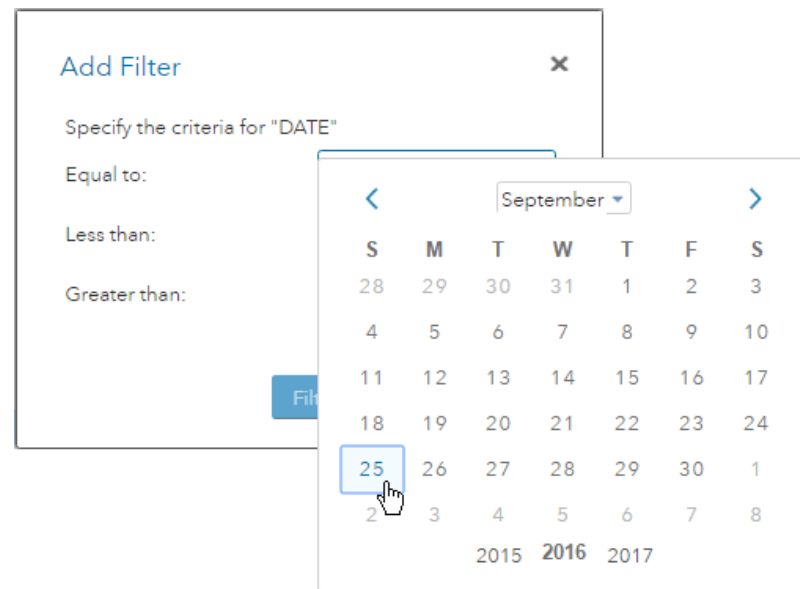
```
column-name = 15
```



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:

```
(column-name >= 123.456775 and 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.



When you create a filter on your data, the filter criteria are displayed at the top of the workspace. You can click  to edit the filter and  to delete the filter. SAS Studio remembers any filters that you create each time you open the data.

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 105](#).

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.

The screenshot displays the SAS Studio interface for importing an Excel file. The top window shows the file name `as_products.xls` and the source location. The worksheet name is set to `First worksheet`. The output data is configured to be saved to the `WORK` library with the data set name `IMPORT`. The code editor shows the following SAS code:


```

11 PROC IMPORT DATAFILE=REFFILE
12     DBMS=XLS
13     OUT=WORK.IMPORT;
14     GETNAMES=YES;
15 RUN;
16
17 PROC CONTENTS DATA=WORK.IMPORT; RUN;
18
19
20 %web_open_table(WORK.IMPORT);

```








The interface includes tabs for `CODE`, `LOG`, and `RESULTS`, and a toolbar with icons for file operations and execution. The encoding is set to `UTF-8`.

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.

CODE LOG **RESULTS** OUTPUT DATA

▶ Table of Contents

The CONTENTS Procedure			
Data Set Name	WORK.IMPORT	Observations	60
Member Type	DATA	Variables	6
Engine	V9	Indexes	0
Created	10/26/2016 10:50:16	Observation Length	80
Last Modified	10/26/2016 10:50:16	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	817
Obs in First Data Page	60
Number of Data Set Repairs	0

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.

CODE LOG RESULTS **OUTPUT DATA**

Table: WORK.IMPORT View: Column names Filter: (none)

Columns Total rows: 60 Total columns: 6 Rows 1-60


Property	Value
Select all	<input checked="" type="checkbox"/>
SUPPLIER_ID	<input checked="" type="checkbox"/>
PRODUCT_CODE	<input checked="" type="checkbox"/>
DESCRIPTION	<input checked="" type="checkbox"/>
COST	<input checked="" type="checkbox"/>
PRICE	<input checked="" type="checkbox"/>
PRODUCT_TYPE	<input checked="" type="checkbox"/>

	SUPPLIER_ID	PRODUCT_C...	DESCRIPTION
1	KOL	1005	TRACK-IT-ALL
2	BBI	1207	OMNIBUS
3	CAS	1526	EASY DRAW
4	TCD	1612	DISK MAGIC
5	CCS	1807	BACKTRACK
6	BBI	1828	MAKE-A-COPY
7	SDC	1991	QUEST FOR THE KINGS EAR
8	CSI	2123	DAZED KNIGHTS TOURNAMENT
9	DJC	2147	VIRUS INNOCULATOR
10	PCI	2181	MEMORY EXPANDER
11	SDC	2288	UNCORRUPT
12	TCD	2782	WORKMATE IV
13	PCI	2842	STOCK MARKET ANALYST
14	HSS	2927	SUPERQUIK
15	BBI	3049	BALBOA
16	CAS	3104	PRINTIT FAST

Import a Delimited File

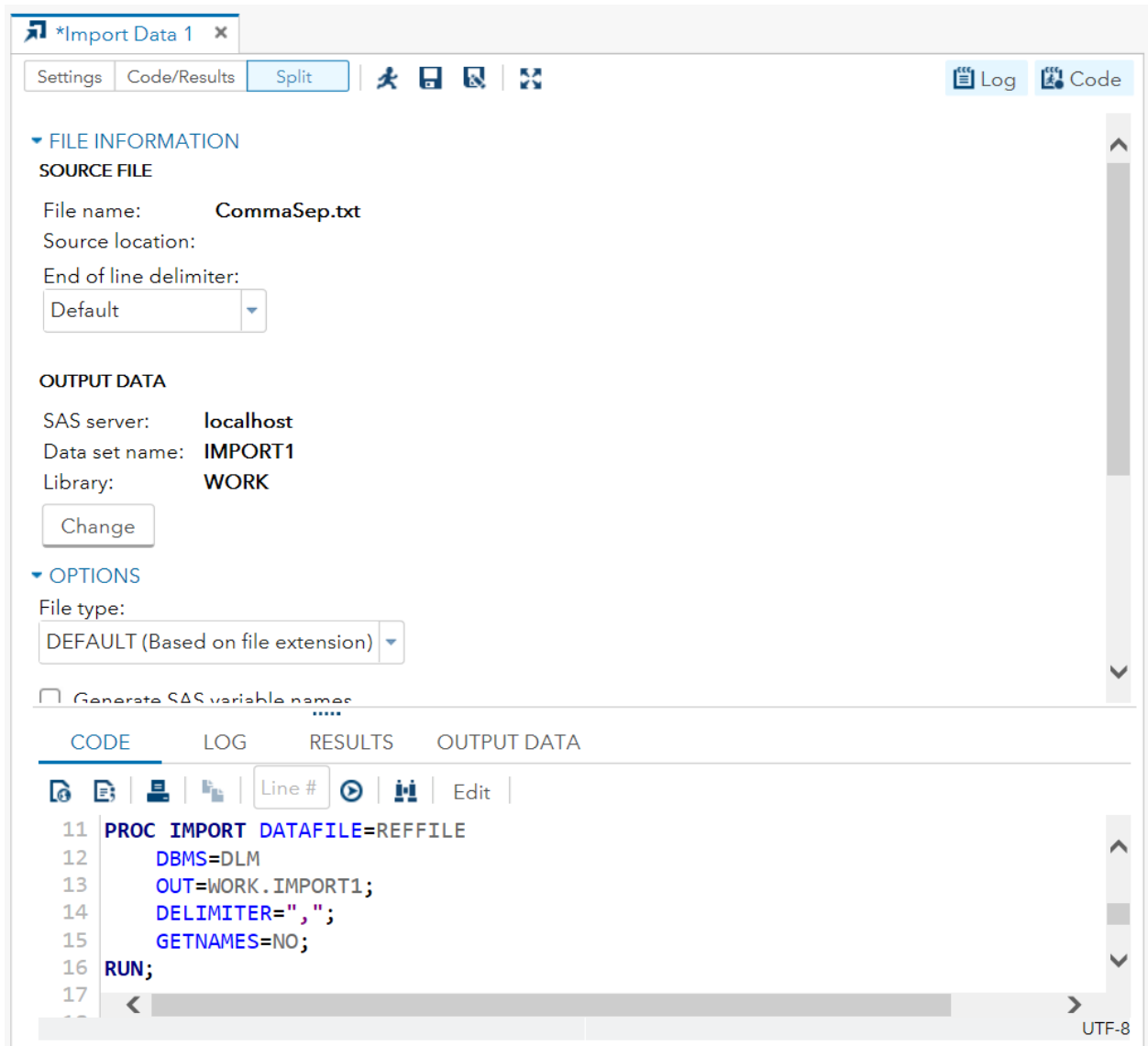
Note: For some delimited files (such as files with a .dat extension), the Import Tool might not be available. To import these delimited files, save the file as a text file. Then you can use the Import Tool. To import a tab-delimited file, the filename must have a TAB extension.

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.

CODE LOG **RESULTS** OUTPUT DATA



▶ Table of Contents

The CONTENTS Procedure

Data Set Name	WORK.IMPORT1	Observations	74
Member Type	DATA	Variables	5
Engine	V9	Indexes	0
Created	10/26/2016 13:42:18	Observation Length	72
Last Modified	10/26/2016 13:42:18	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Engine/Host Dependent Information

Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	908
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.

CODE LOG RESULTS **OUTPUT DATA**

Table: WORK.IMPORT1 | View: Column names | Filter: (none)

Columns Total rows: 74 Total columns: 5 Rows 1-74


	VAR1	VAR2	VAR3
1	1	Chai	10 box
2	2	Chang	24 - 12
3	3	Aniseed Syrup	12 - 55
4	4	Chef Anton's Cajun Seasoning	48 - 6 c
5	5	Chef Anton's Gumbo Mix	36 box
6	6	Grandma's Boysenberry Spread	12 - 8 c
7	7	Uncle Bob's Organic Dried Pears	12 - 1 l
8	8	Falserthwoods Cranberry Sauce	12 - 12
9	9	Mishi Kobe Niku	18 - 50
10	10	Ikura	12 - 20
11	11	Queso Cabrales	1 kg pk
12	12	Queso Manchego La Pastora	10 - 50
13	13	Konbu	2 kg bc
14	14	Tofu	40 - 10

Property Value

Label	
Name	
Length	
Type	

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 81.

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.

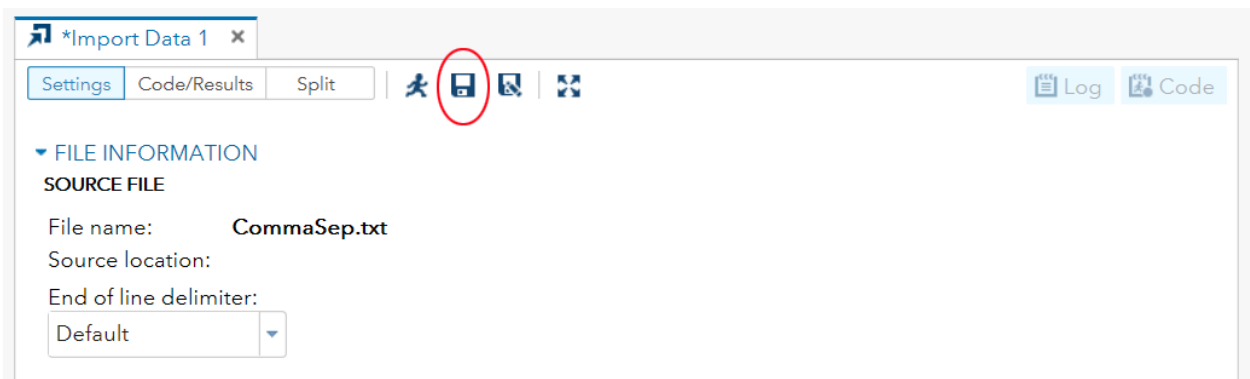


Save the Import Task

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 .



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

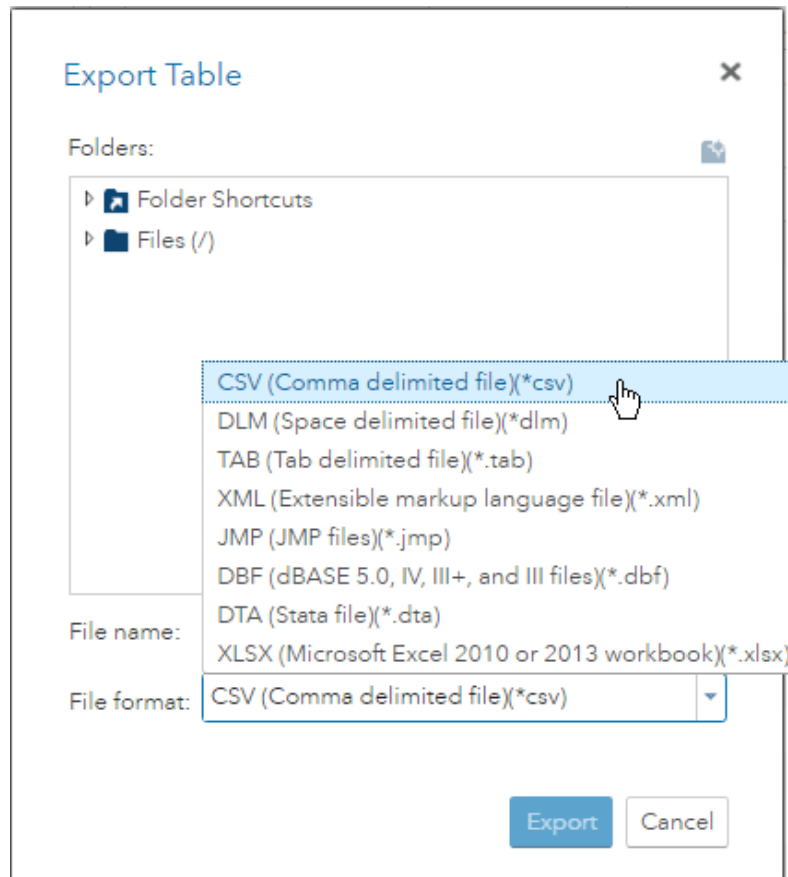
Exporting Data

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.

Chapter 6

Working with Results

Viewing Results	91
Default SAS Studio Output	91
Viewing Default Results	91
Downloading Default HTML5, PDF, and RTF Results from SAS Studio	93
Downloading Generated Data	93
Sending Your Results to Another User	94
About the SAS Output Delivery System	95
About SAS ODS Statistical Graphics	95
About SAS ODS Statistical Graphics	95
SAS ODS Graphics Designer	95
SAS ODS Graphics Editor	97
How to Edit Your Graphics Output	98

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 Result Preferences](#)” on page 110.

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

The screenshot shows the SAS Studio interface with the **Results** tab selected. The left pane shows the **DATA** tab with the dataset **SASHELP.CARS** and variables **Model**, **MSRP**, and **MPG_City** listed. The right pane shows the **RESULTS** tab with a **Table of Contents** and two data tables:

List Data for SASHELP.CARS

Make=Acura

Obs	Model	MSRP	MPG (City)
1	MDX	\$36,945	17
2	RSX Type S 2dr	\$23,820	24
3	TSX 4dr	\$26,990	22
4	TL 4dr	\$33,195	20
5	3.5 RL 4dr	\$43,755	18
6	3.5 RL w/Navigation 4dr	\$46,100	18
7	NSX coupe 2dr manual S	\$89,765	17

Make=Audi

Obs	Model	MSRP	MPG (City)
8	A4 1.8T 4dr	\$25,940	22
9	A4 1.8T convertible 2dr	\$35,940	23
10	A4 3.0 4dr	\$31,840	20
11	A4 3.0 Quattro 4dr manual	\$33,430	17
12	A4 3.0 Quattro 4dr auto	\$34,480	18
13	A6 3.0 4dr	\$36,640	20
14	A6 3.0 Quattro 4dr	\$39,640	18
15	A4 3.0 convertible 2dr	\$42,490	20
16	A4 3.0 Quattro convertible 2dr	\$44,240	18
17	A6 2.7 Turbo Quattro 4dr	\$42,840	18
18	A6 4.2 Quattro 4dr	\$49,690	17
19	A8 L Quattro 4dr	\$69,190	17
20	S4 Quattro 4dr	\$48,040	14
21	RS 6 4dr	\$84,600	15
22	TT 1.8 convertible 2dr (coupe)	\$35,940	20
23	TT 1.8 Quattro 2dr (convertible)	\$37,390	20
24	TT 3.2 coupe 2dr (convertible)	\$40,590	21
25	A6 3.0 Avant Quattro	\$40,840	18
26	S4 Avant Quattro	\$49,090	15

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.

The screenshot shows the SAS Studio interface. On the left, the 'DATA' pane shows the dataset 'SASHELP.CARS' with variables Model, MSRP, and MPG_City. The 'ROLES' pane shows 'Model' as a key role. The 'RESULTS' pane shows a 'Table of Contents' with 'Make=BMW' selected. Below it, a table displays the data for BMW cars.


Obs	Model	MSRP	MPG (City)
27	X3 3.0i	\$37,000	18
28	X5 4.4i	\$52,195	18
29	325i 4dr	\$28,495	20
30	325Ci 2dr	\$30,795	20
31	325Ci convertible 2dr	\$37,995	19
32	325xi 4dr	\$30,245	19
33	330i 4dr	\$35,495	20
34	330Ci 2dr	\$36,995	20
35	330xi 4dr	\$37,245	20
36	525i 4dr	\$39,995	19
37	330Ci convertible 2dr	\$44,295	19
38	530i 4dr	\$44,995	20
39	545iA 4dr	\$54,995	18
40	745i 4dr	\$69,195	18
41	745Li 4dr	\$73,195	18
42	M3 coupe 2dr	\$48,195	18
43	M3 convertible 2dr	\$58,595	18
44	Z4 convertible 2.5i 2dr	\$33,895	20
45	Z4 convertible 3.0i 2dr	\$41,045	21
46	325xi Sport	\$32,845	19

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 

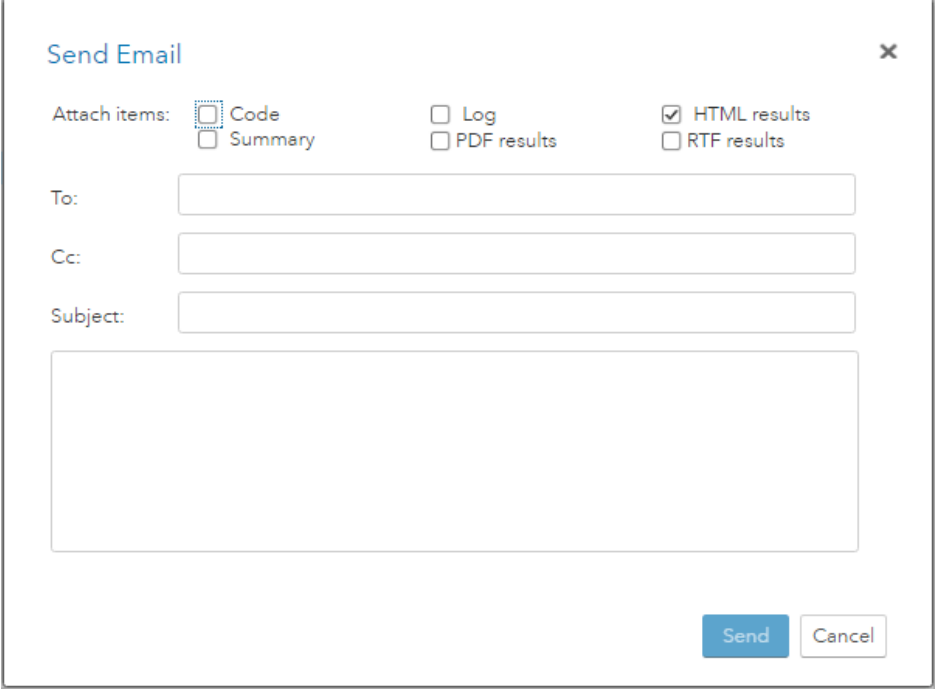
Sending Your Results to Another User

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 . The Send Email window appears.




The screenshot shows the "Send Email" dialog box. It features a title bar with the text "Send Email" and a close button (X). Below the title bar, there is a section labeled "Attach items:" with six checkboxes: "Code", "Summary", "Log", "PDF results", "HTML results", and "RTF results". The "HTML results" checkbox is checked. Below the checkboxes are three text input fields labeled "To:", "Cc:", and "Subject:". At the bottom of the dialog is a large, empty text area. In the bottom right corner, there are two buttons: "Send" and "Cancel".

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

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.

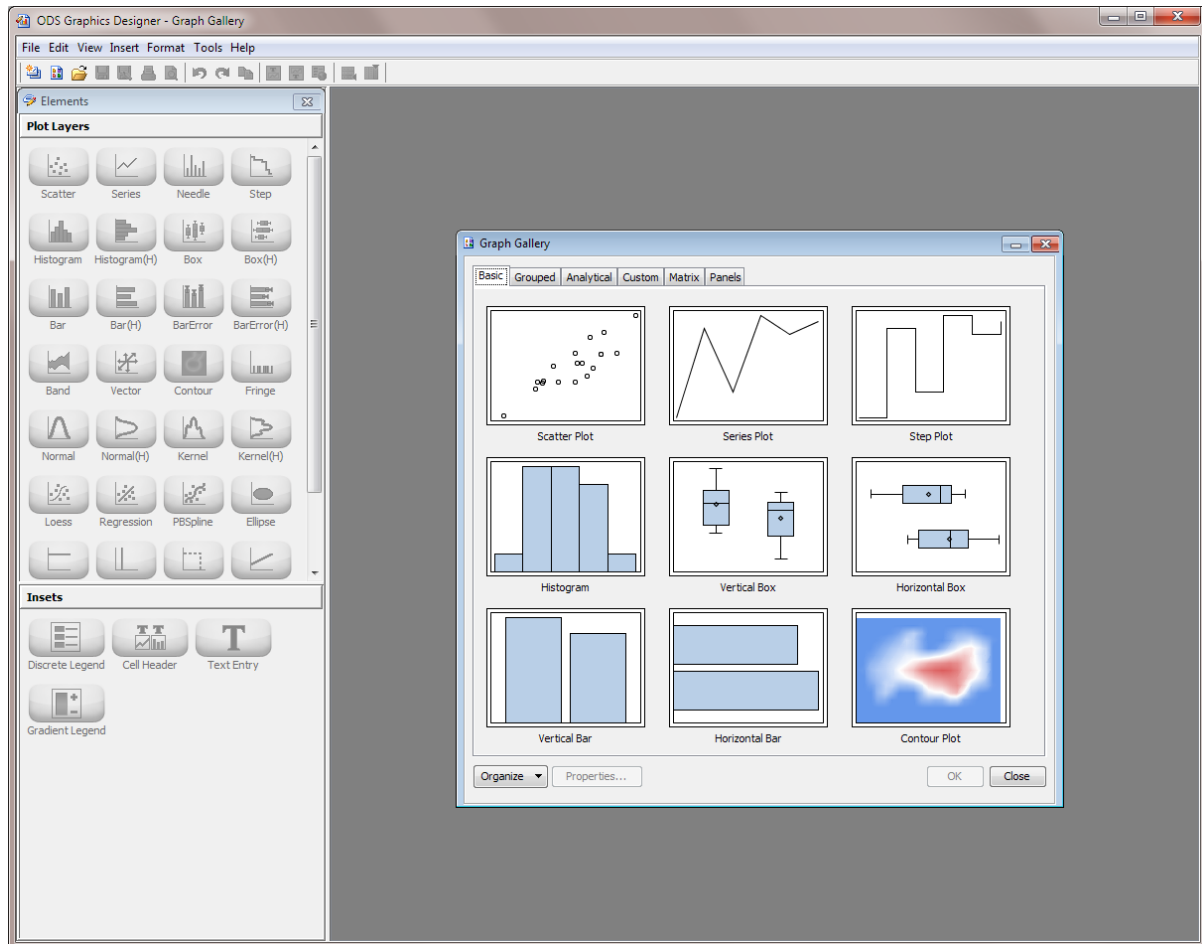
Note: Only the Windows version of the SAS ODS Graphics Designer is supported when you are running SAS Studio. The SAS ODS Graphics Designer must be installed on the same machine as the browser that you are using to access SAS Studio. (SAS Studio might be installed on a different machine.) If multiple 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 . 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 . 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.

- 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

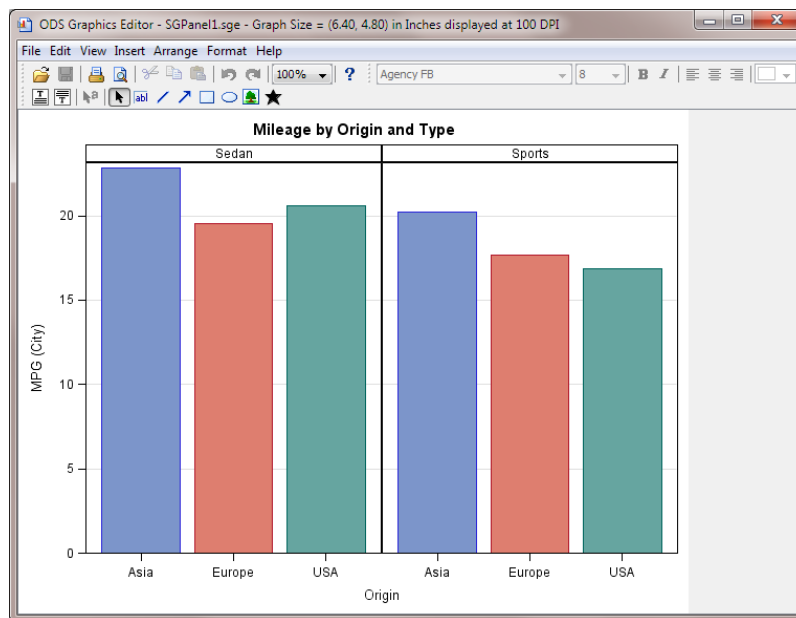
- Include this statement in your SAS code so that you can edit your graphics output:

```
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.

- 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 Tasks in SAS Studio

What Is a Task?	99
How to Run a Task	100
Save a Task and Its Option Settings	102
Edit a Predefined Task	102
Create a New Task	103
Customizing the Task Code and the Task Layout in the Workspace	104

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. Some categories and their tasks might not be available at your site because you do not have the required SAS product.

The tasks in the Data, Graph, and Combinatorics and Probability categories are always available. This table lists the task categories and their product dependencies.

Task Category in SAS Studio	Required SAS Product
Statistics	SAS/STAT
High-Performance Statistics	SAS/STAT
Power and Sample Size	SAS/STAT
Multivariate	SAS/STAT
Cluster Analysis	SAS/STAT
Econometrics	SAS/ETS
Forecasting	SAS/ETS

Task Category in SAS Studio	Required SAS Product
Statistical Process Control	SAS/QC
Data Mining	SAS Enterprise Miner
Network Optimization	SAS/QC
Viya Machine Learning	SAS Viya, SAS/CONNECT

For more information about each of these tasks, see *SAS Studio: Task Reference Guide*.

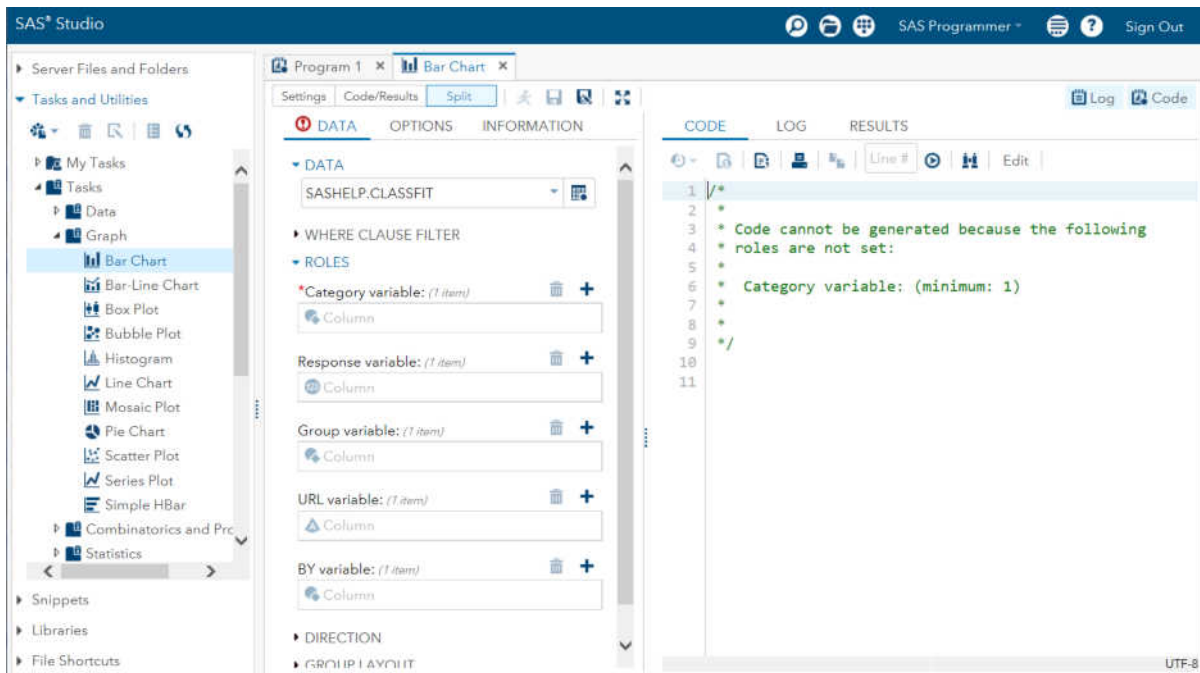
You can edit a copy of these predefined tasks in order to customize the tasks for your site. You can also build your own tasks.

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.

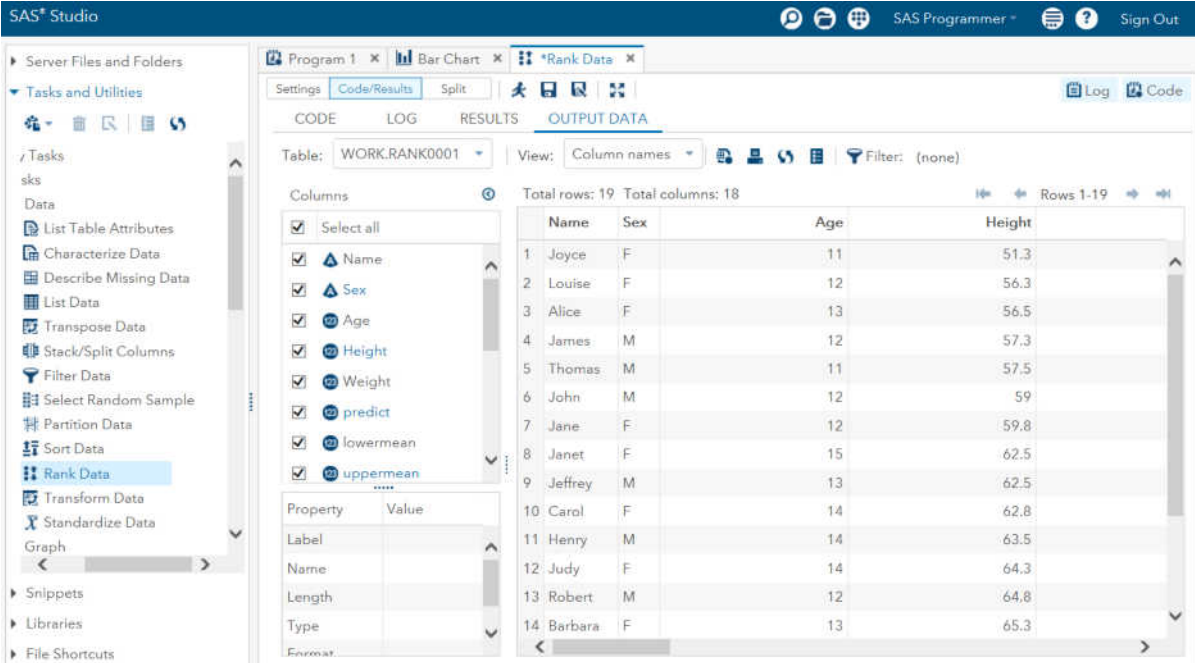


4. If the **Data** tab is available, specify an input data source and select columns for the roles in the data source. A role is a description of a variable's purpose in the task. To add a column to a role, click **+**. A list of available columns for that role appears. If only one column can be assigned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

- 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](#).
- To run the task, click .

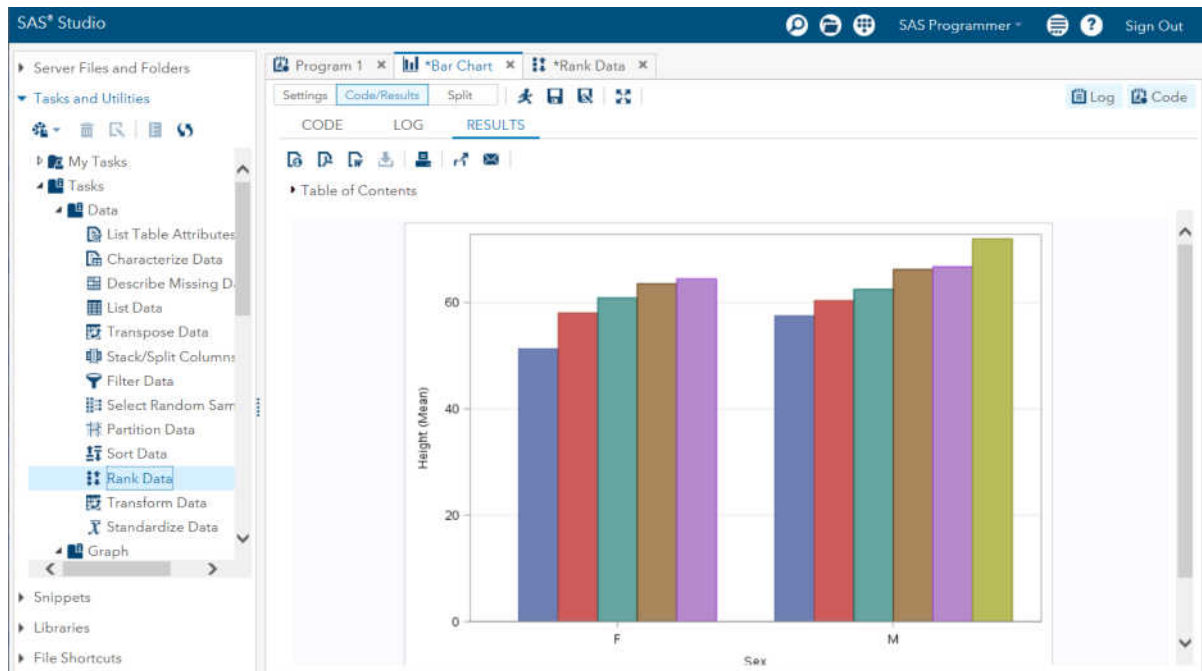
If the task generates output data, the table opens in the **Output Data** tab.



The screenshot shows the SAS Studio interface with the 'Rank Data' task selected in the left-hand pane. The main window displays the 'OUTPUT DATA' tab for the task, showing a table with 19 rows and 18 columns. The table contains the following data:

	Name	Sex	Age	Height
1	Joyce	F	11	51.3
2	Louise	F	12	56.3
3	Alice	F	13	56.5
4	James	M	12	57.3
5	Thomas	M	11	57.5
6	John	M	12	59
7	Jane	F	12	59.8
8	Janet	F	15	62.5
9	Jeffrey	M	13	62.5
10	Carol	F	14	62.8
11	Henry	M	14	63.5
12	Judy	F	14	64.3
13	Robert	M	12	64.8
14	Barbara	F	13	65.3

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.ctl 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**.

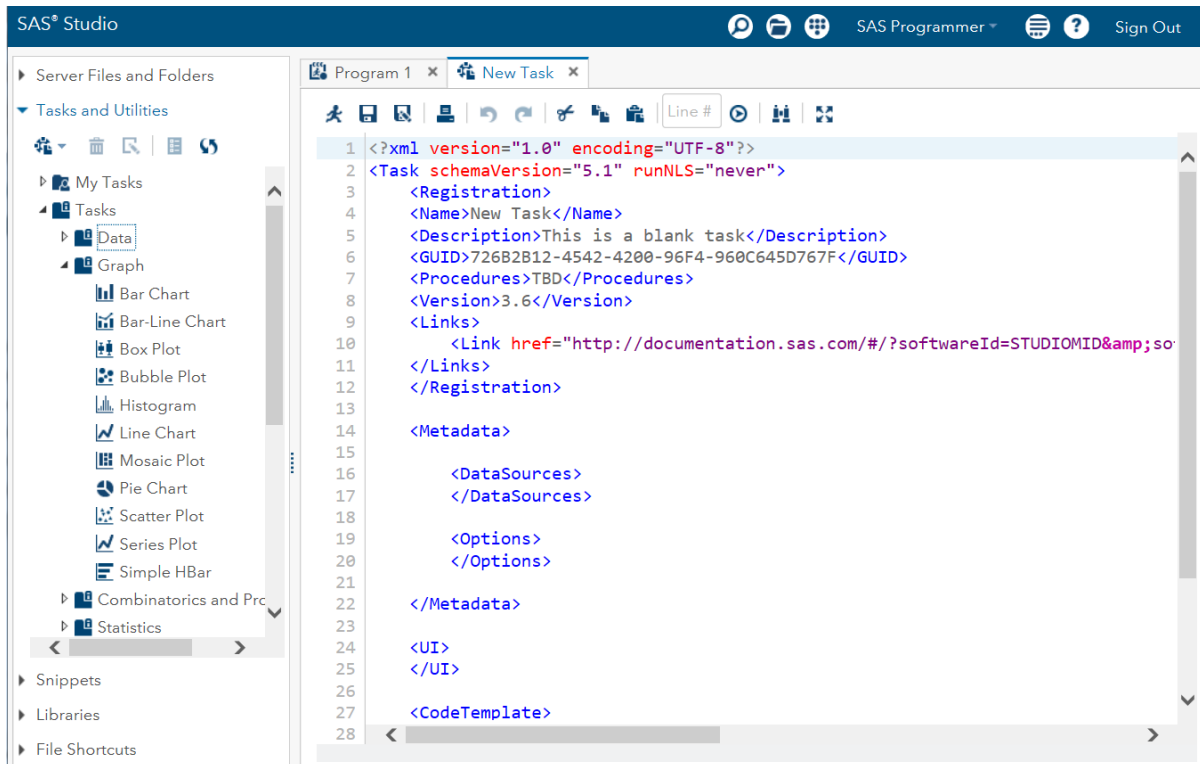
5. Open the **My Tasks** folder and select the copied task.
6. Click . The XML file for the task appears.
7. Edit the XML file and save your changes. To preview your changes, click .



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 111.


Appendix 1

Customizing SAS Studio

About Setting Your Preferences	105
Setting General Preferences	105
Setting the Start Up Preferences	107
Setting the Editor Preferences	108
Setting the Result Preferences	110
Setting Task Preferences	111
Access Repositories in SAS Studio	112
Setting Preferences for Background Job Submissions	112

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.

Option	Description
Show generated code in the SAS log	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.
Include a Show Details button in error messages	adds a Show Details button to any error messages that SAS Studio generates.

Option	Description
Size grid columns to content	automatically adjusts the width of the columns in the table viewer to fit the size of the column content.
Start new programs in interactive mode	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 29 .
Automatically refresh libraries after each submission	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.
Automatically refresh files and folders after each submission	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.
SAS variable name policy	<p>enables you to specify one of the following sets of rules to apply to SAS variable names.</p> <ul style="list-style-type: none"> • 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.

Option	Description
Default text encoding	<p>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 2, “Text Encoding Options and Language Mappings,” on page 115.</p> <p><i>Note:</i> 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).</p>
Display a message on arrival	<p>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.</p>
Capture all log events	<p>Records all diagnostic messages each time you run a program, task, or query. You can view the diagnostic messages by clicking Messages in the lower right corner of your SAS Studio browser window. In the Messages window, click the Filter by drop-down list and select Debug. The debug messages are available in the Messages window only when you have selected this option.</p>

Setting the Start Up Preferences

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

On startup	<p>specifies the tabs to be displayed when you start the SAS Studio application. You can choose from these options:</p> <ul style="list-style-type: none"> • Open a new program/process flow tab opens a new program tab in the SAS Programmer perspective and a new process flow tab in the Visual Programmer perspective. • Continue where you left off restores the tabs that were open in your prior session of SAS Studio. This option is selected by default.
Time-out interval	specifies the amount of time in hours that SAS Studio allows you to be logged on without any activity. The default value is one hour.

Note: For information about how these options affect the auto-save feature, see [“Setting the Editor Preferences”](#) on page 108.

Setting the Editor Preferences

From the **Editor** page, you can specify these options for the code editor.

Option	Description
Enable autocomplete	turns on the autocomplete feature of the code editor. This feature can predict the next keyword that you want to type before you actually type it completely. For more information, see “Using the Autocomplete Feature” on page 21.
Enable hint	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 Syntax Help . This option is not selected by default.
Tab width	<p>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.</p> <p><i>Note:</i> In Microsoft Internet Explorer and Apple Safari, spaces are used instead of Tab characters. If you are using those browsers, you must select the Substitute spaces for tabs check box in order for the value of the tab width to be used.</p>

Option	Description
Substitute spaces for tabs	<p>inserts the number of spaces listed in the Tab width box instead of a single tab character. This option applies to both text that you type in the code editor and text that you paste into the code editor.</p>
Enable color coding	<p>displays the text in the code editor in different colors to help you identify different elements in the syntax.</p>
Show line numbers	<p>displays line numbers in the leftmost column of the program and log windows.</p>
Font size	<p>specifies the font size of the text in the code editor and log window.</p>
Enable autosave	<p>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.</p> <p><i>Note:</i> New program files are not auto-saved until you save them first.</p> <p>The files are saved at the interval specified in the Autosave Interval option. The default interval is 30 seconds.</p> <p><i>Note:</i></p> <ul style="list-style-type: none"> • 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. <p>For more information, see “Setting the Start Up Preferences” on page 107.</p>

Setting the Result Preferences

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

Option	Description
HTML output style	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.
Generate HTML graphs as SVG	creates SVG graphs instead of PNG graphs in HTML output. SVG graphs maintain clarity when you zoom in and out.
Display warning if results are larger than <i>n</i> MB	displays a warning message when you attempt to open a results file that is larger than <i>n</i> megabytes (MB). The default value is 4 MB.
Enable accessible graph option (Pre-production)	adds accessibility metadata to graphs that are created by ODS Graphics. Users with disabilities will access the accessibility metadata using a companion product to ODS Graphics that will be released in the near future. This option is available only if you are using Google Chrome as your web browser. For more information, see the ODS HTML5 Statement in <i>SAS Output Delivery System: User's Guide</i> . This option is preproduction for SAS Studio 3.6.
Produce PDF output	generates results in PDF format. This option is selected by default.
PDF output style	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.
Generate the default table of contents	creates a table of contents in the PDF file.
Produce RTF output	generates results in RTF format. This option is selected by default.
RTF output style	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.
Automatically open generated output data	displays the output data that is created when you run a task or submit code.

Note: If you want to use a custom style, you must customize the SAS Studio output environment. For more information, see [Appendix 3, “Customized Output Environment,”](#) on page 117.

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.

Option	Description
Trim all leading and trailing spaces in generated code	removes any blank spaces that appear before or after the generated code.
Generate header comments for task code	adds comments before the generated code for a SAS task.
Automatically format generated code	automatically formats any code that is generated by a task and displayed in the code editor.
View	<p>specifies how to lay out the task options, task code, and task results in your workspace. You can choose from these options:</p> <ul style="list-style-type: none"> • 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.

Access 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 access additional repositories. For more information about repositories, see *SAS Studio: Developer's Guide to Repositories*.

Setting Preferences for Background Job Submissions

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

If the background job log or output file already exists

specifies how to handle the background job submission if a log and output file already exist. You can choose from these options:

- **Delete existing log and output** deletes the existing log and output files and replaces them with new files when you resubmit the program as a background job.
 - **Fail background job submission** cancels the background job submission. You must delete the existing log and output files and resubmit the program.
 - **Prompt for fail or overwrite** 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.
 - **Number log/output** saves all log and output files by creating a unique filename for each file. The log and output files are saved as **program-name (userid-YYYY-MM-DD HHMMSS)**.
-

Output and log destination

specifies where to save the output and log files. You can choose from these options:

- **Same folder as .sas file** saves the log and output files to the same folder as the .sas code file.
 - **Use folder** enables you to specify a location in which to save the log and output files. Click **Browse** to search for a folder.
 - **Prompt for output and log file names** 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 **Browse** to search for a folder and specify a filename.
-

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 105](#).

Appendix 2

Text Encoding Options and Language Mappings

About the Text Encoding to Language Mappings	115
Text Encoding Options and Language Mappings	115

About the Text Encoding to 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 105](#).

Text Encoding Options and Language Mappings

Text Encoding Option	Language
Windows-1250	(Central European languages): Polish, Czech, Slovak, Hungarian, Slovenian, Serbian Latin, Croatian, Bosnian, Romanian, Albanian
Windows-1251	(Cyrillic languages): Russian, Byelorussian, Bulgarian, Serbian Cyrillic, Macedonian, Ukrainian
Windows-1252	(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
Windows-1253	Greek
Windows-1254	Turkish

Text Encoding Option	Language
Windows-1255	Hebrew
Windows-1256	Arabic
Windows-1257	(Baltic languages): Estonian, Latvian, Lithuanian
Windows-1258	Vietnamese

Appendix 3

Customized Output Environment

Overview	117
Generate Output for Other Output Destinations	118
Send Your Results to Another Location	118
Use a Custom Style for Your Output	119
Use an Image Format Other Than the Default	119
Create a Drill-down Graph	119
Create an Animated GIF or SVG Image	120

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:

```
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:

```
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:

```
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:

```
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=file-reference` option in your ODS statement. Here is an example:

```
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:

```
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*.

- If you are using the Graph Template Language to create the graph, see *SAS Graph Template Language: User's Guide*.

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 4

Keyboard Shortcuts

Keyboard Shortcuts in SAS Studio Workspace	121
Keyboard Shortcuts for the Code Editor	123

Keyboard Shortcuts in SAS Studio Workspace

Action	Keyboard Shortcut for Microsoft Windows	Keyboard Shortcut for Mac OS X
Zoom in. <i>Note:</i> When you zoom into SAS Studio, no scroll bars are displayed for the browser window.	Ctrl+plus sign	Cmd+plus sign
Zoom out.	Ctrl+minus sign	Cmd+minus sign
Reset the zoom state.	Ctrl+0	Cmd+0
Maximize the view of the currently displayed tab or exit the maximized view.	Alt+F11	Fn+Shift+F11
Open a pop-up menu in the navigation pane.	Select an item, and press Shift+F10. <i>Note:</i> If you use 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.	Select an item, and press Fn+Shift+F10.
Create a new SAS program.	F4	Fn+F4

Action	Keyboard Shortcut for Microsoft Windows	Keyboard Shortcut for Mac OS X
Save the SAS program.	Ensure that the Code tab for a SAS program is displayed, and press Ctrl+S. <i>Note:</i> 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. <i>Note:</i> 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. <i>Note:</i> 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. <i>Note:</i> When you reset the SAS session, all libraries and file shortcuts that you created during the current session are deleted.	F9	Fn+F9
Download the log or results as an HTML file.	Ensure that the Log or Results tab is displayed, and press Ctrl+S.	Ensure that the Log or Results tab is displayed, and press Cmd+S.
Print the SAS program, log, or results.	Ctrl+P <i>Note:</i> 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. <i>Note:</i> You cannot close the Code , Log , and Results tabs.	Ensure that the focus is on the tab label (press Alt+3 if necessary), and then press Delete.	Ensure that the focus is on the tab label (press Option+3 if necessary), and then press Delete. <i>Note:</i> 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 <i>Note:</i> This shortcut requires Internet Explorer 10 or later.	Option+1

Action	Keyboard Shortcut for Microsoft Windows	Keyboard Shortcut for Mac OS X
<p>Move the focus to the label of the currently displayed secondary tab.</p> <p>In the SAS Programmer perspective, the secondary tabs are Code, Log, or Results.)</p> <p>In the Visual Programmer perspective, the secondary tabs are Flow, Results, and Properties.</p>	Alt+2	Option+2
<p>Move the focus to the label of the currently displayed primary tab and then navigate among all open tabs in the work area.</p>	Alt+3	Option+3
<p>Move the focus to the body of the Code tab on the currently displayed primary tab.</p>	Alt+4	Option+4
<p>Move the focus to the selected item in the Log tab on the currently displayed primary tab.</p>	Alt+5	Option+5

Keyboard Shortcuts for the Code Editor

This table contains keyboard shortcuts for the code editor in SAS Studio.

Action	Keyboard Shortcut for Microsoft Windows	Keyboard Shortcut for Mac OS X
<p>Open a pop-up menu in the code editor.</p>	Shift+F10	Fn+Shift+F10
<p>Create a new snippet from SAS code.</p> <p><i>Note:</i> The snippet is saved to your My Snippets folder.</p>	In the code editor, select the code and press Shift+Alt+A.	In the code editor, select the code and press Cmd+ Option+A.
<p>Insert an existing snippet.</p> <p><i>Note:</i> Only snippets saved to your My Snippets folder are available.</p>	Alt+I	Cmd+Option+I

Action	Keyboard Shortcut for Microsoft Windows	Keyboard Shortcut for Mac OS X
Add or remove comment tags for SAS code in the code editor.	Ctrl+/ <i>Note:</i> 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.	Cmd+/ <i>Note:</i> 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.
In the code editor, display the currently applicable list of SAS keywords (autocomplete feature).	Ctrl+Spacebar (This shortcut also navigates to the next list if more than one is available.) Ctrl+Shift+Spacebar (This shortcut also navigates to the previous list if more than one is available.) <i>Note:</i> These shortcuts do not work when JAWS is on. <i>Note:</i> 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: <ul style="list-style-type: none"> • First, in the region and language options of the Control Panel, select the option for changing your keyboard or input method. • Next, select the option or button for changing your keyboard. • 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. • Finally, restart any application that needs to use the new shortcut. (You might need to also restart your computer.) 	Control+Spacebar (This shortcut also navigates to the next list if more than one is available.) Control+Shift+Spacebar (This shortcut also navigates to the previous list if more than one is available.)

Action	Keyboard Shortcut for Microsoft Windows	Keyboard Shortcut for Mac OS X
Navigate between an autocomplete list and its associated Help pop-up window in the code editor.	Tab, Shift+Tab	Tab, Shift+Tab
Select all text in the code editor.	Ctrl+A	Cmd+A
Copy text in the code editor.	Ctrl+C	Cmd+C
Cut text in the code editor.	Ctrl+X	Cmd+X
Paste text in the code editor.	Ctrl+V	Cmd+V
Undo an action in the code editor.	Ctrl+Z	Cmd+Z
Redo an action in the code editor.	Ctrl+Y	Cmd+Shift+Z
Find and replace text in the code editor.	Ensure that the focus is in the body of the code editor, and then press Ctrl+F.	Ensure that the focus is in the body of the code editor, and then press Cmd+F.
Move the cursor to the beginning of the word on the left of the cursor.	Ctrl+left arrow	Option+left arrow
Move the cursor to the beginning of the word on the right of the cursor.	Ctrl+right arrow	
Move the cursor to the end of the word on the right of the cursor.		Option+right arrow
Move the cursor to the beginning of the previous paragraph.		Option+up arrow
Move the cursor to the beginning of the next paragraph.		Option+down arrow
Scrolls the editor (the cursor does not move).	Ctrl+down arrow, Ctrl+up arrow	
Scrolls a page down or up (moves cursor).	Page Down, Page Up	Page Down, Page Up <i>Note:</i> If you are not using the full extension keyboard, use Fn+down and Fn+up to page down and page up.

Action	Keyboard Shortcut for Microsoft Windows	Keyboard Shortcut for Mac OS X
Move the cursor to the end of the line.	End	Cmd+right arrow
Move the cursor to the beginning of the line.	Home	Cmd+left arrow
Move to the bottom of the currently displayed section of the navigation pane.	End	Fn+right arrow
Move to the top of the currently displayed section of the navigation pane.	Home	Fn+left arrow
Move the cursor to the top of the document.	Ctrl+Home	Cmd+up arrow
Move the cursor to the bottom of the document.	Ctrl+End	Cmd+down arrow
Select (highlight) text to the left of the cursor character-by-character.	Shift+left arrow	Shift+left arrow
Select (highlight) text to the right of the cursor character-by-character.	Shift+right arrow	Shift+right arrow
Select (highlight) text from the cursor to the same location on the previous line.	Shift+up arrow	Shift+up arrow
Select (highlight) text from the cursor to same location on the next line.	Shift+down arrow	Shift+down arrow
Delete the word on the left of the cursor.	Ctrl+Backspace	Option+Delete (backward) <i>Note:</i> If your keyboard does not contain a key for deleting backward, then press Fn+Delete.
Delete the word on the right of the cursor.	Ctrl+Delete	Option+Delete (forward) <i>Note:</i> If your keyboard does not contain a key for deleting forward, then press Fn+Delete.

Action	Keyboard Shortcut for Microsoft Windows	Keyboard Shortcut for Mac OS X
Delete the current line.		Cmd+Delete (backward) <i>Note:</i> If your keyboard does not contain a key for deleting backward, then press fn +delete.
Select (highlight) the word on the left of the cursor.	Ctrl+Shift+left arrow	Option+Shift+left arrow
Select (highlight) the word on the right of the cursor.	Ctrl+Shift+right arrow	Option+Shift+right arrow
Select (highlight) text from the cursor to the beginning of the document.		Cmd+Shift+up arrow
Select (highlight) text from the cursor to the end of the document.		Cmd+Shift+down arrow
Select a column or vertical block of text.	Alt+left mouse button. Then drag the mouse pointer to select the text.	Option+left mouse button. Then drag the mouse pointer to select the text.
Change case of selected text.	Ctrl+Shift+U cycles among initial capital, uppercase, and lowercase. Ctrl+U converts the selected text to uppercase. Ctrl+L converts the selected text to lowercase.	Cmd+Shift+U cycles among initial capital, uppercase, and lowercase. Cmd+U converts the selected text to uppercase. Cmd+L converts the selected text to lowercase.

Recommended Reading

- *Getting Started with Programming in SAS Studio*
- *The Little SAS Book: A Primer* (Buy)
- *Learning SAS by Example: A Programmer's Guide* (Buy)
- *SAS Statistics by Example* (Buy)
- *Elementary Statistics Using SAS* (Buy)

For a complete list of SAS publications, go to 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

Index

Special Characters

[_CLIENTAPP](#) 27
[_CLIENTAPPVERSION](#) 27
[_SASPROGRAMFILE](#) 27

A

autocomplete 21, 108
 autoexec file 14
 autosaving
 enabling 109

B

background jobs
 messages 107
 background submissions
 cancelling 112
 creating unique log and output
 filenames 112
 deleting log and output 112
 prompting for fail or overwrite 112
 prompting for location to save log and
 output 113
 saving log and output to code folder
 location 113
 saving log and output to specified
 location 113
 brackets
 matching 24

C

changing your SAS workspace server 15
 code
 editing from a task 26
 formatting SAS code 28
 selecting a column 25
 showing in the SAS log 105
 trimming blank spaces 111
 code editor
 about 17
 autocomplete 21, 108
 color coding 109

 customizing 38, 108
 enabling autosave 109
 specifying font size 109
 substituting spaces for tabs 109
 syntax help window 108
 tab width 108
 code snippets 31
 custom tasks 103

D

data
 exporting 90
 extended attributes 73
 filtering 78
 importing 81
 opening 75
 process flows 61
 sorting 78
 table viewer 73
 DBMS file, importing 88
 delimited file, importing 85

E

editing columns 25
 editor preferences 38, 108
 encoding
 specifying 107
 error messages
 show details 105
 Excel worksheet, importing 82
 exporting data 90

F

File Shortcuts section of the navigation
 pane 7
 filter
 changing the relationship 50
 creating 48
 filtering data 78
 folder shortcuts 3
 formatting SAS code 28

FTP folder shortcuts 3

G

general preferences 105
grouping query output 52

H

hints
code editor 108
HTML output 110

I

importing
DBMS file 88
delimited file 85
Excel worksheet 82
importing data 81
interactive mode 29, 106

J

joins
about 41
creating 41
modifying 44
types 43

L

libraries
using to add table and column names
25
Libraries section of the navigation pane 6
log
specifying font size 109
logs
displaying line numbers 109

M

macro variables 27
_CLIENTAPP 27
_CLIENTAPPVERSION 27
_SASPROGRAMFILE 27
matching parentheses 24
maximizing the work area 9
messages
details in the log 107
displaying in SAS Studio 107
My Tasks folder 102

N

names
SAS variables 106
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) 95
ODS Graphics Designer 95
ODS Graphics Editor 97
output data
displaying automatically 110

P

parentheses
matching 24
PDF output 110
perspectives 13
preferences
editor 38, 108
general 105
results 110
setting 105
start up 107
tasks 104, 111
Preferences window 105
process flows
adding a snippet 64
adding queries 67
adding SAS programs 62
adding tasks 68
arranging 61
color of nodes 61
control ports 58, 71
creating 58
data ports 58
generating code 71
importing data in 89
libraries 61
linking nodes 71
overview 57
priority of execution 61
properties 61
results 59
running 71
saving 72
status of nodes 59
subflows 69
program package 26

- program summary 27
- programs
 - autocomplete 21
 - creating 18
 - displaying line numbers 109
 - enabling autosave 109
 - formatting 28
 - interactive mode 29
 - opening 18
 - running 18
 - syntax help 23

Q

- query
 - adding tables 40
 - changing filter relationship 50
 - creating 40
 - creating a filter 48
 - creating a join 41
 - eliminating duplicate rows 52
 - grouping output 52
 - joins 41
 - modifying a join 44
 - running 56
 - saving results 55
 - selecting columns for output 44
 - sorting output 51
 - summary functions 46
 - types of joins 43

R

- repositories
 - accessing 112
- results
 - changing styles 110
 - displaying output data 110
 - downloading 93
 - downloading generated data 93
 - HTML output 110
 - PDF output 110
 - RTF output 110
 - sending to another user 94
 - viewing 91
- RTF output 110

S

- SAS ODS Graphics 95
- SAS Output Delivery System (ODS) 95
- SAS program package 26
- SAS program summary 27
- SAS Programmer perspective 13
- SAS workspace server 15
- searching in SAS Studio 11

- selecting a column of text 25
- server 15
- Server Files and Folders section of the navigation pane 3
- snippets
 - about 31
 - creating 37
 - inserting 37
- Snippets section of the navigation pane 6
- sorting data 78
- submission history 28
- syntax help 23
- syntax help window 108

T

- tab characters
 - number of spaces 108
- tab width
 - code editor 108
- table viewer 73
 - specifying width of columns 106
- tables
 - displaying the generated code 77
 - extended attributes 73
 - opening 75
 - resizing column widths 75
- tabs
 - rearranging in the workspace 9
- task code
 - adding comments 104, 111
 - formatting 104, 111
- tasks
 - about 99
 - adding comments before generated code 111
 - creating 103
 - customizing the layout 111
 - editing 102
 - editing the generated code 26
 - formatting generated code 111
 - running 100
 - showing SAS code 111
 - showing the log 111
- Tasks and Utilities section of the navigation pane 5

V

- Visual Programmer perspective 13

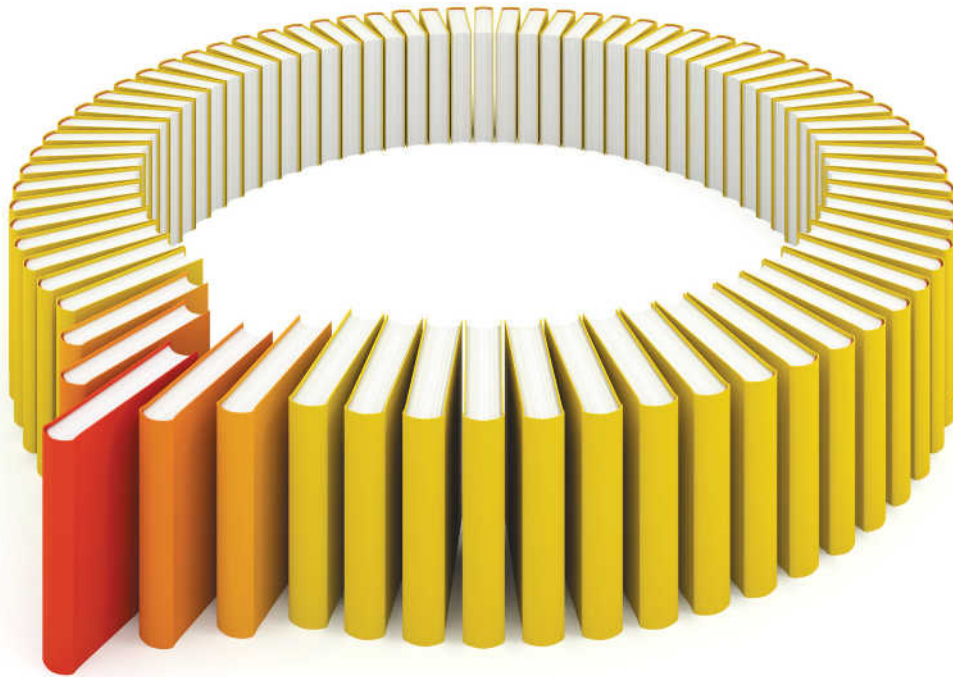
W

- warnings
 - opening large files 110
- work area 8

customizing 9
rearranging tabs 9
workspace server 15

X

XML file, creating 3
XML templates 103



Gain Greater Insight into Your SAS® Software with SAS Books.

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

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


THE POWER TO KNOW.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613

