SAS® Enterprise Guide® 8.3: User's Guide

SAS® Documentation
April 16, 2024

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2020. *SAS® Enterprise Guide® 8.3: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Enterprise Guide® 8.3: User's Guide

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

April 2024

8.3-P1:egug

Contents

1

Introduction to SAS Enterprise Guide

About SAS Enterprise Guide

SAS Enterprise Guide is a powerful Microsoft Windows client application that provides a guided mechanism to exploit the power of SAS and publish dynamic results throughout your organization.

What Can I Do with SAS?

SAS 9 is an integrated software suite for advanced analytics, business intelligence, data management, and predictive analytics. You can use SAS 9 software through both a graphical interface and the SAS programming language, or Base SAS.

With SAS 9 software, you can perform these tasks:

- access data in almost any format, including SAS tables, Microsoft Excel tables, and database files.
- manage and manipulate your existing data to get the data that you need. For example, you can subset your data, combine it with other data, and create columns.
- analyze your data using statistical techniques that range from descriptive measures such as correlations to logistic regression and mixed models to sophisticated methods such as modern model selection and Bayesian hierarchical models.
- present the results of your analyses in a meaningful report that you can share with others. The reports that you create can be saved in a wide variety of formats, including HTML, PDF, and RTF.

If your site has licensed and installed SAS Viya products, you can also write programs in SAS Enterprise Guide using SAS Viya syntax. The SAS Viya architecture is designed to enable high-performance, in-memory analytics. In SAS Viya, the data is held in memory, which means that complex algorithms and other analytical and statistical operations run very fast. SAS Viya is also designed to handle big data and cloud processing. For more information, see *SAS Cloud Analytic Services: Fundamentals*.

What Are ‘Tasks’?

The SAS Enterprise Guide tasks are based on SAS procedures and generate SAS code and formatted results for you. The tasks include SAS procedures from simple data listings to the most complex analytical procedures. You can choose a task by its procedure name or by the type of report that you want to generate, and there are task windows to help you set up each task.

For more information, see “Using SAS Tasks in SAS Enterprise Guide” on page 518.

Can I Still Write SAS Programs?

You can still write and run SAS programs just as you always have in the SAS Program Editor. However, in SAS Enterprise Guide, you use the color-coded, syntax-checking SAS language editor to create a program file.

For more information, see “About the Program Editor” on page 292.

SAS Enterprise Guide offers several options to select for programs that you write. If you have some code that you always want to add before or after code that you write, you can select one of the options to insert custom code.

Note: If your site has licensed and installed SAS Viya products, you can also write programs in SAS Enterprise Guide using SAS Viya syntax.

Am I Required to Use a Project?

Starting with SAS Enterprise Guide 8.1, you are not required to use a project to manage all of your data, tasks, programs, and results. You can still use projects, if you like, or you can use SAS Enterprise Guide as a tool to open and work on individual files. You can work with the following items without creating a project:

- Programs
- Data
- OLAP Cubes
- SAS Report Results
- SAS Visual Analytics Reports
- Logs

For more information, see “What Is a Project?” on page 28.

If you create a project, you can use the **When an item is opened, automatically add to the project** option to specify whether you want to automatically add any items you open to your project. For more information, see “Setting Project and Process Flow Options” on page 484.

Where Do I Start?

If you are not familiar with SAS Enterprise Guide, but you know the type of task that you want to perform, then this list can help you get started. This list includes tasks for a SAS 9 server.

Table 1.1 *Frequently Used Tasks in SAS Enterprise Guide for a SAS 9 Server*

Task That You Want to Perform	Functionality in SAS Enterprise Guide
Create or run a SAS program	Write a SAS 9 program or add a program to a project.

Task That You Want to Perform	Functionality in SAS Enterprise Guide
Create a detail or summary report with column subtotals and totals	Depending on your needs, there are several tasks that can create a report like this: List Report, List Data, Summary Tables, and Summary Statistics.
Create a simple report on a single table by filtering and sorting the data	Filter and Sort.
Create a report in which you can join two or more tables and create a computed column or a conditional expression	Query Builder.
Create a report in which you recode the values of a column by replacing values in a column with values that you specify	Use the Query Builder to create a recoded computed column.
Create a SAS data set from a Microsoft Excel or other text, HTML, or PC-based database file	Import Data wizard.
Rank your data based on one or more variables	Rank task.
Compute descriptive statistics for all observations or groups of observations	Summary Statistics task or Summary Tables task.
Create a report and plots that show the distribution of numeric data	Distribution Analysis task.
Create a vertical, horizontal, or three-dimensional bar chart to compare data values	Bar Chart task.
Create a pie chart to display the relative contribution of parts of the data to the whole	Pie Chart task.
Create a line plot to display the mathematical relationships between variables by revealing trends or patterns of data points	Line Plot task.

If your site has licensed and installed SAS 9.4M5 (or later) and SAS Viya, this list can help you get started programming in SAS Viya.

Note: If you are connecting to a server that is running SAS 9.4M4 or earlier, you can use SAS/CONNECT to submit code to a SAS Viya environment. For more information, see *SAS/CONNECT User's Guide*.

Table 1.2 *Frequently Used Tasks in SAS Enterprise Guide for a SAS Viya Server*

Task That You Want to Perform	Functionality in SAS Enterprise Guide
Create a CAS connection	Connect to a CAS server
Create a CAS session	Create a CAS session
Create a CAS library	“CASLIB= LIBNAME Statement Option” in *SAS Cloud Analytic Services: User’s Guide*
Load data to a CAS library	Upload to CAS task or “Load Data to a Caslib” in *An Introduction to SAS Viya Programming*

Getting Started Tutorial and Training Resources

If you are a new user, the Getting Started tutorial introduces you to the main features of SAS Enterprise Guide.

The tutorial guides you through these tasks:

- adding different types of data to a project
- importing data to create a SAS data set
- creating and modifying reports
- creating bar and pie charts
- using a query to join data and create a computed column
- generating a table of summary statistics
- running analyses on your data
- combining results into a single document
- creating and running a process flow

Start the tutorial by selecting **Help** ⇨ **Getting started tutorial**.

SAS provides a wide variety of training courses, videos, and documentation to help new and experienced SAS Enterprise Guide users. To access these resources, see Learn SAS Enterprise Guide.

Check for Software Updates

You can apply an update to SAS Enterprise Guide when you first open the application or while you are working in this application. To apply a software update, you must close all SAS Enterprise Guide sessions.

To apply any hot fixes or software updates while working in SAS Enterprise Guide, select **Help** ⇨ **Check for updates**.

If an update is available, a message appears. In this message dialog box, a link to `support.sas.com` is available where you can learn more about the contents of this software update. Then you can choose from these options:

- close all applications and install the update now
- continue working and have SAS Enterprise Guide remind you about this update at a later time
- skip this update

Transitioning to SAS Studio

SAS Enterprise Guide projects cannot be directly opened in SAS Studio, but you can use SAS Environment Manager to import your SAS Enterprise Guide project and create SAS Studio flow files. You can also export the code in your project and open the exported code in SAS Studio. For more information, see "Tips for SAS Enterprise Guide Users" in *SAS Studio: User's Guide*.

2

Connecting to a SAS Environment

Connecting to the SAS Viya Environment

About SAS Viya

The SAS Viya environment is the third generation of high-performance in-memory analytics. In SAS Viya, the data is held in memory, which means that complex algorithms and other analytical and statistical operations run very fast. The high-performance processing power of the SAS Viya environment is SAS Cloud Analytic Services (CAS). CAS is a server that provides the run-time environment for data management and analytics with SAS. (Run-time environment refers to the combination of hardware and software where data management and analytics take place.)

For more information about the SAS Viya environment and programming in CAS, see these resources:

- *SAS Cloud Analytic Services: User's Guide*
- *An Introduction to SAS Viya Programming*
- *SAS Cloud Analytic Services: Fundamentals*

Note: If your site has licensed and installed SAS 9.4M5 and SAS Viya, use this code to connect and manage your CAS sessions. If you are connecting to a server that is running SAS 9.4M4 or earlier, you can use SAS/CONNECT to submit code to a SAS Viya environment. For more information, see *SAS/CONNECT User's Guide*.

Connecting to the CAS Server

To work with the SAS Viya environment in SAS Enterprise Guide, your first step is to create a connection.

1 Open a new program window by selecting **File** ⇨ **New** ⇨ **Program**.

2 To create a connection to the CAS server, enter this code:

```
options cashost="cas-server-name" casport=port-number"; cas;
```

3 Click **Run**.

Creating a CAS Session

In SAS Cloud Analytic Services, sessions are used to enable clients to communicate with the server to request actions. To use CAS, you explicitly start a session with the CAS server.

You use the CAS statement to create a session. Run this code in the Program Editor:

```
cas myCasSession;
```

For more information, see the CAS Statement in *SAS Cloud Analytic Services: User's Guide*.

Disconnecting from a CAS Session

To save resources, disconnect from your CAS session after you have finished your work. Run this code in the Program Editor:

```
cas myCasSession disconnect;
```

Configuring the Server Connection

Examining Your Metadata Server Connection

The Connections window displays which metadata server you are currently connected to and includes information about that connection. You can also connect to a metadata cluster, which contains multiple servers. Any specifications that you make in the Connections window are valid for your current and future SAS Enterprise Guide sessions.

Note: Metadata configuration information is shared between SAS Enterprise Guide and SAS Add-In for Microsoft Office. Any changes that you make to your connection in SAS Enterprise Guide also appear in SAS Add-In for Microsoft Office.

To examine your connection:

1. From the main menu in SAS Enterprise Guide, select **Tools** ⇨ **Connections**. The Connections window appears.
2. From the selection pane in the Connections window, select the following items:
   - **Profiles** to add, modify, or delete a profile. You can also specify the active profile. For more information, see “Managing Profiles” on page 9.
   - **Servers** to specify the default server for a profile. For more information, see “Selecting Your Default Server” on page 10.
   - **Updates** to specify whether to automatically update the configuration. For more information, see “Updating Your Configuration” on page 11.

Managing Profiles

A profile defines the connection between SAS Enterprise Guide and a SAS Metadata Server. The SAS Metadata Server contains metadata definitions for objects such as workspace servers, libraries, and users that are used by SAS Enterprise Guide and other SAS applications. The primary application for creating metadata definitions is SAS Management Console, which you can also use to define libraries, as well as servers, users, groups, and authorization settings.

Note: Metadata configuration information is shared between SAS Enterprise Guide and SAS Add-In for Microsoft Office. Any changes that you make to your profile in SAS Enterprise Guide also appear in SAS Add-In for Microsoft Office.

If you use SAS Enterprise Guide without a profile, you can access only a SAS server installed on your local machine. You cannot define libraries or share SAS resources with other SAS applications.

Using SAS Enterprise Guide, you can perform these tasks:

- create a profile
- select the active profile

Defining Workspace Servers

In SAS Enterprise Guide, you can connect directly to a SAS 9 or SAS Viya workspace server. You define these workspace servers in the Connections window.

To add a workspace server:

1 From the main menu in SAS Enterprise Guide, select **Tools** ⇨ **Connections**. The Connections window appears.

2 In the selection pane, select **Servers**.

3 Click **Add**.

4 Specify the configuration of the workspace server and click **Save**.

The new workspace server appears in the **Additional SAS Workspace Servers** pane.

Note: In SAS Viya 3.5, TLS and SSL are automatically configured for the SAS object spawner. To connect to the SAS Viya workspace server from SAS Enterprise Guide, you must install the appropriate certificates. For more information, see “TLS Support for the SAS Object Spawner (Programming-Only Deployment)” in *SAS Viya Administration: Programming Run-Time Servers* and “Add Your Certificates to the Windows CA Store” in *Encryption in SAS*.

Selecting Your Default Server

Using the Connections window, you can specify which SAS Workspace Server you want to access by default.

To select a default server:

1 From the main menu in SAS Enterprise Guide, select **Tools** ⇨ **Connections**. The Connections window appears.

Note: If you are working in SAS Enterprise Guide Explorer, select **File** ⇨ **Manage Profiles** to open the Connections window.

2 In the selection pane, select **Servers**.

3 Select the default server for the active profile.

4 To automatically add the local SAS server to the list of servers, select the **Automatically add local SAS server (if installed) to server list** check box.

5 Click **Close** to save your changes and close the Connections window.

Updating Your Configuration

In the Connections window, you can specify whether to automatically update the configuration for your connection. You can save your profiles in a configuration update file that is stored in a network location. If your version of SAS Enterprise Guide uses automatic updating, then SAS Enterprise Guide checks the contents of the configuration update file whenever the application starts. If you have made any changes to the profiles, those changes automatically appear in SAS Enterprise Guide.

To automatically update the configuration of your connection:

1 From the main menu in SAS Enterprise Guide, select **Tools** ⇨ **Connections**. The Connections window appears.

Note: If you are working in SAS Enterprise Guide Explorer, select **File** ⇨ **Manage Profiles** to open the Connections window.

2 In the selection pane, select **Updates**. The Updates pane opens.

3 Select the **Update configuration automatically** check box.

4 In the **Location of update file** field, enter the location of the update file.

5 To update the configuration, click **Check for updates now**.

6 Click **Close** to save your changes and close the Connections window.

Working with a SAS Grid

Using SAS Enterprise Guide with a SAS Grid

A SAS grid enables the workload from SAS Enterprise Guide to be distributed among multiple computers on a network, all under the control of SAS Grid Manager. SAS Grid Manager provides load balancing, policy enforcement, efficient resource allocation, prioritization, and a highly available analytic environment. SAS Grid Manager also separates the SAS applications from the infrastructure that is used to execute the applications.

You can use any or all of these methods to enable SAS Enterprise Guide to use a SAS grid.

Method	Description
Using server-side load balancing through a workspace server	You can convert a workspace server to use load balancing, and then specify a SAS grid as the balancing algorithm. When you send jobs from SAS Enterprise Guide to this workspace server, the server automatically sends the job to the least busy node in the grid.
Configuring a project to run on the grid	You can select the option **Use grid if available** in the properties window for a project. This option specifies that the project code is automatically run on a SAS grid if a SAS grid has been configured. For more information, see “Enabling Projects to Run on a SAS Grid” on page 41.
Configuring a task to run on the grid	You can select the option **Use grid if available** in the properties window for a task. This option specifies that the project code for the selected task is automatically run on a SAS grid if a SAS grid has been configured. For more information, see “Using a Grid Computing Environment” on page 528.
Specifying a value of Force for the EGGridPolicy extended attribute on the grid logical server	Using this parameter causes output from SAS Enterprise Guide to be automatically sent to the grid.

Note: If you specify that a project or a task should run on the grid, you can also use the **Allow parallel execution on the same server** option in the appropriate properties window to specify that each task use a separate grid node. Running tasks on multiple grid nodes enables you to take advantage of the parallel process capabilities of the SAS grid. However, you must plan for access to shared data and libraries between the tasks. For more information, see "Using SAS Enterprise Guide with Parallel Execution" on page 13.

For more information about the SAS grid computing environment, see Scalability and Performance Community and *Grid Computing in SAS*.

Using SAS Enterprise Guide with Parallel Execution

If you select the **Use grid if available** option in the properties window for a project or a task, the code from SAS Enterprise Guide is run on a SAS grid. However, the **Allow parallel execution on the same server** option changes how jobs run on the grid. For more information, see Code Submission Project Properties on page 39.

- If you do not select the **Allow parallel execution on the same server** option, all code from the project or task always runs on the same grid node. The project or task establishes a connection to a workspace server, and the workspace server connects to a grid node. Any further projects and tasks use the same connection and the same grid node. This configuration enables the grid to provide load balancing, but not the other grid benefits (such as distributed computing). Any macro variables or temporary data that is created on the grid node by a task is available for later tasks to use.
- If you select the **Allow parallel execution on the same server** option, each task that can be run separately establishes a new connection to a workspace server, which then connects to the SAS grid. These separate connections enable each parallel task to run on separate grid nodes. However, because each task can be in a separate grid session, data that is written to the Work library or libraries that are assigned by a task might not be available for later tasks. If grid tasks need to share data, you can pre-assign a common library in each grid node and ensure that all shared data is written to that library.

Connecting to a SAS Grid

When you connect to a SAS grid, the Gridwork library is automatically assigned for you. You can use Gridwork for work that you do on the grid, such as storing output data. The Gridwork library is assigned when a SAS grid is initialized, which usually occurs the first time you run a task on the grid. However, if you need to access the library earlier, there are two ways to ensure that the library is assigned before you run the first task on the grid:

- Select the **Initialize grid (if available) when connecting to a workspace** option from the SAS Programs page of the Options dialog box. If you select this option, a SAS grid connection is automatically initialized and the Gridwork library is assigned when a workspace server connection is made. For more information, see “Setting SAS Program and Editor Options” on page 507.

- In the **Servers** pane, right-click the appropriate workspace server and select **Connect To Grid**. When you select this option, SAS Enterprise Guide establishes the connection to the grid and the Gridwork library is assigned so that it is available for you to use.

3

Using SAS Enterprise Guide

About the Navigation Area

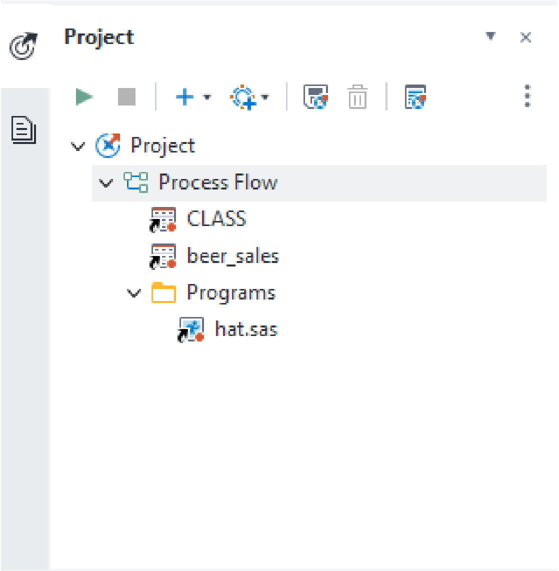
You can change the panes that are displayed in the navigation area by clicking the buttons on the left edge of the application. You can display multiple panes in the navigation area at one time, and you can hide one or more of the panes.

Using the Project Pane

You can choose to organize your work in SAS Enterprise Guide in projects, or you can work with your files without a project. If a project is open, any items that you open (such as data, code, or results) are automatically added to the project. Any items that you open before creating the project are not added to the project. For more information, see Chapter 4, “Working with Projects,” on page 27.

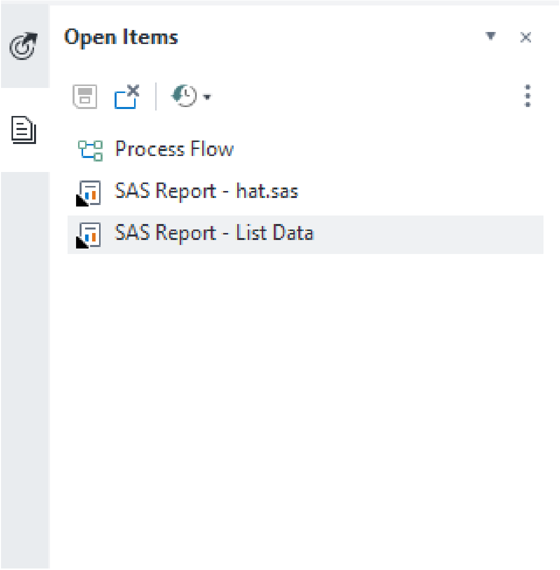
The Project pane displays the active project and its associated data, code, notes, and results. You can use the Project pane to open items in the work area. Items in a project are automatically added to a process flow. You can use the process flow to manage the relationship between objects. For more information, see “Working with Process Flows” on page 50.

To save the entire project, including embedded files and externally referenced files, click 💾 in the Project pane. External files that are not included in the project are not saved.



Accessing Your Open Files

The Open Items pane of the navigation area enables you to quickly view and access all the files that you have opened in your current SAS Enterprise Guide session. To access a file from the list of open files, click the file that you want to view. You can rearrange the items by right-clicking an item and selecting **Move up** or **Move down**. You can also save changes to all files in the list by clicking ⋮ and selecting **Save all**.



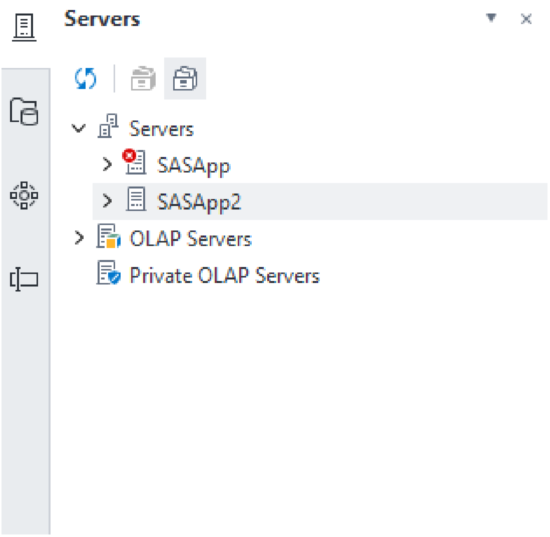
Accessing Your SAS Servers

The Servers pane lists the servers that are known to SAS Enterprise Guide. You can use this list to access the servers and navigate their file structures. The server that SAS Enterprise Guide is connected to is marked with a check mark.

You can perform various server and file management tasks by right-clicking items in the list (servers, folders, libraries, data sets, and files). For example, you can reset the server connection credentials that are stored in metadata so that the user must re-enter the credentials for that server. You can also manage the libraries on a server by modifying the library properties.

To refresh, disconnect, or connect a server, right-click the server name and use the options in the pop-up menu.

To add, delete, or change the properties of a server, contact your system administrator.



Accessing Your SAS Folders

The SAS Folders pane displays a list of the SAS folders that you can access. SAS folders are defined in the SAS Metadata Server and can be used to provide a central location for your stored processes, information maps, and projects so that they can be shared with other SAS applications. SAS folders can also contain content that is not in the SAS Metadata Server, such as data files. You can use the SAS Folders pane to add items to your project or to open a different project.

Note: To change the type of item that is displayed in the list, select an option from the **Show** drop-down list.

You can perform the following tasks in the SAS Folders pane:

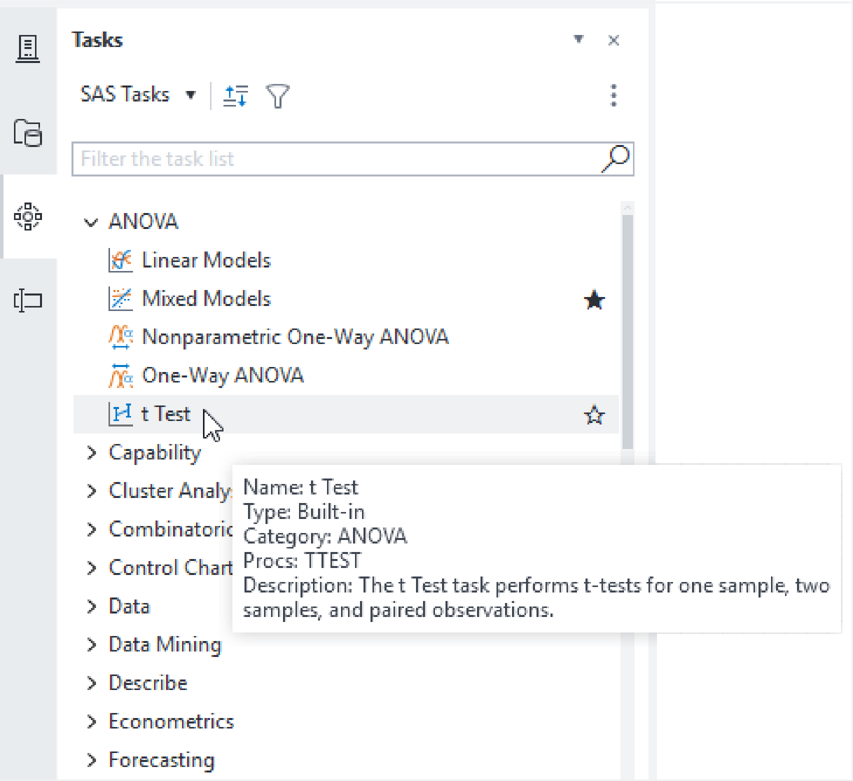
- To refresh the entire SAS Folders tree, click ⟳.
- To create a SAS folder, click 📁 and select **New folder**. You can also select **New top-level folder** to create a top-level folder in the foundation repository. These options are available only if you have the appropriate permissions to create a folder.
- To delete a SAS folder, select the folder that you want to delete and click 🗑. This option is available only if you have the appropriate permissions to delete a folder.

Working with Tasks

The Tasks pane lists the SAS tasks that you can run on your data. This list of tasks can include any tasks that are shipped with SAS Enterprise Guide, any custom tasks, and any tasks that are shipped with SAS Studio. For more information, see the **Display SAS Studio tasks** option in "Setting Task Options" on page 504. Any task templates that you have created also appear in the task pane.

To learn more about a task, place your mouse pointer over the task name. From the tooltip window, you can view the following information:

- whether the task is a built-in task (in other words, a task that shipped with SAS Enterprise Guide), a custom task, or a SAS Studio task
- the category for the task
- the SAS procedures that are used in the task
- a brief description of the task



Name: t Test
Type: Built-in
Category: ANOVA
Procs: TTEST
Description: The t Test task performs t-tests for one sample, two samples, and paired observations.

You can mark your favorite tasks by positioning your mouse pointer over a task and clicking the star on the right. When you are viewing the task list, you can use the drop-down list on the toolbar to display all tasks, only your favorites, or only your recently used tasks.

To sort the task list by category or name, click ⇅ and select the appropriate option.

You can filter the tasks by entering a value in the filter box. For example, enter "Model" to display only tasks that include "Model" in the name. You can also click ▽ to toggle the filter panel. The filter panel enables you to filter the task list by one or more task categories, task types, and associated SAS procedures.

Using the Prompt Manager

From the Prompt Manager pane, you can create, edit, and delete prompts. Prompts enable you to request input from the user when code is run. You can define prompts that require the user to select or enter one or more values when a query, program, SAS task, or stored process is run. For more information, see "About Prompts" on page 372.

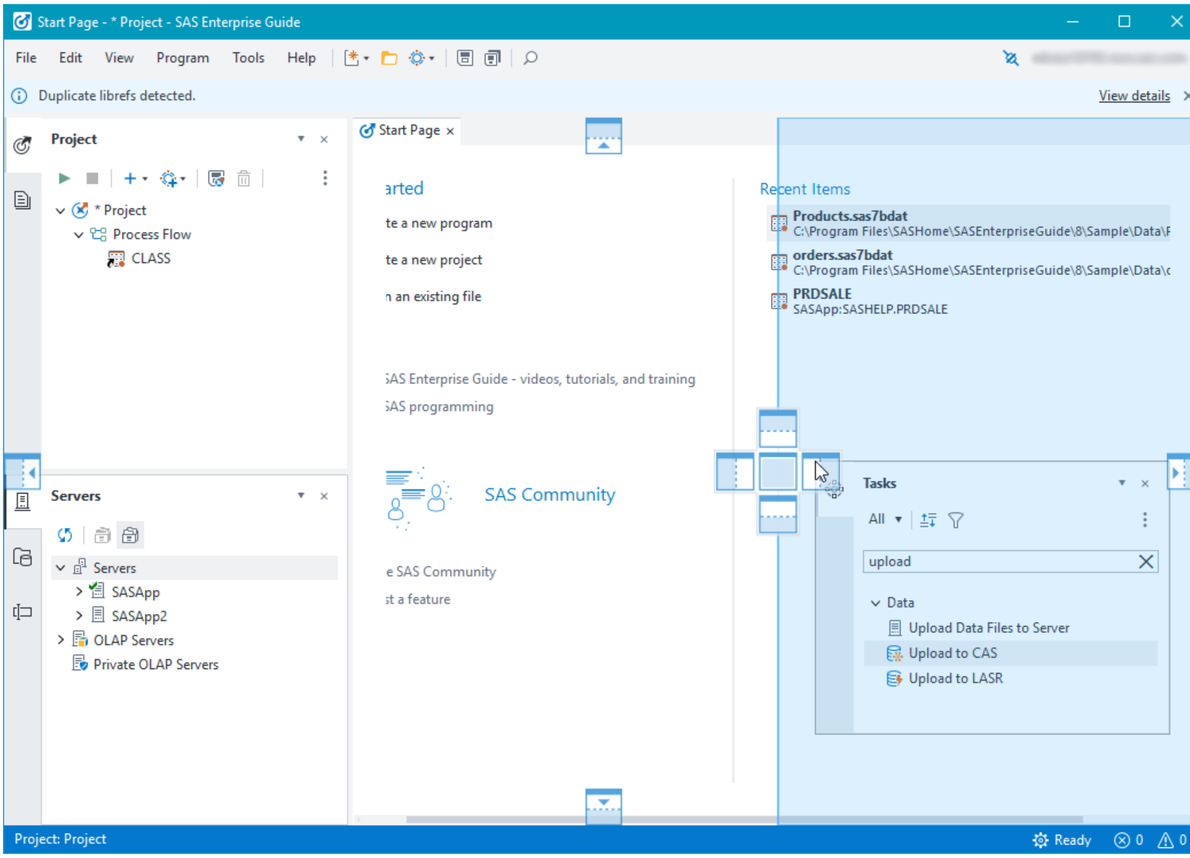
You can also use the Prompt Manager to view the macro variables that are associated with a prompt.

Note: The Prompt Manager is available only if you create a project.

Customizing the Navigation Area

You can display multiple panes in the navigation area at one time. By default, the Project pane is displayed in the upper left corner of the SAS Enterprise Guide application, and it provides access to the project and its associated data, code, notes, and results. The Servers pane is displayed by default in the lower left corner of the application, and it displays a list of your available SAS servers. Use the options in the **View** menu to display or hide panes.

You can also rearrange the panes by dragging them to another area of the application window. The positioner icon appears as you drag a pane to help you see where you can dock the pane. To move a single pane, click and drag the pane icon. To move a group of panes, click and drag the pane title bar.



To float a pane, click ▼ above the pane toolbar and select **Float**.

To hide a pane, click ▼ above the pane toolbar and select **Hide**.

Note: To reset SAS Enterprise Guide to its default appearance with the Project and Servers panes on the left and all other panes closed, select **View** ⇨ **Reset to default layout**.

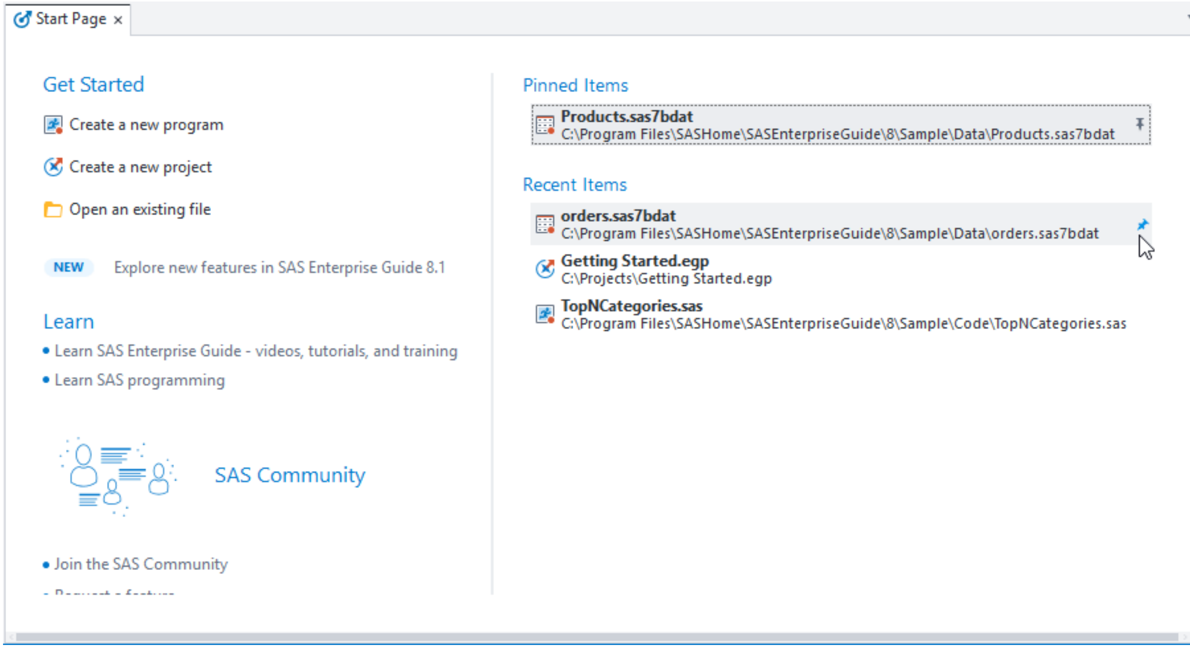
About the Work Area

The work area is the main area of the SAS Enterprise Guide application and is used to display your data, code, logs, task results, and process flows. The work area is always displayed and cannot be minimized.

Note: You can maximize the work area by selecting **View** ⇨ **Full screen**. To return to the default view, click **Restore**.

When you first open SAS Enterprise Guide, the Start Page tab is displayed in the work area by default. The Start Page provides quick access to a few basic tasks (for example, creating a program or project), a list of your recently viewed items, and links to learning resources. You can click the pin icon beside an item to ensure that the item always appears at the top of the list. If you close the Start Page tab, you can re-open it by selecting **View** ⇨ **Start Page**. You can use the **Show Start Page** option to control whether to display the Start Page tab each time you start SAS Enterprise Guide. For more information, see “Setting Start-Up Options” on page 482.

Note: The Open and Save windows also include a list of recently viewed items and locations.



Start Page ×

Get Started

- Create a new program
- Create a new project
- Open an existing file

NEW Explore new features in SAS Enterprise Guide 8.1

Learn

- Learn SAS Enterprise Guide - videos, tutorials, and training
- Learn SAS programming

SAS Community

- Join the SAS Community
- Request a feature

Pinned Items

Products.sas7bdat
C:\Program Files\SASHome\SASEnterpriseGuide\8\Sample\Data\Products.sas7bdat

Recent Items

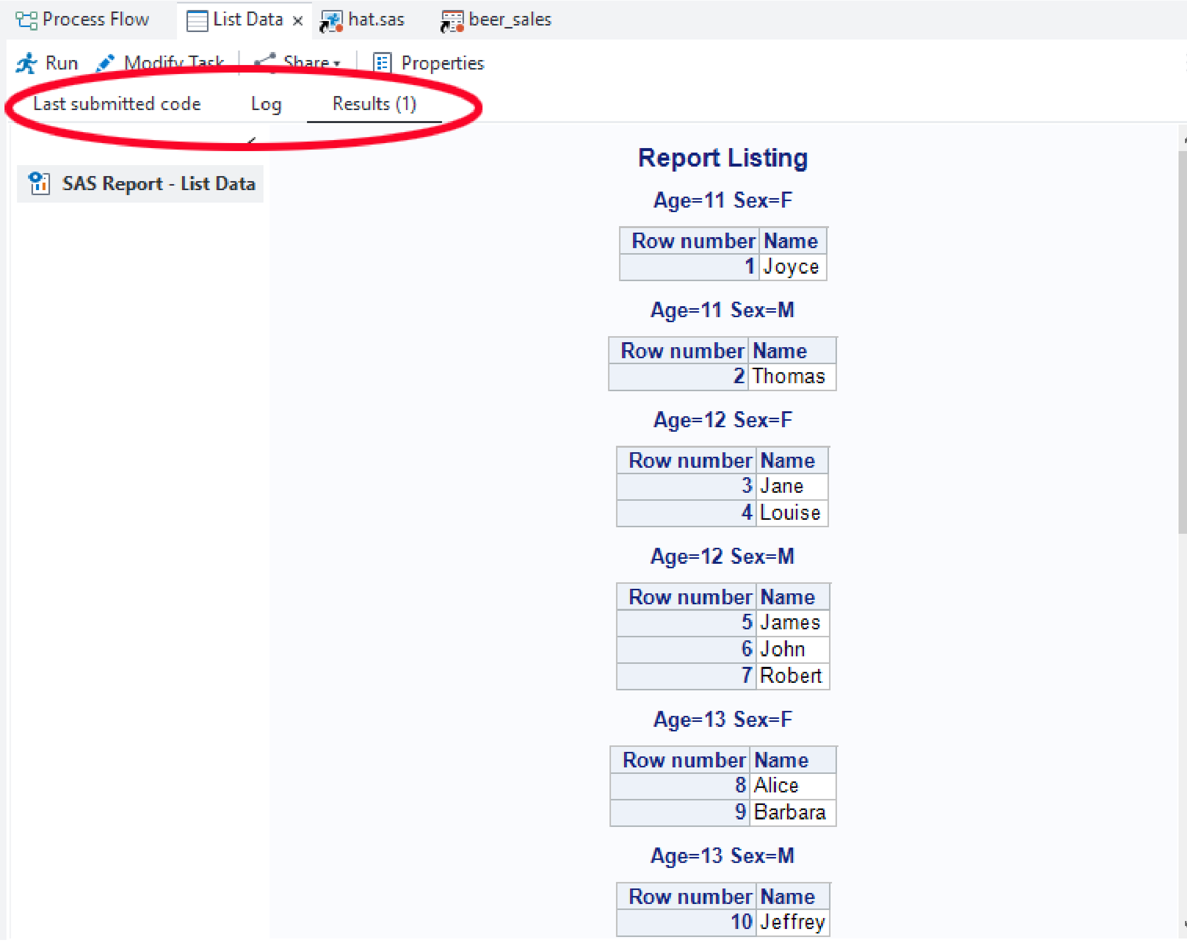
orders.sas7bdat
C:\Program Files\SASHome\SASEnterpriseGuide\8\Sample\Data\orders.sas7bdat

Getting Started.egp
C:\Projects\Getting Started.egp

TopNCategories.sas
C:\Program Files\SASHome\SASEnterpriseGuide\8\Sample\Code\TopNCategories.sas

In the work area, your data, code, task and query results, and process flows are displayed as tabs. When you select a task or a query in the work area (or in the Project pane), the code, log, and output of the task are displayed as sub-tabs. When you run a task or a query, the results or output data tab is displayed by default.

This screen shot shows the work area for the List Data task and the location of the three sub-tabs. By default, the results are displayed. However, you can select **Last submitted code** or **Log** to view the contents of those tabs.



Report Listing

Age=11 Sex=F

Row number	Name
1	Joyce

Age=11 Sex=M

Row number	Name
2	Thomas

Age=12 Sex=F

Row number	Name
3	Jane
4	Louise

Age=12 Sex=M

Row number	Name
5	James
6	John
7	Robert

Age=13 Sex=F

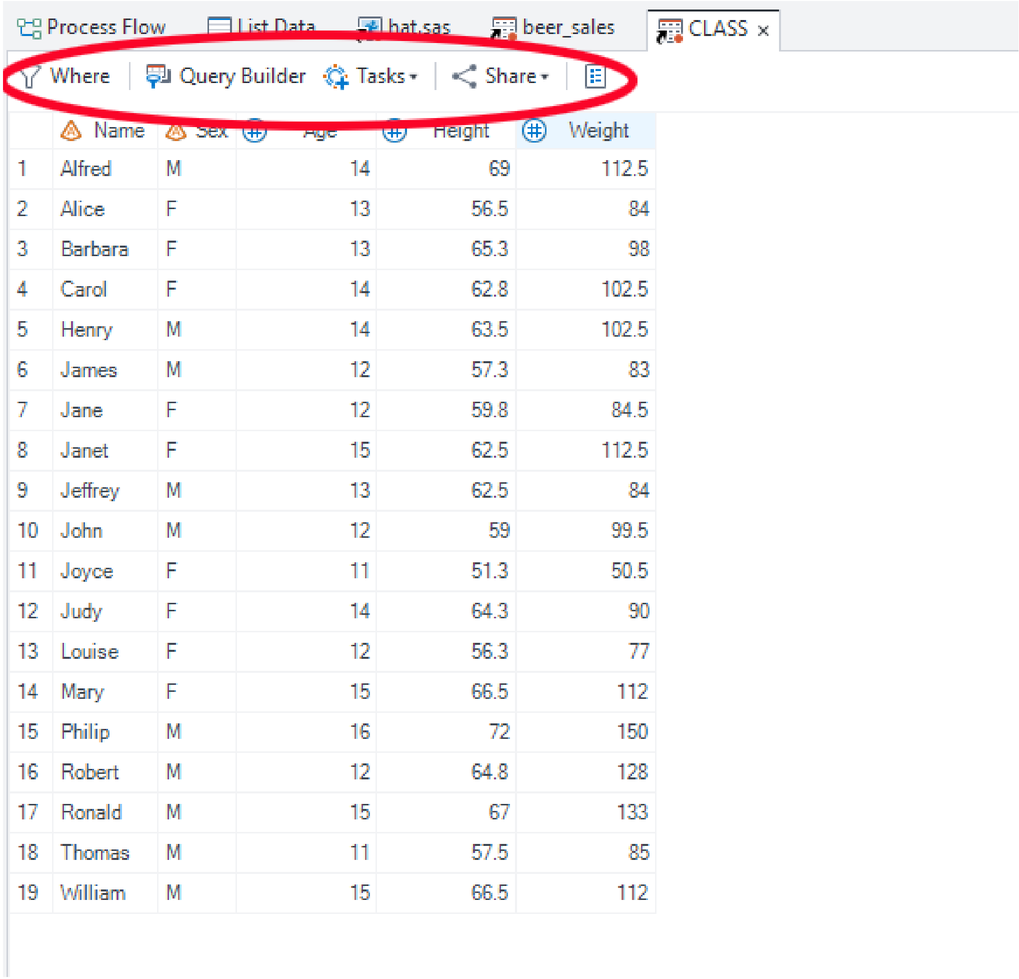
Row number	Name
8	Alice
9	Barbara

Age=13 Sex=M

Row number	Name
10	Jeffrey

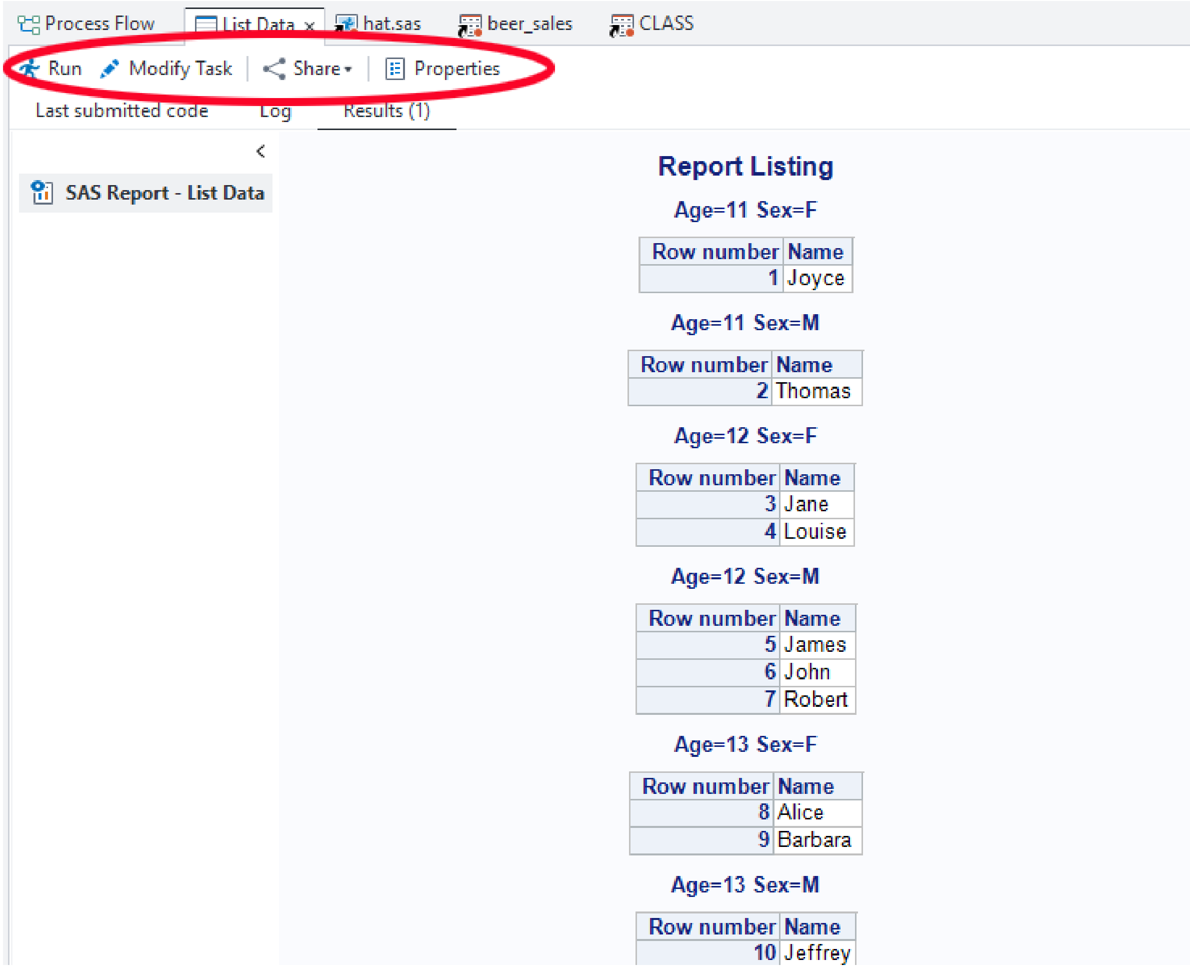
The work area contains a context-sensitive toolbar that changes depending on what you are viewing in the work area. For example, when you are viewing data, this toolbar includes options to filter the data, create a query or run tasks based on the data, share the data, and view the data properties.

This screen shot shows the context-sensitive toolbar for the Sashelp.Class data set.



	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84
10	John	M	12	59	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90
13	Louise	F	12	56.3	77
14	Mary	F	15	66.5	112
15	Philip	M	16	72	150
16	Robert	M	12	64.8	128
17	Ronald	M	15	67	133
18	Thomas	M	11	57.5	85
19	William	M	15	66.5	112

However, if you are viewing task results, the options change to include options to run the task, modify the task that created the results, export the results, and so on. This screen shot shows the context-sensitive toolbar for the List Data task.



Report Listing

Age=11 Sex=F

Row number	Name
1	Joyce

Age=11 Sex=M

Row number	Name
2	Thomas

Age=12 Sex=F

Row number	Name
3	Jane
4	Louise

Age=12 Sex=M

Row number	Name
5	James
6	John
7	Robert

Age=13 Sex=F

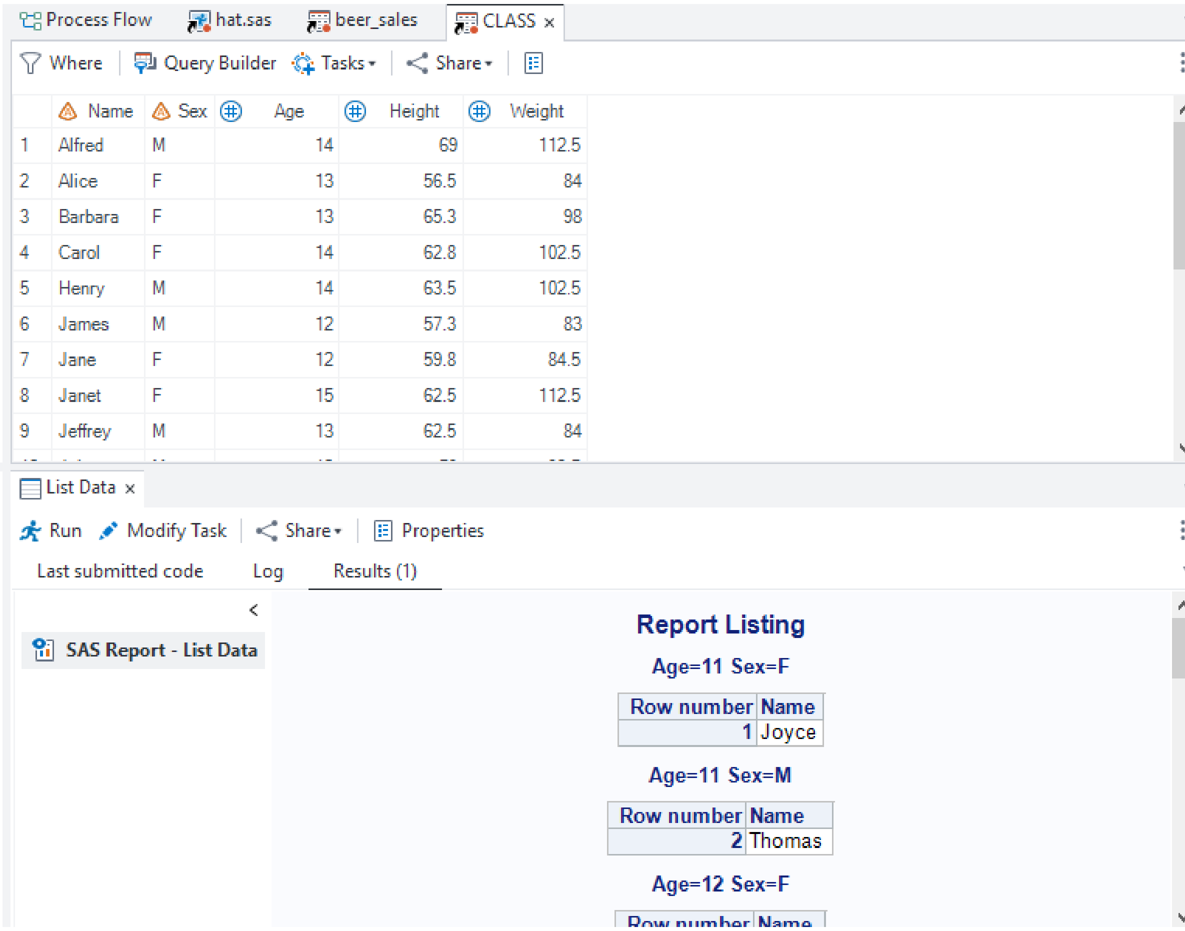
Row number	Name
8	Alice
9	Barbara

Age=13 Sex=M

Row number	Name
10	Jeffrey

By default, the work area displays one item at a time from the Project pane. You can change the layout of the work area by displaying two items at a time. If you display two items at a time, you can display them vertically or horizontally. To display two items at a time, select **View** ⇨ **Apply tab layout**.

This screen shot shows two tabs (one for the Class data set and one for the List Data task) displayed horizontally.



	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84

Report Listing

Age=11 Sex=F

Row number	Name
1	Joyce

Age=11 Sex=M

Row number	Name
2	Thomas

Age=12 Sex=F

Row number	Name

Understanding Role-Based Settings in SAS Enterprise Guide

Because SAS Enterprise Guide can be used by a variety of people and groups within an organization, your administrator might choose to restrict the menu items and functionality that users can access. By default, SAS Enterprise Guide is shipped with several default roles. These roles manage the availability of application features, or capabilities.

By using SAS Management Console, your administrator for SAS Enterprise Guide can modify these roles and specify the capabilities that are available to users at your site. For example, your administrator might want to restrict who can save content to SAS Folders or who can open OLAP cubes. As a result, all of the capabilities that are documented in the Help might not be available to you.

To view the capabilities that are available to you, open the Connections window and click **View Capabilities**. If all capabilities are available to you, then this option is not available. To enable a capability in your copy of SAS Enterprise Guide, contact your administrator for SAS Enterprise Guide.

Using Autoexec Processing in SAS Enterprise Guide

An autoexec file in SAS contains SAS statements that are run automatically when you start or connect to SAS. When you use SAS Enterprise Guide, there are two stages of autoexec processing that occur in the following order:

1. SAS autoexec file - Typically named "autoexec.sas", this file usually resides in your SAS application directory. If you use SAS Enterprise Guide to connect to SAS on a remote server, the SAS administrator maintains the contents of the autoexec.sas file, if one exists.
2. SAS Enterprise Guide built-in definitions - SAS Enterprise Guide submits a few macro definitions and system options to improve how SAS Enterprise Guide works with SAS. For example, SAS Enterprise Guide sets the VALIDVARNAME=ANY system option to allow SAS to work with nonstandard variable names.

Beyond this built-in processing, you can customize the "start-up sequence" within SAS Enterprise Guide by using any combination of these three mechanisms:

- **Submit SAS code when server is connected** – You can write custom SAS code to be submitted whenever a workspace server session is created on your SAS server by using the **Submit SAS code when server is connected** option in the Options window. For example, you could use this option to create some temporary data sets in the Work library each time you connect to a new server. For more information, see “Setting SAS Program and Editor Options” on page 507.
- **Autoexec process flow** – You can create a process flow named "autoexec" that can be run automatically whenever you open your project. For example, you could use this process flow to define libraries or to assign macro variables for the project. For more information, see “Running a Process Flow When You Open a Project” on page 59.
- **Submit SAS code before or after you run a SAS task or program** – You can write custom SAS code and have it automatically inserted either before or after programs that you run in the Program Editor. For more information, see “Setting SAS Program and Editor Options” on page 507.

  You can also choose to insert custom SAS code before or after the automatically generated task or query code.

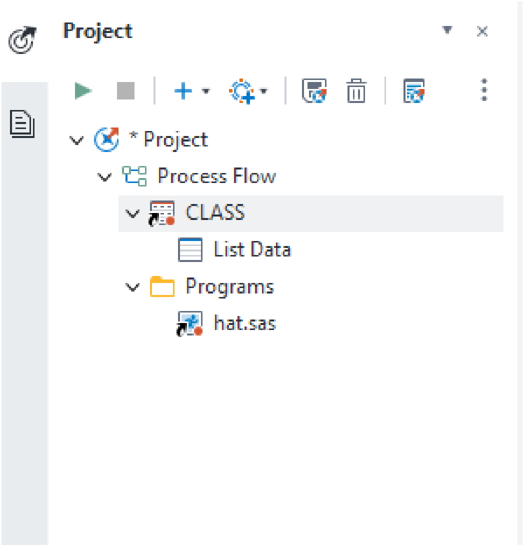
4

Working with Projects

What Is a Project?

SAS Enterprise Guide uses projects to manage collections of related data, tasks, code, and results. With projects, you can run multiple tasks on the same group of data files, and you can save a project to run later in batch mode. When you create a project, you can organize and share your work.

In the navigation area, the Project pane displays the active project and its associated data, code, notes, and results. The Project pane is displayed by default, and you can use it to open items in the work area. For more information, see “Using the Project Pane” on page 15.



Project
- * Project
  - Process Flow
    - CLASS
      - List Data
    - Programs
      - hat.sas

Note: A project is not required to run a program or work with data.

Creating a Project

You can create a project either when you first start up SAS Enterprise Guide or when you close the open project and start another one. You can have only one project open at a time in SAS Enterprise Guide.

To start a new project when you invoke SAS Enterprise Guide, click **Create a new project** on the Start Page.

To create a project when SAS Enterprise Guide is already running, select **File** ⇨ **New** ⇨ **Project**.

Opening an Existing Project

You can open an existing project either when you first start up SAS Enterprise Guide or when you close the current project and open another one. You can have only one project open at a time in SAS Enterprise Guide.

Note: If you open a SAS Enterprise Guide project that was created in version 4.1 or later in the current version of SAS Enterprise Guide, it is automatically migrated to the current version's file format. If you need to migrate a project file from SAS Enterprise Guide 3.0 or earlier, you must first migrate it to SAS Enterprise Guide 4.1, and then migrate it to the current version of SAS Enterprise Guide. For more information, see "Migrating SAS Enterprise Guide Project Files" on page 33.

When you invoke SAS Enterprise Guide, you can open a project from the Start Page. Select the project from the **Recent Items** list, or select **Open an existing file**.

To open an existing project when SAS Enterprise Guide is already running, select **File** ⇨ **Open**, and then select the location where the project resides.

Using the Project Log

Turn on the Project Log

By default, the project log is turned off. You must enable SAS Enterprise Guide to maintain a project log before you can open and view the aggregated project log.

To turn on the project log for the current project, click 📋 in the Project pane. Click **Turn On** to start project logging for the current project.

Note: Using this option turns on the project log only for the current project. By default, when you create a project, the project log is not on.

To turn on the project log for all new projects:

1 Select **Tools** ⇨ **Options**.

2 In the navigation pane, select **Project and Process Flows**.

3 Select **Enable project log for new projects** and click **OK**.

About the Project Log

To view the project log, click 📋 in the Project pane.

The project log displays an aggregated list, or log summary, of all the messages (errors, warnings, and notes) that have been generated. The **Errors**, **Warnings**, and **Notes** buttons include the total number of errors, warnings, and notes for the project. The project log also includes an aggregated log of task and code activity for the entire project.

By default, the log summary messages for the first task or code activity are expanded. Use the **Filter** drop-down list to display the log and messages for a different individual log entry. You can also filter the messages by selecting or deselecting a type of message. For example, to display only the error messages, deselect **Warnings** and **Notes**.

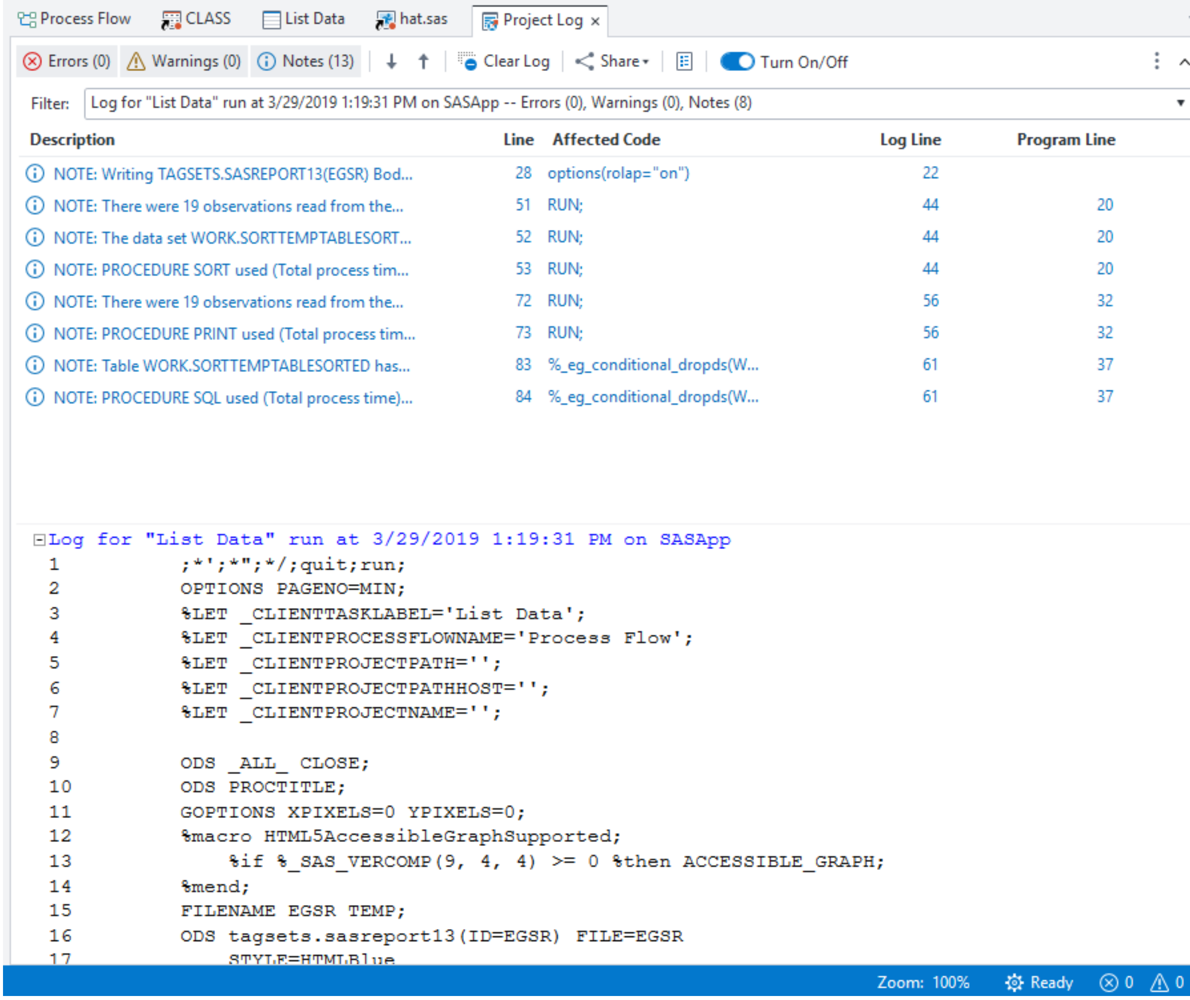
Note: You can collapse, expand, and pin the log summary area by using the buttons on the right end of the toolbar. You can also expand and collapse the log summary area by double-clicking the **Errors**, **Warnings**, or **Notes** buttons.

Note: To navigate among the messages that are displayed, click ↓ and ↑.

Each error, warning, or note includes this information:

- the text of the message.
- the line number in the Log window where the message is displayed.
- the program code that is associated with the message.
- the log line number in the Log window that is associated with the message. The log line numbers identify the lines in the log that are associated with original program statements.
- the line number in the Program window that is associated with the message.

To get more information about a specific message, right-click the message and select **Search online help**.



Filter: Log for "List Data" run at 3/29/2019 1:19:31 PM on SASApp -- Errors (0), Warnings (0), Notes (8)

Description	Line	Affected Code	Log Line	Program Line
NOTE: Writing TAGSETS.SASREPORT13(EGSR) Bod...	28	options(rolap="on")	22	
NOTE: There were 19 observations read from the...	51	RUN;	44	20
NOTE: The data set WORK.SORTTEMPTABLESORT...	52	RUN;	44	20
NOTE: PROCEDURE SORT used (Total process tim...	53	RUN;	44	20
NOTE: There were 19 observations read from the...	72	RUN;	56	32
NOTE: PROCEDURE PRINT used (Total process tim...	73	RUN;	56	32
NOTE: Table WORK.SORTTEMPTABLESORTED has...	83	%_eg_conditional_dropds(W...	61	37
NOTE: PROCEDURE SQL used (Total process time)...	84	%_eg_conditional_dropds(W...	61	37

```
⊟Log for "List Data" run at 3/29/2019 1:19:31 PM on SASApp
1          ;*';*";*/;quit;run;
2          OPTIONS PAGENO=MIN;
3          %LET _CLIENTTASKLABEL='List Data';
4          %LET _CLIENTPROCESSFLOWNAME='Process Flow';
5          %LET _CLIENTPROJECTPATH='';
6          %LET _CLIENTPROJECTPATHHOST='';
7          %LET _CLIENTPROJECTNAME='';
8
9          ODS _ALL_ CLOSE;
10         ODS PROCTITLE;
11         GOPTIONS XPIXELS=0 YPIXELS=0;
12         %macro HTML5AccessibleGraphSupported;
13             %if %_SAS_VERCOMP(9, 4, 4) >= 0 %then ACCESSIBLE_GRAPH;
14         %mend;
15         FILENAME EGSR TEMP;
16         ODS tagsets.sasreport13(ID=EGSR) FILE=EGSR
17             STYLE=HTMLBlue
```

Setting the Project Log Options

By default, if the project log is turned on, you see a warning when the log exceeds 5MB. Also, the project log is not automatically cleared when you close the project. You can change these options from the project properties.

To view the properties of the project log, click ⋮ in the project log and select **Project log options**.

For more information about these properties, see Table 4.2 on page 38.

Migrating SAS Enterprise Guide Project Files

If you open a SAS Enterprise Guide project that was created in version 4.1 or later in the current version of SAS Enterprise Guide, it is automatically migrated to the current version's file format. If you want to migrate multiple files at once, you can use the Migration Wizard for SAS Enterprise Guide and SAS Add-In for Microsoft Office. The wizard also enables you to remap references to metadata objects such as servers, libraries, information maps, and stored processes.

To run the wizard, run MigrationWizard.exe, which is located in your SAS Enterprise Guide installation directory.

If you need to migrate a project file from SAS Enterprise Guide 3.0 or earlier, you must first migrate it to SAS Enterprise Guide 4.1, and then migrate it to the current version of SAS Enterprise Guide.

Adding Files to a Project

You can add SAS data files and SAS programs to your SAS Enterprise Guide project, as well as existing text and HTML files. You can also add other types of files including OLAP cubes, information maps, ODBC-compliant data, and files that are created by other software such as Microsoft Word or Microsoft Excel.

When you add a file, it is automatically added to the current project that you have open. If a project is not already open, you might be prompted to create one. After you add the file, SAS Enterprise Guide automatically opens it.

Note: If you do not want SAS Enterprise Guide to automatically open your data after adding the file, you can clear the **When an item is opened, automatically add to the project** check box. For more information, see "Setting Project and Process Flow Options" on page 484.

When you add text, HTML, or PC-based database files (including Microsoft Excel, Microsoft Access, and other popular formats), you can choose to use the Import Data task, which converts the data files to SAS data sets. When you add other types of files that SAS Enterprise Guide might not be able to use as data, such as Microsoft Word files, they are inserted into the project as is.

You can open the file location of externally referenced files in Microsoft Windows Explorer by right-clicking a file in the Project pane and selecting **Open file location**.

Externally referenced files are files that are saved on your local computer or a remote server.

You can add existing files to your SAS Enterprise Guide project in several ways. You can use the menus, or you can drag the files from the **Servers** pane or from Windows Explorer.

To add files by using the menu, select **File** ⇨ **Open**, and then select the file that you want to add.

To add files by dragging and dropping:

1 If you are dragging files from Windows Explorer to SAS Enterprise Guide, position the Window Explorer window so that you can see both it and the SAS Enterprise Guide application window. Both of these must be visible before you can drag files between them.

2 Use the left mouse button to drag the file from Windows Explorer or the **Servers** pane into the SAS Enterprise Guide application. The mouse pointer changes to indicate where you can drop the file.

Copying Items in a Project

You can copy and paste items from a project in one SAS Enterprise Guide session to another SAS Enterprise Guide session. You can also drag project items from the Project pane of one SAS Enterprise Guide session to the Project pane of another SAS Enterprise Guide session. If you do not have a project open in the target SAS Enterprise Guide session, one is created for you.

You can also make copies of tasks within the same SAS Enterprise Guide project.

Here is what gets copied for each item:

Data

The reference to the data file. Another physical copy of the file is not made on the server.

Note: The reference to data files that you copy might not be valid in the target project if the SAS library in which the file is stored is not defined the same way in the target project. For example, a reference to a data set in the temporary library Work might not be valid from one project to another. The Work library is temporary and is available for use only in the SAS Enterprise Guide session in which it is created.

If you are copying an OLAP slice, the slice (or current view) of the OLAP cube is copied and the OLAP cube that is associated with the slice.

If you are copying output data, the output data is copied as a data item in the Project pane. The task or code that created the output data is not copied.

Queries
: The query item and the input data for the query, if it does not already exist in the project or process flow.

Tasks
: The task item and the input data for the task, if it does not already exist in the project or process flow.

Code
: The code item. No data is copied with the code. You cannot copy output code from tasks or queries.

Notes
: The note item.

External files
: The reference to the external file. Another physical copy of the file is not made on the server.

Note: External files include files that are not data, code, or note files, such as Microsoft Word files or Microsoft PowerPoint files.

Note: You cannot copy results items, log items, or items from the scheduler.

Creating Notes for Project Items

You can create a note to accompany a project or any file in the project. Notes help you keep track of things that you need to do and can give other users valuable information. Notes are saved as part of the project file.

You can link notes to one or more objects in a process flow. For more information, see "Annotating a Process Flow" on page 55.

Note: You cannot add a note to temporary items such as code, log, output data, or results items.

Searching in SAS Enterprise Guide

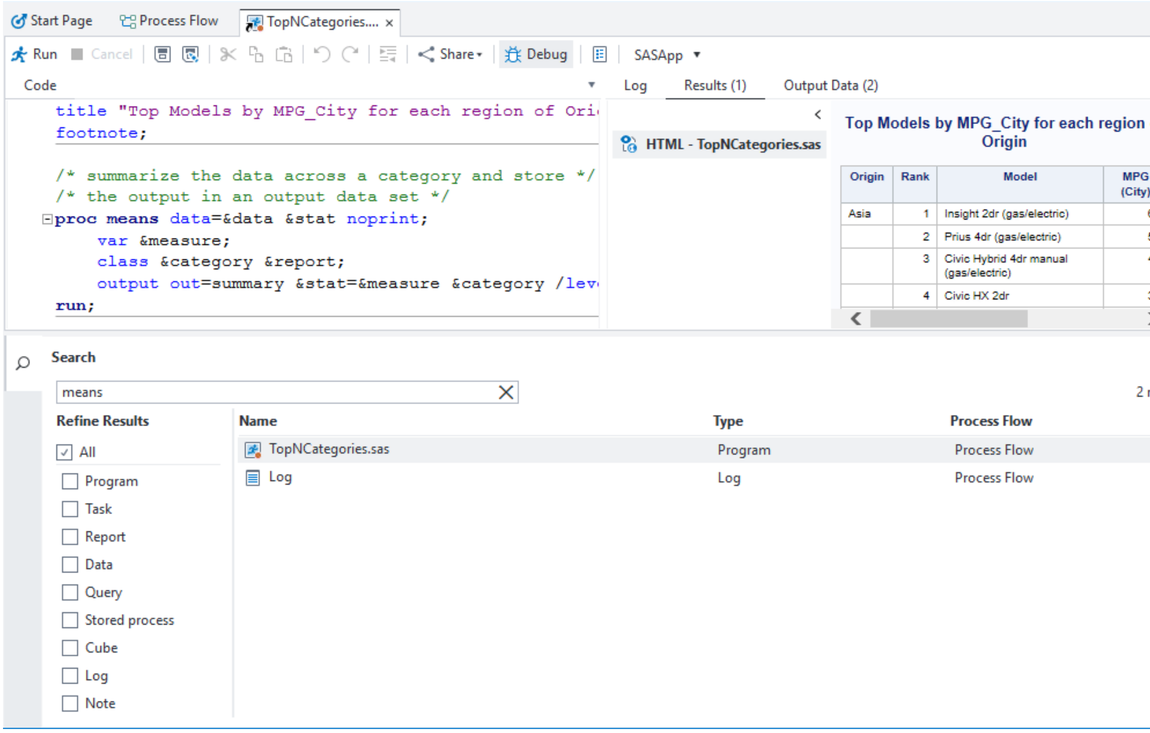
You can use the search feature in SAS Enterprise Guide to search for text that you specify. You can search for items in a project or in the Open Items pane. You do not have to have a project open to use the search feature. You can search for items by name, and you can search for text inside some items:

Item	What Is Searched
Program	Program name and code
Task	Task name and code
Report	Report name
Data	Data table name
Query	Query name and associated code
Stored process	Stored process name
Cube	Cube name
Log	Log name and code
Note	Note name and text

To search in SAS Enterprise Guide:

1 On the SAS Enterprise Guide toolbar, click 🔍. The Search pane opens in the work area.

Note: You can also open the Search pane by selecting **View** ⇨ **Search**.



```
title "Top Models by MPG_City for each region of Ori
footnote;

/* summarize the data across a category and store */
/* the output in an output data set */
proc means data=&data &stat noprint;
    var &measure;
    class &category &report;
    output out=summary &stat=&measure &category /lev
run;
```

Top Models by MPG_City for each region
Origin

Origin	Rank	Model	MPG (City)
Asia	1	Insight 2dr (gas/electric)	
	2	Prius 4dr (gas/electric)	
	3	Civic Hybrid 4dr manual (gas/electric)	
	4	Civic HX 2dr	

means

Refine Results

☑ All
☐ Program
☐ Task
☐ Report
☐ Data
☐ Query
☐ Stored process
☐ Cube
☐ Log
☐ Note

Name	Type	Process Flow
TopNCategories.sas	Program	Process Flow
Log	Log	Process Flow

2 Enter the text that you want to search for and click ⌕ or press Enter. The results are listed in the Search pane. You can perform the following tasks in the Search pane:

- To open an item from the search results, double-click the item.
- To narrow your search to include only specific types of items, select one or more items from the **Refine Results** list. By default, the search results include all types of items.

Viewing Project Properties

You can view information about your project, and you can add a password that is required to open your project.

To view the properties for a project, click ⋮ in the Project pane and select **Properties**.

The summary properties and metadata properties are for informational purposes only. You cannot edit any of these properties from this location.

Table 4.1 *Security*

Property Name	Description

Password to open	The password that is required to open the project. When you enter the password and click **OK**, you are prompted to re-enter the password for verification. You must save the project for this change to take effect. Note: Use this option with caution. If you forget the password that you have set for a project, you can no longer access that project.
Clear Password	Removes a password that you have set. You must save the project for this change to take effect.

Table 4.2 Project Log

Property Name	Description
Turn on project logging	Turns on the project log.
When the log reaches	Specifies the action to complete when the log reaches the specified size. The default size is 5 MB. You can choose from these options: ■ **Display a warning message** ■ **Remove older items** ■ **Export a time-stamped log, then clear**
Clear the log when the project closes	Clears the log when you close the project.

Table 4.3 File Reference

Property Name	Description
Use paths relative to project for programs and importable files	Uses pathnames that are relative to the current project location instead of fully referenced, absolute pathnames. This option enables you to move the project file to another location, and, as long as the programs and imported files are in the same location relative to the project file, SAS Enterprise Guide can still access the files. This option supports pathnames that refer to locations on mapped drives and drives that are referenced by Universal Naming Convention (UNC) paths. This option applies only to SAS program files and files that can be imported, such as Microsoft Excel files and text files. Note: This option is available only for projects that are saved on your local computer. This option does not apply to projects that are saved on a SAS server.

Table 4.4 *Output Data Sets*

Property Name	Description
Maximum number of output data sets to add to the project	Specifies the maximum number of output data sets that can be added to the project by a single task or program. If a task or program creates more than the maximum number that was specified, a note is added to the project. The note indicates that the data set limit was reached and specifies the total number of output data sets that were created. The default value is 50 output data sets.

Table 4.5 *Code Submission*

Property Name	Description
Use grid if available	Enables SAS Enterprise Guide to run tasks in a SAS grid computing environment. For more information, see "Using SAS Enterprise Guide with a SAS Grid" on page 12.
Allow parallel execution on the same server	Enables SAS Enterprise Guide to create one or more additional workspace server connections so that parallel process flow paths can be run in parallel on the same server. For more information, see "Using SAS Enterprise Guide with a SAS Grid" on page 12. **Note:** Because there might be more than one workspace server connection, any changes that you make in one workspace session, such as library assignments or changes to macro variables or options, might not be available to tasks that are running in another parallel workspace session.
Action to take on errors during execution	Specifies how to handle an error during the execution of items in a project or process flow. ■ **Stop all items** stops the execution of all subsequent items in the project or process flow. ■ **Stop current branch** stops the execution of all subsequent items in the current branch of the process flow. This is the default value. ■ **Continue execution** continues the execution of all subsequent items in the project or process flow.

Understanding Saving in SAS Enterprise Guide

You can choose whether to save only the active item in your SAS Enterprise Guide project or the entire project, including both embedded files and externally referenced files.

Note: Embedded files are stored inside the SAS Enterprise Guide project and cannot be accessed when the project is closed. When you save a project, any changes to the embedded files are saved as well.

Externally referenced files are files that are saved on your local computer or a remote server. The pathname of the externally referenced file is stored in SAS Enterprise Guide, and SAS Enterprise Guide writes changes to the file when it is saved. For example, you might have an externally referenced SAS program that you edit in SAS Enterprise Guide.

To save the active item in your project, press Ctrl+S.

Note: If the active item in your project is an embedded file, the entire project is saved.

To save the entire project, including embedded files and externally referenced files, press Ctrl+Shift+S.

Saving a Project

Depending on your access permissions, you can save an existing project to a local computer, a SAS server, or a SAS folder. By default, projects are saved as SAS Enterprise Guide Project Files (*.egp).

SAS folders provide a central location that enables you to share a project. When you save a project to a SAS folder in the metadata repository, information about all the data sources, stored processes, and the location of the project in WebDAV is saved in a metadata entry. The project file is saved in WebDAV. You can then use SAS Data Integration Studio to include the project in impact analysis for a table. Impact analysis identifies a data source and the items that are dependent on this source; this analysis reveals how changes to a data source could affect other documents.

Enabling Projects to Run on a SAS Grid

If your site has configured a SAS grid, you can specify that a SAS Enterprise Guide project run on the grid. Running a project on a SAS grid provides workload balancing among the grid machines based on job priority and resource allocation.

For more information, see "Using SAS Enterprise Guide with a SAS Grid" on page 12.

To run a project on a SAS grid:

1 In the Project pane, click ⋮ and select **Properties**. Click **Code Submission** in the selection pane.

2 Select **Use grid if available.** If a SAS grid has been configured, the project code is automatically sent to the grid for processing. If a SAS grid has not been configured, this option has no effect.

   Note: If you specify that a project should run on the grid, you can also use the **Allow parallel execution on the same server** option to specify that each submission use a separate grid node. This option is also available in the **Code Submission** panel of the project properties window. Running tasks on multiple grid nodes enables you to take advantage of the parallel process capabilities of the SAS grid. However, you must plan for access to shared data and libraries between the tasks. For more information, see "Using SAS Enterprise Guide with Parallel Execution" on page 13.

   Note: You can also configure individual tasks to run on the SAS grid. For more information, see "Enabling Tasks to Run on a SAS Grid" on page 529.

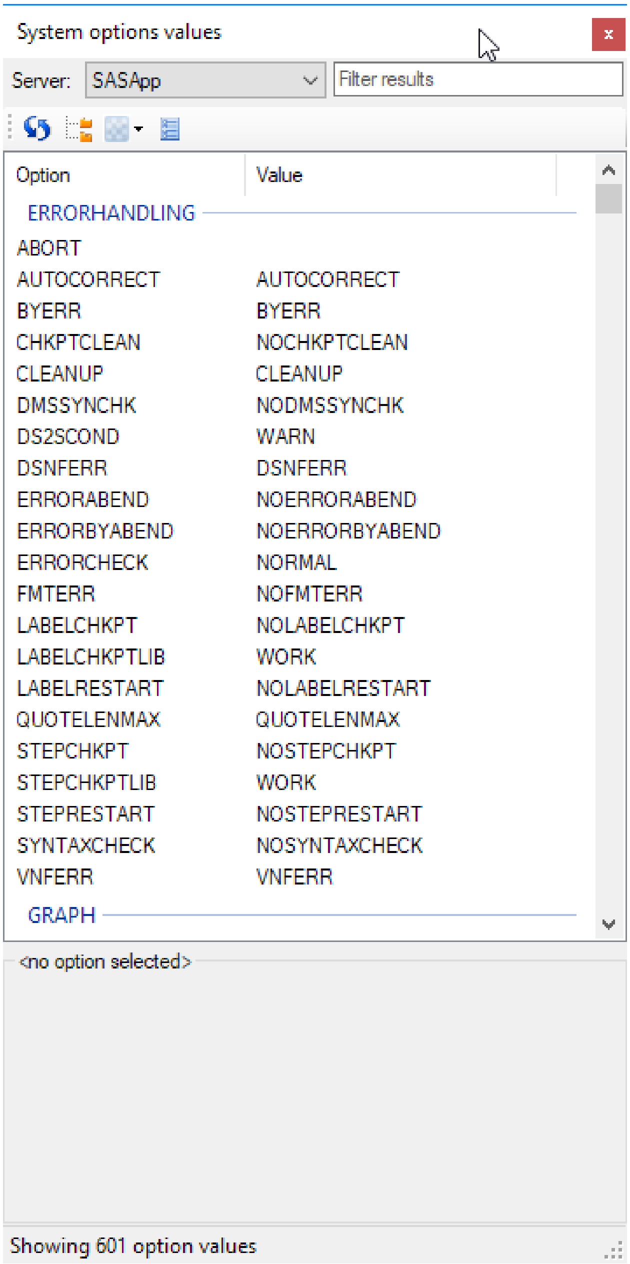
For more information about the SAS grid computing environment, see Scalability and Performance Community and the documentation for SAS Grid Manager.

Running a Project

To run a project, click ▶ in the **Project** pane or the process flow.

Viewing SAS System Options in Your Current SAS Session

SAS system options can be set in a variety of ways. For example, some system options are set when you start your SAS Enterprise Guide session. To view all of the SAS system options that are defined for your current SAS session, select **Tools** ⇨ **System options viewer**.



System options values

Server: SASApp

Filter results

Option	Value
ERRORHANDLING	
ABORT	
AUTOCORRECT	AUTOCORRECT
BYERR	BYERR
CHKPTCLEAN	NOCHKPTCLEAN
CLEANUP	CLEANUP
DMSSYNCHK	NODMSSYNCHK
DS2SCOND	WARN
DSNFERR	DSNFERR
ERRORABEND	NOERRORABEND
ERRORBYABEND	NOERRORBYABEND
ERRORCHECK	NORMAL
FMTERR	NOFMTERR
LABELCHKPT	NOLABELCHKPT
LABELCHKPTLIB	WORK
LABELRESTART	NOLABELRESTART
QUOTELENMAX	QUOTELENMAX
STEPCHKPT	NOSTEPCHKPT
STEPCHKPTLIB	WORK
STEPRESTART	NOSTEPRESTART
SYNTAXCHECK	NOSYNTAXCHECK
VNFERR	VNFERR
GRAPH	

<no option selected>

Showing 601 option values

From the System Options Viewer, you can learn this information about a selected system option:

- the value of the system option in the current session
- the value of the system option at start-up
- where this system option can be set (either in the OPTIONS statement or at start-up)
- how the current value was set

By default, these system options are organized into groups. Examples of groups are ERRORHANDLING, GRAPH, and PERFORMANCE.

This window can remain open during your entire SAS Enterprise Guide session. You can easily switch between other SAS Enterprise Guide windows and the System Options Viewer to see any changes.

If your SAS environment contains connections to multiple SAS Workspace Servers, use the **Server** drop-down list to see the values for the system options on multiple machines.

Viewing the SAS Catalogs and Formats in Your Current SAS Session

With the Catalog and Formats Explorer, you can explore SAS libraries and the catalogs in those libraries. You can view the metadata (name, description, created and modified dates) for every catalog entry. You can view the contents of some entry types such as SOURCE entries and format entries. You can also delete entries.

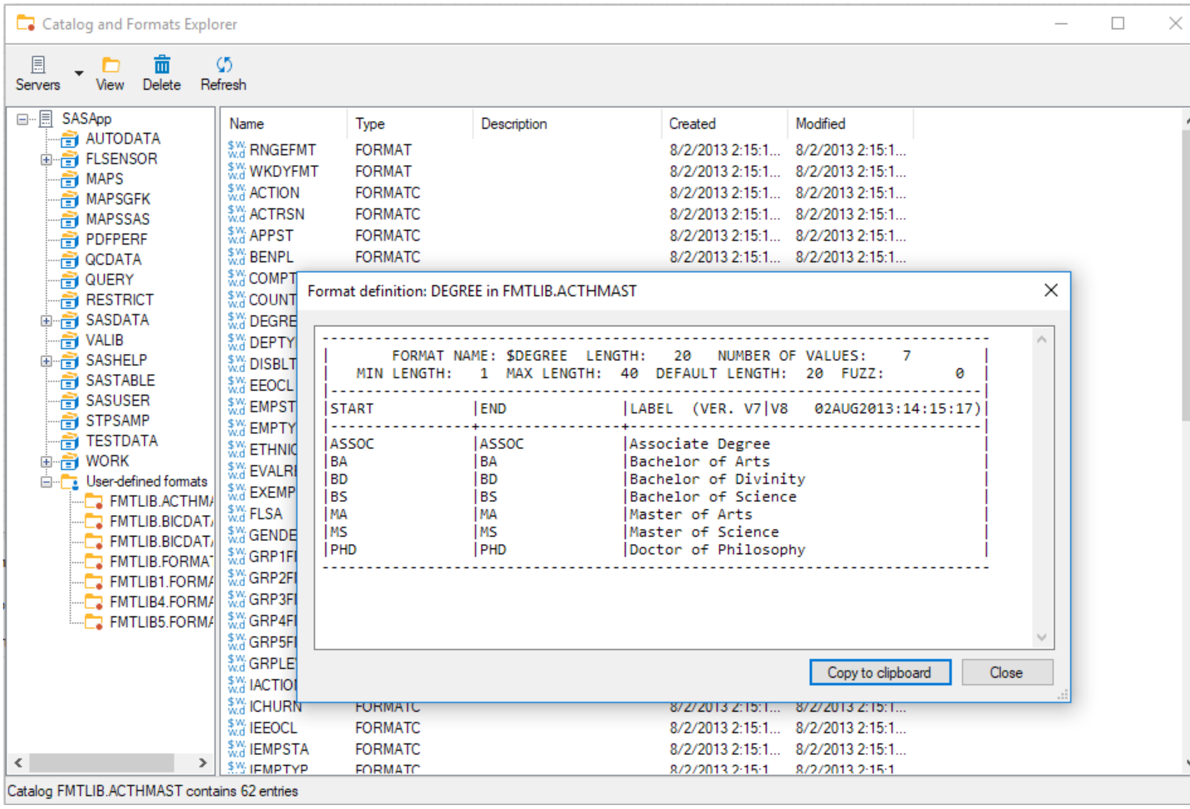
To open the Catalog and Formats Explorer, select **Tools** ⇨ **Catalog and formats explorer**.

Formats can be saved in any catalog. From the Catalog and Formats Explorer, you can view the contents of FORMAT, FORMATC, INFMT, and INFMTC entries. A User-defined formats folder lists the catalogs that contain format definitions within the FMTSEARCH path.

The Catalog and Formats Explorer can remain open during your entire SAS Enterprise Guide session. You can easily switch between other SAS Enterprise Guide windows and the Catalog and Formats Explorer to see any changes. For example, you define a new format in your SAS code. After the code runs, click **Refresh** in the Catalog and Formats Explorer to see your newly defined format.

If your SAS environment contains connections to multiple SAS Workspace Servers, use the **Server** drop-down list to see the SAS catalogs and formats on multiple machines.

This example shows the contents of the DEGREE format definition in the FMTLIB.ACTHMAST catalog.



Name	Type	Description	Created	Modified
RNGEFMT	FORMAT		8/2/2013 2:15:1...	8/2/2013 2:15:1...
WKDYFMT	FORMAT		8/2/2013 2:15:1...	8/2/2013 2:15:1...
ACTION	FORMATC		8/2/2013 2:15:1...	8/2/2013 2:15:1...
ACTRSN	FORMATC		8/2/2013 2:15:1...	8/2/2013 2:15:1...
APPST	FORMATC		8/2/2013 2:15:1...	8/2/2013 2:15:1...
BENPL	FORMATC		8/2/2013 2:15:1...	8/2/2013 2:15:1...
IEEOCL	FORMATC		8/2/2013 2:15:1...	8/2/2013 2:15:1...
IEMPSTA	FORMATC		8/2/2013 2:15:1...	8/2/2013 2:15:1...

Format definition: DEGREE in FMTLIB.ACTHMAST

```
 ----------------------------------------------------------------------------
|       FORMAT NAME: $DEGREE  LENGTH:   20   NUMBER OF VALUES:    7         |
|   MIN LENGTH:   1  MAX LENGTH:  40  DEFAULT LENGTH:  20  FUZZ:        0   |
|---------------------------------------------------------------------------|
|START           |END             |LABEL  (VER. V7|V8   02AUG2013:14:15:17)|
|----------------+----------------+-----------------------------------------|
|ASSOC           |ASSOC           |Associate Degree                         |
|BA              |BA              |Bachelor of Arts                         |
|BD              |BD              |Bachelor of Divinity                     |
|BS              |BS              |Bachelor of Science                      |
|MA              |MA              |Master of Arts                           |
|MS              |MS              |Master of Science                        |
|PHD             |PHD             |Doctor of Philosophy                     |
 ----------------------------------------------------------------------------
```

Catalog FMTLIB.ACTHMAST contains 62 entries

Send Email as a Step in a Project

You can send email as a step in the process flow of a project. The email step is added to the process flow, and it is displayed in the Distribute container in the Project pane. The email step is executed when the project is run.

Note: SAS Enterprise Guide uses SMTP (Simple Mail Transport Protocol) to send email as an unattended step in a process flow. To configure the SMTP server and sender's address, select **Tools** ⇨ **Options**, and then click **Administration** in the selection pane.

To send email as a step in a project:

1 Select an item in the work area.

2 On the toolbar, select **Share** ⇨ **Email as a step in project**.

3 Click **Add** to add files to the **Files To Send** list. You can select files from the current project, your computer, or a SAS server.

Note: You must have SAS installed on your local computer in order to send a SAS data file to an email recipient.

4. If there is an error in creating the file that you are sending, you can use the **Send even when errors occur in the files** option to choose how you want SAS Enterprise Guide to handle the error. If you select **Send even when errors occur in the files**, SAS Enterprise Guide sends the file with the error. If you do not select this option, SAS Enterprise Guide does not send the file that contains the errors.
5. To create a ZIP file that includes all of the companion files to an HTML file, select **Compress all files; include banner image and style for results**. If you do not select this option, only the main HTML file is sent.
6. To rename the compressed file that is attached to the email, select **Rename compressed file to** and enter the name of the file in the box.
7. Click **Next**.
8. Specify the recipients of the email and a subject. Click **Next**.
9. To send the email now, as well as when the project is run, select **Send email immediately**. When you have confirmed the selections that you have made, click **Finish** to send email as a step in the project.

Recovering a Project

About Project Recovery Files

SAS Enterprise Guide can automatically create recovery files for you so that you can recover your project if SAS Enterprise Guide closes unexpectedly due to a situation such as a power loss or unexpected application error. The **Enable auto recovery** option in the Options window enables SAS Enterprise Guide to automatically save the active project to a specified backup location at a specified interval. This option is selected by default. SAS Enterprise Guide also attempts to save a copy of the active project at the time when the application closes unexpectedly.

Typically, after SAS Enterprise Guide closes unexpectedly, there are two types of recovery files for the active project:

- auto-saved copy of the project
- recovered copy of the project

If the project recovery option is enabled, the auto-saved recovery file is automatically saved in the defined save location. At the specified save interval, a new auto-saved recovery file is saved only if SAS Enterprise Guide detects a change in the project since the last time the recovery file was saved. If the project is empty or no changes have occurred since the last time a recovery file was created, no file is created. Each time an auto-saved recovery file is created at the specified interval time, the previous recovery file is overwritten. When you close SAS Enterprise Guide normally, any recovery file that is associated with that session of SAS Enterprise Guide is automatically deleted from the specified save location.

The recovered copy of the project is created at the time when the application closes unexpectedly. This recovery file is created in addition to the auto-saved recovery file and does not overwrite the auto-saved copy. Even if the recovered copy of the project is successfully saved when SAS Enterprise Guide closes, you must carefully review it before you use it to ensure that it is not corrupted.

Recovering a SAS Enterprise Guide Project

You can recover your project if SAS Enterprise Guide closes unexpectedly due to a situation such as a power loss or unexpected application error. When you restart SAS Enterprise Guide after it has closed unexpectedly, a notification appears at the top of the work area. You can choose to recover or delete the recovery files.

Note: Recovered files might be corrupted because they were saved by SAS Enterprise Guide after an unexpected error. Please review the project carefully before you use it. Whenever possible, it is recommended that you use the auto-saved copy of the project.

If the recovered project contained linked files that were saved with the project, you must also save each linked file. If you save the linked files to a location other than their original location, you must update the references to each item in the recovered project.

Performing Project Maintenance

About Performing Project Maintenance

You can use the Project Maintenance window to change references within a single project. You can change global references that might affect many items in a project, such as SAS servers and libraries. You can also change references to individual items such as stored processes or data files.

If you need to change references in multiple projects or update the file format of a project to the current release of SAS Enterprise Guide, use the Migration Wizard for SAS Enterprise Guide and SAS Add-In for Microsoft Office.

To run the wizard, run MigrationWizard.exe, which is located in your SAS Enterprise Guide installation directory.

To change references within a project:

1 Open the SAS Enterprise Guide project that contains the references that you want to change.

2 Select **Tools** ⇨ **Project Maintenance**.

3 You are prompted to close the project before you can perform any maintenance on it. If the project has been changed since the last time you saved it, you are prompted to save the project before it is closed. After the project is closed, it opens in the Project Maintenance window.

4 After you open the Project Maintenance window, you can change these types of items:

   - project
   - servers
   - libraries
   - data items
   - OLAP servers
   - OLAP catalogs

Changing the Project Metadata Profile and Default Server

When you view the project details in the Project Maintenance window, you can see information about when the project was created and where it is stored. You can also view information about the project metadata. You can change the metadata profile and default server that a project uses.

To change the metadata profile and default server:

1 In the Project Maintenance window, click **Project** in the selection pane.

2 Click **Change Active Metadata Profile**. In the Connections window, specify your changes and click **Close**.

3 To save any changes that you have made and to refresh the project information, click **Save Changes and Reload Project**.

Changing a Server Reference

You can use the Project Maintenance window to replace the SAS execution servers throughout a project. When you replace a server, you need to verify that any libraries on that server that are used in the project also exist on the new server. If they do not, you might need to replace some of the libraries in the project. For more information, see "Changing a Library Reference" on page 48.

To change a server reference:

1 In the Project Maintenance window, click the server that you want to change under the **Servers** heading in the selection pane.

2 Click **Replace In Project**.

Note: Servers to which you have active connections are indicated with a green check mark.

3 Select the server that you want to use and click **OK**. The Project Maintenance window is updated with information about the new server.

Changing a Library Reference

You can use the Project Maintenance window to replace a library throughout a project. You can select a different library on the same server, or you can choose a library on another server. When you replace a library, you need to verify that any data items in that library that are used in the project also exist in the new library. If they do not, you might need to replace some of the data items in the project. For more information, see "Changing a Data Item Reference" on page 49.

To change a library reference:

1 In the Project Maintenance window, click the library that you want to change under the **Libraries** heading in the selection pane.

2 Click **Replace In Project**.

Note: Servers to which you have active connections are indicated with a green check mark.

3 Select the library that you want to use and click **OK**. The Project Maintenance window is updated with information about the new library.

Changing a Data Item Reference

You can use the Project Maintenance window to replace a data item in a project. You can select a different data item in the same library, or you can choose a data item in another library. When you replace a data item, you can verify that the data item exists in the new location that you selected.

To change a data item reference:

1. In the Project Maintenance window, click the data item that you want to change under the **Data Items** heading in the selection pane.
2. Click **Change Location**.
3. Select the data item that you want to use. You can select a data item from the same library, or you can browse other libraries and servers or files on your local computer.
4. Click **OK**. The Project Maintenance window is updated with information about the new data item.
5. To verify that the data item actually exists, click **Verify Data Exists**. The verification status is updated to indicate whether the data item exists or is not found.

Changing an OLAP Server Reference

You can use the Project Maintenance window to replace the OLAP servers throughout a project. When you replace an OLAP server, you need to verify that any OLAP catalogs on that OLAP server that are used in the project also exist on the new OLAP server. If they do not, you might need to replace some of the catalogs in the project. For more information, see "Changing an OLAP Catalog Reference" on page 50.

To change an OLAP server reference:

1. In the Project Maintenance window, click the server that you want to change under the **OLAP Servers** heading in the selection pane.
2. Click **Replace In Project**.

   Note: Servers to which you have active connections are indicated with a green check mark.

3. Select the OLAP Server that you want to use and click **OK**. The Project Maintenance window is updated with information about the new OLAP server.

Changing an OLAP Catalog Reference

You can use the Project Maintenance window to replace an OLAP catalog throughout a project. You can select a different catalog on the same OLAP server, or you can choose a catalog on another OLAP server. When you replace a catalog, you need to verify that any cubes in that catalog that are used in the project also exist in the new catalog. If they do not, you might need to replace some of the cubes in the project.

To change an OLAP catalog reference:

1. In the Project Maintenance window, click the catalog that you want to change under the **OLAP Catalogs** heading in the selection pane.
2. Click **Replace In Project**.

   Note: Servers to which you have active connections are indicated with a green check mark.

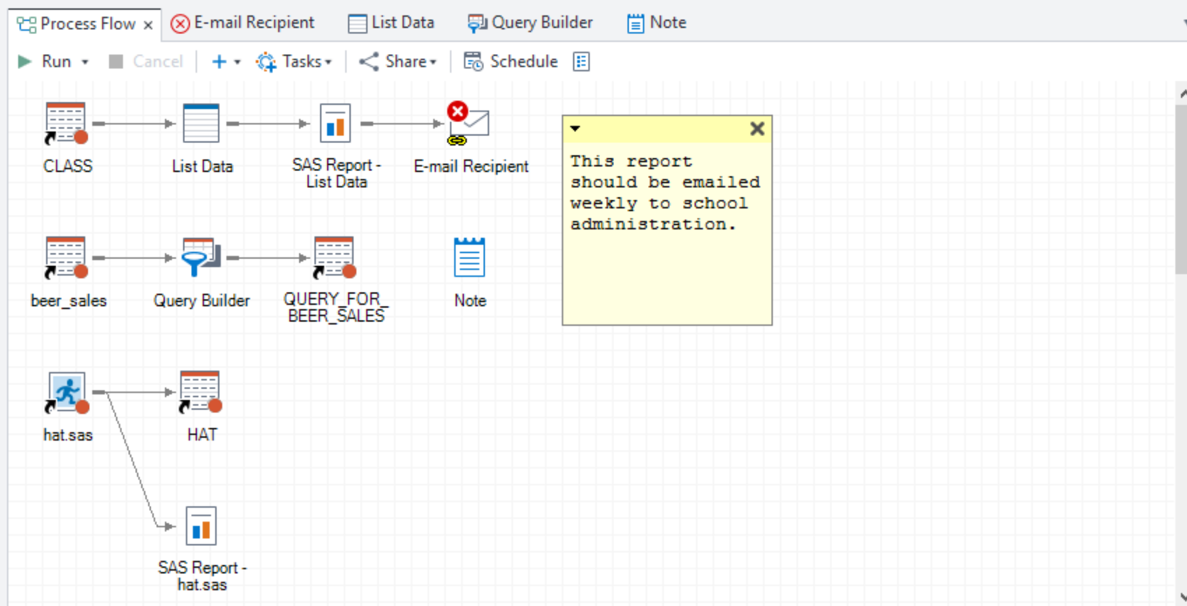
3. Select the OLAP catalog that you want to use and click **OK**. The Project Maintenance window is updated with information about the new catalog.

Working with Process Flows

About Process Flows

A process flow consists of one or more objects and could contain a process tree. An object is represented by an icon in the process flow. Examples of objects include a data set, a task, and your results. A process tree shows the relationship between two or more objects. SAS Enterprise Guide creates a default process flow when you start a project. For a single project, you can have multiple process flows.

To view the process flow, click **Process Flow** in the Project pane or the **Process Flow** tab in the work area.



This report should be emailed weekly to school administration.

Note: By default, the generated code objects and the SAS output log objects do not appear in the process flow. For information about including objects for the generated code, the SAS output log, and the generated results in the process flow, see "Setting Project and Process Flow Options" on page 484.

Differences between the Process Flow and the Project Pane

The process flow has functionality similar to that of the Project pane. However, these features are unique to this area of the work area:

- You can customize the process flow by turning grid lines on and off, by zooming, or by changing the background color.
- You can create multiple process flows and copy and paste or move objects among them.
- You can run a portion of the process flow or the entire process flow.

Creating a Process Flow

A process flow is automatically created when you start a project. Multiple process flows can be associated with a project. To view a process flow, select the name of the process flow from the Project pane. If you use automation to run these process flows, the process flows run in the order in which they appear in the Project pane.

To create a process flow, select **File** ⇨ **New** ⇨ **Process flow**. By default, the process flow is named Process Flow `n`, where `n` is the number of the process flow.

Renaming an Object

You can rename some of the objects in a process flow. When you rename an object, you are changing the label of that object. The first time you save an object, the label is used as the default file name. When you rename an object in the process flow, you automatically rename the corresponding item in the Project pane.

Note: You cannot rename the code, log, or results objects. If you are displaying the full pathname of a data object, then you cannot rename the data object.

If you have already saved the object, renaming it in the process flow does not affect the file name that is used when you saved it. To save the file with a different file name, you must select **File** ⇨ **Save *item-name* As**.

Arranging Objects within a Process Flow

By default, SAS Enterprise Guide automatically arranges the objects in a process flow. If you want to change the arrangement of the objects in a process flow, you must turn off the auto-arrange functionality. You can turn the auto-arrange functionality on and off by right-clicking in the process flow and selecting **Auto arrange**. The setting of the **Auto arrange** option applies only to the currently selected process flow. When you save a project, the setting of the **Auto arrange** option for each process flow is also saved.

When the auto-arrange functionality is off, you can manually move objects wherever you want them in the process flow. These objects remain in that specified location until you move them again or until you turn the auto-arrange functionality back on. If you turn the auto-arrange functionality back on, the objects in the process flow are moved to the auto-arrange position.

To turn the auto-arrange functionality on and off, right-click in a process flow and select **Auto arrange**.

Copying and Moving Objects in Process Flows

You can copy and paste one or more objects within a process flow and from one process flow to another. You can also copy an entire process flow and paste it into the same project or paste it into another project. When you copy and paste an entire process flow or a branch of a process flow, the links between the objects are maintained.

Note: You cannot paste a data object into a process flow that already contains a reference to the same data object.

When you paste a task or query object into a process flow, SAS Enterprise Guide automatically creates a shortcut to the associated data object if the process flow does not contain a reference to that data set. If a reference to the data object does exist, the copied task or query object uses the existing reference.

You can also move task, code, and query objects from one process flow to another. You can move only one object at a time. When you move an object, SAS Enterprise Guide moves the associated data object if the process flow does not already contain the data object. If the process flow already contains a moved data object, that data object is associated with the moved task or query object. When you move an object, any existing results that are associated with the object are also moved.

Note: If you move an object that has a condition applied, the condition is removed when you copy and paste or move the object.

To copy and paste an object within a process flow:

1. Right-click the object in the process flow that you want to copy, and select **Copy**.
2. Right-click anywhere in the process flow and select **Paste**.

To copy and paste an object from one process flow to another:

1. Right-click the object in the process flow that you want to copy and select **Copy**.
2. Open the process flow where you want to paste the object.
3. Right-click anywhere in the process flow and select **Paste**.

To copy an entire process flow:

1. In the Project pane, right-click the process flow that you want to copy and select **Copy**.
2. Open the project in which you want to paste the process flow. Right-click anywhere in the Project pane and select **Paste**.

To move an object from one process flow to another:

1. Right-click the object that you want to move and select **Move *object-name* to process flow**. Another menu appears that lists all of the process flows for the current project.
2. Select the name of the process flow where you want to move the object.

Linking Objects in a Process Flow

You can link the task, code, data, query, and note objects in a process flow to create a branch in the current process flow. Objects that you link are connected by a dotted line. Links that are automatically generated by SAS Enterprise Guide are connected by a solid line.

To link objects in a process flow:

1. Right-click the object that you want to link to another object and select **Link to**.
2. Select the name of the object that you want to link to.
3. Click **OK**.

You can also link objects by completing these steps:

1. In the process flow, select the object that you want to link. When you select an object, SAS Enterprise Guide surrounds the object with a dotted-line box.
2. Move the mouse pointer to the border of the dotted-line box until the mouse pointer changes to a plus sign. Click the left mouse button, drag the arrow that appears to the object that you want to link to, and then release the mouse button.

If a link is allowed between the two selected objects, the two objects are connected by a dotted line in the process flow. If a link is not allowed, the dotted line does not appear.

Note: If you link a data object to a code object, then you can insert a reference to the SYSLAST macro variable directly into SAS code in place of a data set name. In your SAS code, you can replace any reference to a SAS data set (<*libref.member*>) with &SYSLAST. The following example shows how you could add a reference to the SYSLAST macro variable in the code for the PRINT procedure:

```
proc print data=&SYSLAST; run;
```

By adding references to the SYSLAST macro variable in your code, you can run the code against different data sources within the project simply by linking a different data object to the code object. SAS Enterprise Guide automatically prepends the assigned macro variable to the submitted code. You can view the code that SAS Enterprise Guide generates if you have selected the **Show generated code items** option in the **Project and Process Flows** pane of the Options window.

Annotating a Process Flow

You can use notes to add information to a process flow or to specific objects in the process flow. When you create a note, you can link it to one or more objects in the process flow. You can display the notes so that they can be read in the process flow, or you can collapse them so that they appear as another object in the process flow.

To manage a note in the process flow, click ▼ in the upper left corner of the expanded note or right-click the collapsed note.

Working with Conditional Processing

About Conditional Processing

Conditional processing (also referred to as conditional logic) enables you to control the flow of a process by adding a condition to an element in SAS Enterprise Guide. You can create conditions for tasks, queries, and code nodes. You can also create conditions to specify when to export or email files. Each element can have only one condition. A condition can be based on a value in an input data set, on the value of a prompt, or on a time or date. You can create complex conditions by using IF-THEN/ELSE statements. You can also create ELSE IF conditions.

In the process flow, the nodes that have conditions are flagged (⚑). Each condition has a unique number. Nodes that are part of the same condition have the same number, so you can identify which nodes are part of a single condition. Only nodes that are siblings can be involved in the same condition. These siblings can be in the same branch of a process flow, or they can be siblings that are created by a user-defined link. When you run the process flow, conditions that ran are identified by ✓ to indicate that the condition has run, and conditions that did not run are identified by ⊗ to indicate that the condition did not run.

Note: When creating a condition, all the data for the condition must be stored on a single server. You cannot select data from multiple servers for a condition.

Add a Condition

1 Right-click the node in the process flow and select **Condition** ⇨ **Add**.

2 Click **Add** to specify the definition of the condition.

a From the **Based on** drop-down list, select the value for the condition. You can base the value on an input data set, on the value of a prompt, on a time or date, or on the value of a macro variable.

b Specify the definition options for the condition.

Note: When you are defining a condition for a prompt or macro variable, review this information:

- If you are basing your condition on a prompt, only single-line text, numeric, date, and time prompts can be used in conditional processing.
- If you are basing your condition on a macro variable, you can specify a global macro variable, macro variables that are defined by the macro processor, or a user-defined macro variable. You must specify the name of the macro variable as it should appear in the code, so precede all names with &. If the macro variable is a string, the name must be enclosed in double quotation marks. If the value for the macro variable is a character value, enclose this value in double quotation marks.

Click **OK**.

3 (Optional) Specify how to combine conditions. When you create multiple conditions, these conditions are automatically combined by using the AND operator. To change this to an OR operator, select the first definition and click **Combine**.

4 From the **Then run this task** drop-down list, select the task to run when this condition is true.

5 (Optional) Click **Add Else If** to create an ELSE IF condition. Repeat steps 2–4 to create a condition on the **Else if** tab.

6 (Optional) From the **Else run this task** drop-down list, select the task to run if the conditions on the **If** and **Else if** tabs are false.

7 Click **OK** to add the condition to the selected node. In the process flow, the condition is identified by a flag and numbered.

Note: Not every node in the process flow is available from the **Then run this task** and **Else run this task** drop-down lists. These drop-down lists contain only the sibling nodes that are currently not part of the condition.

Edit a Condition

1 Right-click the node in the process flow and select **Condition** ⇨ **Edit**.

2 In the **If this condition is true** box, select the definition that you want to edit and click **Edit** to specify the definition of the condition.

3. (Optional) Specify how to combine conditions. When you create multiple conditions, these conditions are automatically combined by using the AND operator. To change this to an OR operator, select the first definition and click **Combine**.

4. From the **Then run this task** drop-down list, select the task to run when this condition is true.

5. (Optional) Click **Add Else If** to create an ELSE IF condition. Repeat steps 2–4 to create a condition on the **Else if** tab.

6. (Optional) From the **Else run this task** drop-down list, select the task to run if the conditions on the **If** and **Else if** tabs are false.

7. Click **OK** to add the condition to the selected node. In the process flow, the condition is identified by a flag and numbered.

Note: Not every node in the process flow is available from the **Then run this task** and **Else run this task** drop-down lists. These drop-down lists contain only the sibling nodes that are currently not part of the condition.

Running a Process Flow

Understanding How a Process Flow Runs

You can use a process flow to run an entire project or to run only specific tasks. The objects in the process flow are run in the following order of precedence:

- from top to bottom. Objects that are closer to the top of the process flow are run first.
- from left to right. If the objects are in the same horizontal position in the process flow, objects on the left are run before objects on the right (or the reverse, for a right-to-left display).

Objects in a process tree are run based on their hierarchical dependencies, regardless of the object's position in the process flow. Independent objects, such as a code object or a custom task, are run based on their position in the process flow. If these objects rely on the results from a process tree in the same process flow, then the independent object might fail depending on its position in the process flow.

When you run a process flow (or a part of a process flow), the Submission Status pane displays the order in which the objects are processed. In the process flow, all of the objects that are waiting to be processed are highlighted in yellow, and the object that is currently being processed is highlighted in green.

If a user runs a process flow that contains two or more objects with prompts, then the prompts are displayed to the user in the order in which the objects are run in the process flow. For example, assume that you have four prompts listed in the following order: A, B, C, and D. If Object1 is run first and uses prompts A and C, and Object2 uses prompts B and D, then the prompts are displayed in the following order: A, C, B, D.

If two or more objects share a prompt, then the shared prompt is displayed with the prompts of the first object that uses it. For example, assume that Object1 uses prompts A, C, and D, and Object2 uses prompts B and D. The prompts are displayed to the user in the following order: A, C, D, B.

Run the Entire Process Flow

Complete either of these steps:

- Select **Run** on the toolbar.
- Right-click in the process flow and select **Run**.

Running the entire process flow replaces the results for all of the tasks that it contains.

Note: If the process flow contains tasks from different servers, each server starts running a task at the same time.

Run Part of the Process Flow

Complete either of these steps:

- Select the object where you want to start, and then select **Run** ⇨ **Run from selected item**.
- Select the last object that you want to run, and then select **Run** ⇨ **Run to selected item**.

Run a Process Flow in Batch Mode

Write a VBscript program that runs the process flow and saves the output to a specified location, or use the sample VBScript program.

This VBScript program prompts you to enter the location and name of your saved SAS Enterprise Guide project, runs the tasks that are specified in the process flow, and saves the results to `C:\My Documents\output.egp`.

Note: In the `Set pfd` command, you must replace *My PFD* with the name of the process flow in your project.

```
'----
    ' Get project file name
    '----
    Dim sFile
    sFile = InputBox("Enter the full path of your SAS Enterprise Guide project:")

    '----
    ' Start up SAS Enterprise Guide using the project name
    '----
    Dim app
    Dim prj

    Set app = WScript.CreateObject("SASEGObjectModel.Application.8.1")
    Set prj = app.Open(sFile)

    '-----
    ' Run a particular item such as a pfd
    '-----
    Dim pfd
    Set pfd = prj.ProjectItems.Item("My PFD")

    pfd.Run

    '-----
    ' Save the new project
    '-----
    prj.SaveAs ("c:\my documents\output.egp")

    '-----
    ' Close the project
    '-----
    prj.Close
    app.Quit

    '-----
    ' Display a message when complete
    '-----
   MsgBox (sFile & " Complete")
```

Running a Process Flow When You Open a Project

You can create a process flow named "autoexec" that can be run automatically whenever you open your project. For example, you could use this process flow to define libraries or to assign macro variables for the project. The autoexec process flow can contain any of the objects that can be contained in a process flow, including SAS programs, tasks, stored processes, and custom tasks. For more information, see "Creating a Process Flow" on page 51.

A project can contain only one process flow named autoexec, and, by default, you are prompted to run the autoexec process flow when you open the project. You can also specify an option to have the autoexec process flow run automatically every time you open the project. For more information about the autoexec option, see "Setting Project and Process Flow Options" on page 484.

Note: For information about other autoexec functionality in SAS Enterprise Guide, see "Using Autoexec Processing in SAS Enterprise Guide" on page 26.

Creating an Ordered List of Tasks

You can create a list of tasks and programs in the current project that can be run in the order that you choose. You can select objects from more than one process flow. In this way, you can group objects without changing their positions in process flows or their relationships to other objects.

To create an ordered list of tasks and programs:

1. Select **File** ⇨ **New** ⇨ **Ordered list**.
2. In the Ordered List window, click **Add** to display the tasks and programs that are in the current project. Select one or more items and click **Open** to add them to the ordered list.
3. To change the position of an item in the list order, select the item in the Ordered List window and click **Up** or **Down**.
4. When the list is complete, click **Run** to run the tasks and programs in the list, or click **Save** to save the list.

Automating Projects

Running a Project in Batch Mode

You can run a SAS Enterprise Guide project in batch mode outside the SAS Enterprise Guide application by saving the project and using a script program to run the project.

1. Set up the SAS Enterprise Guide project. Create a project in SAS Enterprise Guide and add the data and perform the tasks that you want to run on the data.

Be sure to run the tasks so that they are saved into your project. Save the project.

2 Write a script program that runs your project and saves the output to a specified location, or use our sample VBScript program.

3 Run the script by double-clicking the file name in Windows Explorer. To view your results, open the project in SAS Enterprise Guide.

Here are some troubleshooting tips:

The project does not run.
    A DCOM server cannot be used to run a project in batch mode.

Windows Explorer does not know how to open your script program.
    Ensure that you have installed the scripting host.

The program runs very quickly without creating the output project file.
    Ensure that the output directory exists.

You get a script execution error when you run the program.
    Ensure that the project is not already open in SAS Enterprise Guide and that you have entered the project name correctly.

This VBScript program prompts you to enter the location and name of your saved SAS Enterprise Guide project and saves the results in the **C:\Program Files\SASHome\SASEnterpriseGuide\*release-number*\Sample** directory. This program runs all of the tasks in your project. If you want to run selected tasks from your project, use the sample program that uses a process flow.

```
' **************************************************************************
' This sample uses the SAS Enterprise Guide automation object model to
' open an example project, run the entire project, and save the project file
' as a new file. The script also saves a copy of all data and task results in
' the folder you specify in that SaveAs statement. The sample also creates
' a .txt file that lists the data sets, tasks, and results objects in the
' project.
'
' The goal of this example is to illustrate how to run and save objects
' in a project by using automation.
'
' To run the automation script, copy this example to a text file and save
' it as Sample_RunSave.vbs in C:\Program Files\SASHome\SASEnterpriseGuide\release-number\Sample
' If you prefer, you can change the project path statements to point to a
' different folder.
'
' Create a SAS Enterprise Guide project called Sample_RunSave.egp that has
' at least one task that creates at least one result and save it in the same
' path as the script. The name of the project file and the path to that file
' must match the project name and the path that is specified in the script.
' **************************************************************************

Dim app         'Represents the SAS Enterprise Guide Application object
Call dowork    'Call the subroutine to perform the work and run the task

'shut down the app
If not (app is Nothing) Then
    app.Quit
    Set app = Nothing
```

```
End If

'************************************************************************
' The subroutine dowork is where the task is run and the output is produced.
' The output is saved to a file that is represented in the variable "outfile". One
' output file is saved for each results format that is produced by the task.
'***********************************************************************

Sub dowork()
    On Error Resume Next
    '----
    'Start Enterprise Guide using the project name

Dim prjName     'Represents the name of the project being run
Dim prjObject    'The actual project object that is running
Dim dc          'Represents the DataCollection in the project
Dim d           'index for DataCollection
Dim tc          'Represents the TaskCollection under each data object
Dim t           'index for TaskCollection
Dim rc          'Represents ResultsCollection under each Task
Dim r           'index for ResultsCollection
Dim outfile    'Used to define a path for the Contents file
Dim fileSys    'Represents the FileSystemObject that lets you write files
Dim Contents    'Represents the File you are going to write
Dim ext        'Used to store extension of results


outfile = "C:\Program Files\SASHome\SASEnterpriseGuide\8.1\Sample\SampleRunSave_ProjectContents

    Set fileSys = CreateObject("Scripting.FileSystemObject")
    Set Contents = fileSys.OpenTextFile(outfile, 2, True)

    'Name the project to run
    prjName = "C:\Program Files\SASHome\SASEnterpriseGuide\8.1\Sample\Sample_RunSave.egp"    'P

    Set app = CreateObject("SASEGObjectModel.Application.8.1")
    If Checkerror("CreateObject") = True Then
        Exit Sub
    End If

    'Open the Enterprise Guide application and the project
    '-----
    ' open the project
    '-----
    Set prjObject = app.Open(prjName,"")
    If Checkerror("app.Open") = True Then
        Exit Sub
    End If


    '-----
    ' run the project
    '-----
    prjObject.run
    If Checkerror("Project.run") = True Then
        Exit Sub
    End If

    '-----
    ' Save all the results from all of the tasks under all of the data sets
    '-----
    d=0
    Set dc = prjObject.DataCollection
    Do While d < dc.Count
    Contents.WriteLine("DataSet Name: " & dc(d).Name)
      t=0
      Set tc = dc(d).Tasks
      Do While t < tc.Count
        Contents.WriteLine("     TaskName:  " & tc(c).Name)
        r=0
        Set rc = tc(t).Results
        Do While r < rc.Count
           ext = ".txt"          'default extension for output file
           Contents.WriteLine("        Results Name:  " & rc(r).Name)
           if Left(rc(r).Name,3) = "HTM" then ext = ".html"
           if Left(rc(r).Name,3) = "RTF" then ext = ".rtf"
           if Left(rc(r).Name,3) = "PDF" then ext = ".pdf"
           rc(r).SaveAs("C:\Program Files\SASHome\SASEnterpriseGuide\release-number\Sample\Samp
           r = r + 1
        Loop
      t = t + 1
      Loop
   d = d + 1
   Loop


    '-----
    ' Save the new project
    '-----
    prjObject.Save
    If Checkerror("Project.Save") = True Then
        Exit Sub
    End If

    '-----
    ' Close the project
    '-----
    prjObject.Close
    If Checkerror("Project.Close") = True Then
        Exit Sub
    End If

End Sub

'************************************************
' This function is an error handler.
'************************************************
Function Checkerror(fnName)
    Checkerror = False

    Dim strmsg
    Dim errNum
    If Err.Number <> 0 Then
        strmsg = "Error #" & Hex(Err.Number) & vbCrLf & "In Function " & fnName & vbCrLf & Err.
        'MsgBox strmsg  'Uncomment this line if you want to be notified via MessageBox of Error
        Checkerror = True
    End If

End Function
```

Scheduling Projects and Process Flow

In SAS Enterprise Guide, you can use the Microsoft Windows Task Scheduler to schedule projects and process flows to run at a specified time or as the result of a system event. By default, when you open the Task Scheduler, a script is automatically created in SAS Enterprise Guide to run the project or process flow. When you schedule the project or process flow, the Task Scheduler creates a scheduled task that includes the script and the criteria that specify when the task should be run. The scheduled task is added to the Project pane.

Note: You must save the project to your local computer before you can create a scheduled task.

To schedule a project or process flow:

1 To schedule a project, select ⋮ and select **Schedule project**.

  To schedule a process flow, open a process flow and perform either of these tasks:

  - Right-click the name of the process flow in the Project pane and select **Schedule**.
  - From the process flow toolbar, click **Schedule**.

  An interface to the Microsoft Windows Task Scheduler opens.

2 On the **General** tab, you can change the default name of the scheduled task by entering a new name in the **Name** box. The task name must be unique. You can also enter or change any of the other properties on the **General** tab. By default, the scheduled task runs using the account of the user who scheduled the task.

3 On the **Triggers** tab, you can specify the triggers, or criteria, that are used to start the scheduled task. Triggers can be time based or event based. For example, time-based triggers can start the task at a specific time of day or on a regular, repeated schedule. Event-based triggers start the task as a result of a system event, such as when a user logs on or when the computer enters an idle state. By default, the task is scheduled to run every day at the current time. If a task has more than one trigger, it starts when any of the triggers occur.

4 On the **Actions** tab, you can specify one or more actions that are performed when the task is run. By default, the scheduled task runs the project or process flow. The default action is **Start a program**, and the default program or script is

the generated script file. To run a different script, select the script and click **Edit**. To add another action, click **New**.

5 On the **Conditions** tab, you can specify conditions that control whether the task is run. The conditions are evaluated after a task is triggered to start.

6 On the **Settings** tab, you can specify options for how the task is run, stopped, or deleted.

7 On the **Run Times** tab, you can view a list of the times that the task is scheduled to run. On the **History** tab, you can view a list of the times that the task has run.

8 To save the scheduled task, click **OK**. The scheduled task and the associated script are added to the Project pane.

To run a scheduled project or process flow, close the project so that the results of the scheduled task can be saved. If you scheduled a task for a process flow, close the project that contains the process flow.

To edit a scheduled task, right-click the scheduled task in the Project pane and select **Open** . The scheduled task opens in the Task Scheduler.

5

Working with Data

Accessing Data in SAS 9

In SAS 9, SAS data sets are stored in SAS libraries. A SAS library is a collection of one or more SAS data sets that are recognized by SAS. For example, in a Microsoft Windows environment, a SAS library is typically a group of SAS data sets in the same folder or directory.

To access a SAS library, you assign it a name. The name is known as the libref, or library reference, and it refers to the physical location of the files. The libref is the nickname that you use to identify libraries in a SAS program.

SAS automatically assigns two libraries for you each time you start SAS.

- Sashelp – a permanent library that contains sample data that you can use. This is a read-only library. You cannot save content to this library.
- Work – a temporary library for files that do not need to be saved. Files in this library are deleted when you close SAS.

All SAS data sets have a two-level name. The first name is the libref, which tells SAS which library contains the data set. The second name is the name of the file that contains the data. The two names are separated by a period.

For example, sashelp.class refers to the file named Class that is stored in the Sashelp library. If the file is stored in the temporary Work library, then you do not have to specify the two-level name. You can use only the name of the file, and SAS assumes that it is stored in the Work library.

If you want to save data in a library so that you can use it again the next time you open SAS Studio, you must create your own library. For more information, see "Creating a SAS Library" on page 70.

Accessing Data in SAS Viya

If your site has licensed and installed SAS Viya products, you can work with SAS Cloud Analytic Services (CAS) tables. SAS Cloud Analytic Services performs analysis on in-memory tables only. You must first load your data into in-memory CAS tables and then run analyses against the CAS tables.

All access to data with SAS Cloud Analytic Services is through a caslib. A caslib is an in-memory space to hold tables, access control lists, and store data source information. All data is available to CAS through caslibs, and all operations in CAS that use data are performed with a caslib in place. Authorized users can add or manage caslibs with the CASLIB statement.

CAS tables are referenced by their caslib and the table name. After you create a caslib, you can load your data into CAS. You can load data from your client machine, from a server, or from a database table.

For more information, see the CASLIB Statement in *SAS Cloud Analytic Services: User's Guide* .

Note: To access CAS tables with a SAS 9.4 procedure or the DATA step, you use the LIBNAME statement to assign a CAS engine libref to a caslib.

Creating a SAS Library

To access SAS 9 data, you create a SAS library by using the LIBNAME statement. The LIBNAME statement associates the name of the library, or libref, with the physical location of the library.

There are a few guidelines to keep in mind when you create the name of your library:

- limit your library name to eight characters.
- start with a letter or underscore.
- use only letters, numbers, or underscores. Blanks are not allowed.

You need to submit a LIBNAME statement only once during a SAS session. The libref remains in effect for the duration of the session. A LIBNAME statement consists of these elements:

- the keyword LIBNAME
- the libref name that you want to use
- the pathname that represents the physical location of the library
- a semicolon

The following LIBNAME statement assigns the mydata libref to a folder on your local computer. The directory that you associate with the libref must already exist before you can assign it to the libref.

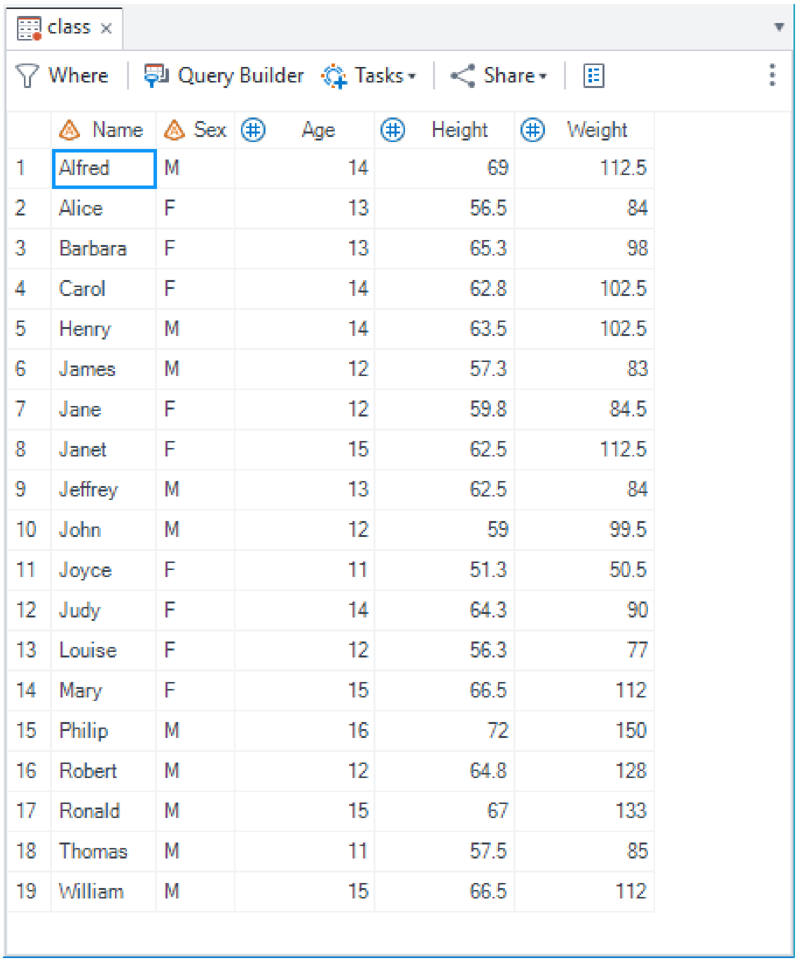
```
libname mydata 'c:\projects\data\';
```

You can submit the LIBNAME statement alone, or you can add it to the top of your program so that it is run every time you run the program. If you want to assign the libref every time you open your project, you can create a program that contains the LIBNAME statement and then add it to the ″autoexec″ process flow. For more information, see “Running a Process Flow When You Open a Project” on page 59.

Note: For more information about working with data in SAS Viya, see “Accessing Data in SAS Viya” on page 69.

About the Data Grid

When you open an existing table or create a table in SAS Enterprise Guide, you use the data grid.



	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84
10	John	M	12	59	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90
13	Louise	F	12	56.3	77
14	Mary	F	15	66.5	112
15	Philip	M	16	72	150
16	Robert	M	12	64.8	128
17	Ronald	M	15	67	133
18	Thomas	M	11	57.5	85
19	William	M	15	66.5	112

To view properties for columns, right-click the column heading and select **Properties** from the pop-up menu. To set these properties for an existing table, you must first change the protection mode to Update mode. For more information, see "Changing the Protection Mode" on page 115.

Creating Data

About Creating Data

You can create a data table from within SAS Enterprise Guide. If you have an open project, the new table is added automatically to the current project.

You create a data table by using the New Data wizard to specify the name and location and to create the columns.

Step 1: Specify the Name and Location of the New Data

1 Select **File** ⇨ **New** ⇨ **Data**.

2 Specify the name of the new file.

3 Select the location in which you want to store the data.

4 Click **Next** to create columns and specify their properties.

Step 2: Create Columns and Specify Their Properties

You can create the initial columns for the new data and specify properties for each column. By default, six character columns are created for you.

To create columns and specify properties:

1 To edit a default column, select the column from the Columns list. The properties for that column are displayed in the **Column Properties** box.

2 Specify the properties for the column. The value of **Group** depends on what you select for **Type**. If the data type is Numeric, the value of **Group** can be Numeric, Date, Time, or Currency. If the data type is Character, the value of **Group** can only be Character.

Note: For information about naming conventions for column names, see "Creating Variable Names in SAS Enterprise Guide" on page 73.

3 To create a column, click **New**. A new character column is added to the list of columns. You can follow steps 2 and 3 to edit the column, if necessary.

4 To copy an existing column, select the column in the **Columns** list that you want to copy and click **Duplicate**. A copy of the column is added to the list of columns.

5 To paste the properties from one or more existing columns from another data set or a Microsoft Excel spreadsheet, click **Paste**.

Note: The **Paste** button is available only if you have copied the columns from the data set or Microsoft Excel spreadsheet to the clipboard before opening the New Data wizard.

From the Confirm Paste window, you can select the columns from an existing data set or a Microsoft Excel spreadsheet that you want to paste into a SAS Enterprise Guide data grid. Here are some things to note:

- SAS Enterprise Guide pastes only the columns properties, such as the name and column type, into the data grid. It does not paste the data values for the column.

  If you are copying and pasting columns from a Microsoft Excel spreadsheet, SAS Enterprise Guide uses the data values in the copied column to determine the column type (either character or numeric). Depending on your data, this assignment might not be correct. For example, if you have a column that contains phone numbers, SAS Enterprise Guide sets the column type to numeric. However, because you do not want to perform any calculations on the data in this column, you should set the column type to character. You can edit this column property.

- If the copied columns do not have names, then the columns are named Column*n*, where *n* is an integer value. The first column is named Column1, the second column is named Column2, and so on.

- If a column with the same name already exists, a warning appears. To overwrite the existing column, click **OK**.

6 Click **Finish** to create the data. The new data table opens in the SAS Enterprise Guide work area.

Creating Variable Names in SAS Enterprise Guide

The VALIDVARNAME= system option controls the type of SAS variable names that can be created and processed. When you create data in SAS Enterprise Guide, by default, SAS Enterprise Guide uses VALIDVARNAME=ANY, which means that variable names must follow these rules:

- The name can begin with or contain any characters, including blanks, national characters, special characters, and multi-byte characters.
- The name can be up to 32 bytes in length.
- The name cannot contain any null bytes.
- Leading blanks are preserved, but trailing blanks are ignored.
- The name must contain at least one character. A name with all blanks is not permitted.
- The name can contain mixed-case letters. SAS stores and writes the variable name in the same case that is used in the first reference to the variable. However, when SAS processes a variable name, SAS internally converts it to uppercase. Therefore, you cannot use the same variable name with a different combination of uppercase and lowercase letters to represent different variables. For example, cat, Cat, and CAT all represent the same variable.

Using SAS Enterprise Guide Data in Base SAS

Reading and writing data that you create in SAS Enterprise Guide might fail in Base SAS because the default value for VALIDVARNAME= is V7 in Base SAS. There are more restrictions for variable names that are created with VALIDVARNAME=V7. Variable names that are created with VALIDVARNAME=V7 must follow these rules:

- The name can be up to 32 characters.
- The first character must begin with a letter of the Latin alphabet (A–Z, a–z) or the underscore. Subsequent characters can be letters of the Latin alphabet, numerals, or underscores.
- Trailing blanks are ignored. The variable name alignment is left-justified.
- A variable name cannot contain blanks or special characters except for the underscore.
- A variable name can contain mixed-case letters. SAS stores and writes the variable name in the same case that is used in the first reference to the variable. However, when SAS processes a variable name, SAS internally converts it to uppercase. Therefore, you cannot use the same variable name with a different combination of uppercase and lowercase letters to represent different variables. For example, cat, Cat, and CAT all represent the same variable.
- Do not assign variables the names of special SAS automatic variables (such as _N_ and _ERROR_) or variable list names (such as _NUMERIC_, _CHARACTER_, and _ALL_) to variables.

If data that you create in SAS Enterprise Guide might be used in Base SAS, you can complete either of these steps:

- follow the variable name guidelines for VALIDVARNAME=V7
- submit `OPTIONS VALIDVARNAME=ANY` in Base SAS before you use the data

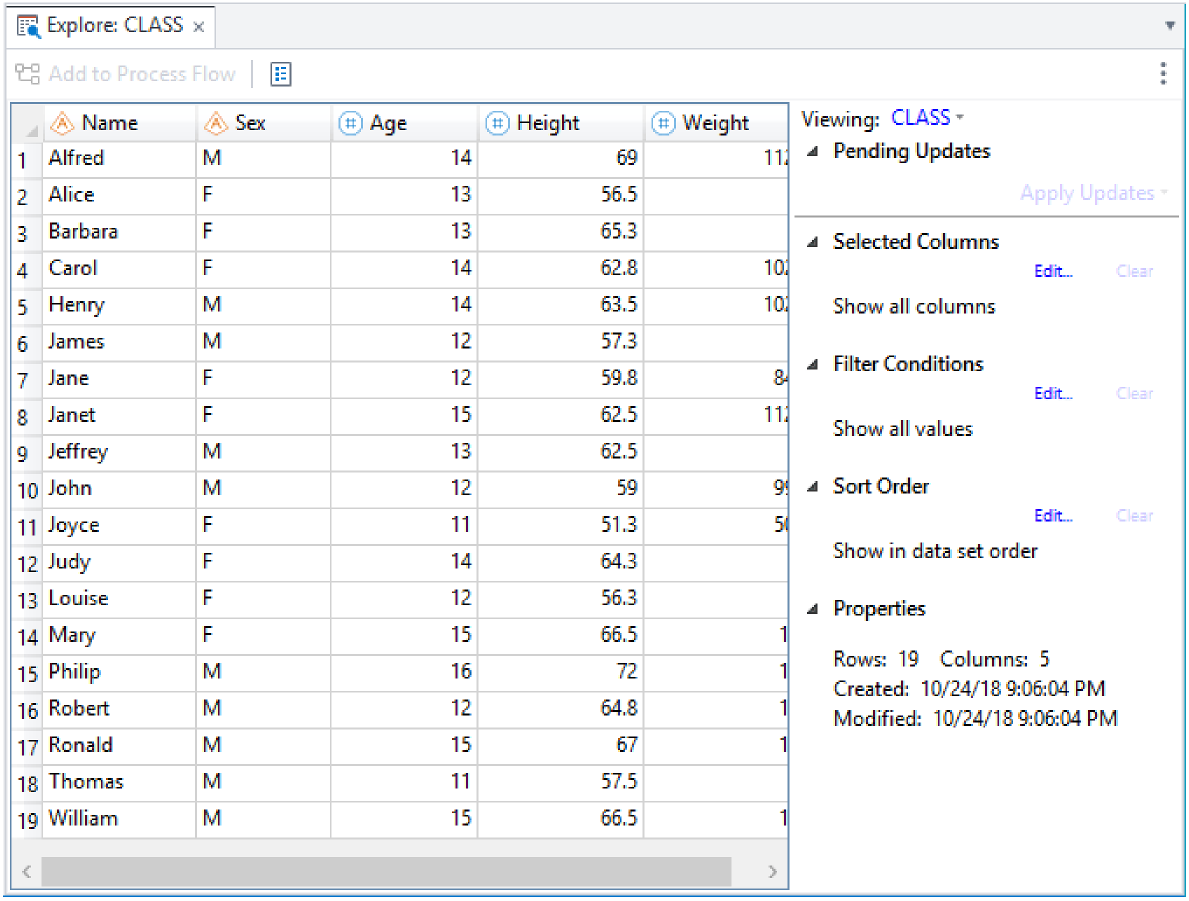
Exploring Your Data

About the Data Explorer

You can use the Data Explorer to create different views of a SAS data set by specifying columns to include, creating filters, and specifying sorting criteria. After you have explored the data, you can then choose to add it to a project.

Note: You can explore only data in a SAS folder or data that resides on a SAS server and is not in the Work library. You cannot explore data that resides on your local machine. If you are accessing data through a user-assigned library, you can explore only data in a user-assigned library that uses the BASE LIBNAME engine. Exploring data in a user-assigned library for a DBMS system is not supported.

When you open a data set in the Data Explorer, the data is displayed in a grid view with the Data Exploration panel on the right. You can use the Data Exploration panel to create a specific view of the data.



	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	11
2	Alice	F	13	56.5	
3	Barbara	F	13	65.3	
4	Carol	F	14	62.8	10
5	Henry	M	14	63.5	10
6	James	M	12	57.3	
7	Jane	F	12	59.8	8
8	Janet	F	15	62.5	11
9	Jeffrey	M	13	62.5	
10	John	M	12	59	9
11	Joyce	F	11	51.3	5
12	Judy	F	14	64.3	
13	Louise	F	12	56.3	
14	Mary	F	15	66.5	1
15	Philip	M	16	72	1
16	Robert	M	12	64.8	1
17	Ronald	M	15	67	1
18	Thomas	M	11	57.5	
19	William	M	15	66.5	1

Viewing: CLASS

Pending Updates

Selected Columns

Show all columns

Filter Conditions

Show all values

Sort Order

Show in data set order

Properties

Rows: 19 Columns: 5
Created: 10/24/18 9:06:04 PM
Modified: 10/24/18 9:06:04 PM

Opening a Data Set in the Data Explorer

You can use the Data Explorer to open a SAS data set that resides on a SAS server and is not in the Work library or any data sets in a SAS folder. You cannot open a data set that resides on your local machine in the Data Explorer.

To open a data set in the Data Explorer

- In the Servers or SAS Folders tab of the navigation pane, right-click the data set that you want to explore and select **Explore**. The data set opens in a grid view with the Data Exploration panel on the right.

Creating a Custom View of Your Data

Specifying Columns to Include in the View

You can use the Data Explorer to create a view of your data that includes only the columns that you specify.

1 In the Selected Columns area of the Data Exploration panel, click **Edit**. The Selected Columns window opens.

2 Select one or more columns from the Available columns list and click ⇨.

Note: To change how the columns are displayed in the Available columns list, right-click the **Name** heading and select the option that you want to use.

3 To eliminate any duplicate rows and keep only the distinct rows, select **Select distinct rows only**.

4 To change the order in which the columns appear in the data view, select the column in the Selected Columns list that you want to move and click ⇧ or ⇩.

5 To save your column selection, click **Add to Pending Updates**. Your column selection is added to the Pending Updates list in the Data Exploration panel. Your selections are not applied to the data view until you apply the updates. For more information, see "Applying Updates to the Data" on page 80.

Note: To clear a column selection after you have applied it, click **Clear** in the Selected Columns area. The **Show all columns** selection is added to the **Pending Updates** list.

Creating a Filter by Specifying Column Values

1 In the Filter Conditions area of the Data Exploration panel, click **Edit**. The Filter Conditions window opens.

Note: If you want to create a filter on one column at a time, click the appropriate column heading and select the filter criteria from the filter window. If you want to change the relationship between filter conditions or group conditions together, you must use the Data Exploration pane.

2 Click the column box to select the column on which you want to filter the data.

3. Click the condition box to define the filter condition.

4. To select the specific values that you want to include in the data, click the **Choose Values** tab and select one or more values.

   Note: By default, the frequency count for each column is displayed with the column value. Use the Showing option to select another statistic to display. If your data set is very large, you might want to display only the distinct values to improve performance. Select from among the following options:

   - **Distinct values**
   - **Frequencies for All Values**
   - **Statistics for All Values**
   - **Frequencies for Filtered Values**
   - **Statistics for Filtered Values**

5. Click **Apply selection** to save the condition.

6. If you want to create additional filter conditions, click ✸, and repeat the preceding steps. If you are creating more than one filter condition, you can change the relationship between filter conditions and group filter conditions together to ensure that the specified conditions are evaluated as a group when the filter is applied. You can also change the order in which filter conditions are applied.

   - You can choose between two filter relationships. Click the filter relationship in the upper left corner of the window to change the relationship.
     - **ALL** – specifies that only rows that meet all of the conditions are included. This is the default value.
     - **ANY** – specifies that rows that meet any one or more of the conditions are included.
   - To group filter conditions, select the conditions that you want to be evaluated as a group and click ⇥. To ungroup the conditions, select the conditions that you want to remove from the group and click ⇤.

   Note: To select a specific condition in the Filter Conditions window, select the filter by clicking the area to the left of the column box or to the right of the condition box.

7. To view the SQL code that is automatically generated for the filter, click **Show query statement**. The code changes dynamically as you add and remove filter conditions.

   Note: The SQL code that is displayed is read only. You cannot modify the code from within the Data Explorer. However, you can add the data exploration view to your project, and modify the associated SQL code in the Query Builder.

8 To save the filter, click **Add to Pending Updates**. Your filter is added to the **Pending Updates** list in the Data Exploration panel. Your filter is not applied to the data view until you apply the updates. For more information, see “Applying Updates to the Data” on page 80.

> **Note:** To clear a filter after you have applied it, click **Clear** in the Filter Conditions area. The **Show all values** selection is added to the **Pending Updates** list.

Creating a Filter by Using a Conditional Expression

To specify the column values:

1 In the Filter Conditions area of the Data Exploration panel, click **Edit**. The Filter Conditions window opens.

> **Note:** If you want to create a filter on one column at a time, click the appropriate column heading and select the filter criteria from the filter window. If you want to change the relationship between filter conditions or group conditions together, you must use the Data Exploration pane.

2 Click the column box to select the column on which you want to filter the data.

3 Click the condition box to define the filter condition.

4 To specify a filter condition, click the **Conditional filter** tab. You can choose to create a conditional filter on a single value, a range of values, or a list of values.

5 Click **Apply selection** to save the condition.

6 If you want to create additional filter conditions, click ✸, and repeat the preceding steps. If you are creating more than one filter condition, you can change the relationship between filter conditions and group filter conditions together to ensure that the specified conditions are evaluated as a group when the filter is applied. You can also change the order in which filter conditions are applied.

- You can choose between two filter relationships. Click the filter relationship in the upper left corner of the window to change the relationship.
  - **ALL** – specifies that only rows that meet all of the conditions are included. This is the default value.
  - **ANY** – specifies that rows that meet any one or more of the conditions are included.
- To group filter conditions, select the conditions that you want to be evaluated as a group and click ⇥. To ungroup the conditions, select the conditions that you want to remove from the group and click ⇤.

- To change the order in which filter conditions are applied, select one or more conditions that you want to move and click ⇧ or ⇩.

Note: To select a specific condition in the Filter Conditions window, select the filter by clicking the area to the left of the column box or to the right of the condition box.

7. To view the SQL code that is automatically generated for the filter, click **Show query statement**. The code changes dynamically as you add and remove filter conditions.

Note: The SQL code that is displayed is read only. You cannot modify the code from within the Data Explorer. However, you can add the data exploration view to your project, and modify the associated SQL code in the Query Builder. For more information, see "Adding a Data Exploration View to Your Project" on page 81.

8. To save the filter, click **Add to Pending Updates**. Your filter is added to the **Pending Updates** list in the Data Exploration panel. Your filter is not applied to the data view until you apply the updates. For more information, see "Applying Updates to the Data" on page 80.

Note: To clear a filter after you have applied it, click **Clear** in the **Filter Conditions** area. The **Show all values** selection is added to the **Pending Updates** list.

Sorting the Data

1. In the Sort Order area of the Data Exploration panel, click **Edit**. The Sort Order window appears.

Note: If you want to sort the data by one column at a time, click the appropriate column heading and select the order from the sort order drop-down list. The default value is to **Sort by Data Order**. If you want to sort by multiple columns, you must use the Data Exploration panel.

2. Select one or more columns from the **Available columns** list and click ⇨.

3. To change the order of the columns on which the data is sorted, select the column in the Selected Columns list that you want to move and click ⇧ or ⇩.

4. To save your sorting selection, click **Add to Pending Updates**. Your sorting selection is added to the Pending Updates list in the Data Exploration panel. Your filter is not applied to the data view until you apply the updates. For more information, see "Applying Updates to the Data" on page 80.

Note: To clear a filter after you have applied it, click **Clear** in the **Sort Order** area. The **Show in data set order** selection is added to the Pending Updates list.

Applying Updates to the Data

When you are exploring your data, you can select which columns to display, specify filter conditions, and specify a sorting order. Any selections that you make are saved as pending updates until you apply your updates to your data. You can apply the updates to the current view of the data, or you can apply the updates to a new view of the data.

To apply updates:

1 In the Pending Updates area of the Data Exploration panel, review the list of updates.

2 To apply the updates to the current view of the data, click **Apply Updates**. To apply updates to a new view of the data, click ▼ beside the **Apply Updates** options and select **New View**.

Note: You can also apply updates by pressing F6.

Using the Manage Views Dialog Box

You can use the Manage Views window to rename, move, and delete data exploration views that you have created.

Note: When you select a view in the Manage Views window, the criteria that were used to create the view are displayed at the bottom of the window.

In the Data Exploration panel, click ▼ beside the **Viewing** option and select **Manage Views**.

Note: The **Manage Views** option is enabled only if you have created more than one view of a data set.

Managing Data Exploration

Adding a Data Exploration View to Your Project

After you explore your data, you can add it to your project. When you add the data to the project, the original data set is added to the project along with the query that is used to create the current view of the data. If you want to change the default options for adding the data to the project, you can add the data with options.

Note: When you add the data to your project, you can use the Query Builder to modify the query. However, you cannot import any changes that you make back into the Data Explorer.

To add a data exploration view to your project, on the Data Exploration toolbar, click **Add to Process Flow**. The original data, a Query Builder node, and the current view of the data are added to the project. This option is available only after you have applied updates to the data.

Note: You must have an open project in order to add a data exploration view to your project. If you do not have an open project, you are prompted to create one.

Viewing the Status of Jobs in the Data Explorer

When you are exploring your data, query jobs are created in order to generate the column selections, filters, and sorting updates that you specify. You can view the status of all the jobs that the Data Explorer is running. By default, completed jobs are automatically removed from the list of jobs every 10 seconds. You can change how often the status log is cleared by using the data exploration options. For more information, see “Setting Data Exploration Options” on page 82.

When you view the job status, you can also display the SAS log file for a specific job. For example, if the Data Explorer is unable to apply an update, you can use the SAS log for that job to help determine what the problem is.

To view the status

On the Data Exploration toolbar, click ⋮ and select **Status**. The Status window opens with a list of jobs for the Data Explorer.

Setting Data Exploration Options

You can set options for the Data Explorer by using the Data Exploration Options window. To open this window, click ⋮ and select **Options**. The Data Exploration Options window opens.

Table 5.1 Data Exploration Options

Option Name	Description
Automatically show the conditional filter tab in the column drop down	Automatically displays the **Conditional filter** tab as the active tab when you open the Filter Conditions window. By default, the **Choose values** tab is active. For more information, see "Creating a Filter by Using a Conditional Expression" on page 78.
Automatically get column values	Specifies whether to automatically retrieve values for the columns in the data set when you are creating a filter. ■ **For a limited number of columns** – Specifies the number of columns for which you want to retrieve values when you are creating a filter. The default value is 20, which means that values are retrieved for the first 20 columns in the data set. ■ **When you open the column drop-down list** – Retrieves values for all columns only when you open the column drop-down list to begin creating your filter. ■ **Limit automatic generation to data sets with rows less than** – Specifies the maximum size of a data set for which to automatically retrieve column values. The default value is 10,000 rows. ■ **Automatically generate** – Determines whether to generate a frequency count or statistics for the filtered values or all values when you are creating a filter.
Create data views when pending changes are submitted	Automatically creates a data view when you apply the pending updates to the data.
Automatically clear status log on job completion every *n* seconds	Specifies how frequently to clear the status log. The default value is 10 seconds. For more information, see "Viewing the Status of Jobs in the Data Explorer" on page 81.

Adding Data to a Data Grid

About Adding Data

You can type your data into each cell in the grid by using the arrow keys to navigate around the grid. You can also copy and paste or drag and drop columns from other grids. To perform many of these functions, you might need to open the data in Edit mode. For more information, see “Changing the Protection Mode” on page 115.

Each data type has a default format and informat. For more information, see Table 5.3 on page 86.

Note: For information about naming conventions for variable names, see “Creating Variable Names in SAS Enterprise Guide” on page 73.

Copying and Pasting Data Columns

1 Click the heading of the column that you want to copy. This selects the entire column.

2 Select **Edit** ⇨ **Copy**. You can also right-click the heading of the column that you want to copy and select **Copy** from the menu that appears.

Note: If you want to copy the column headers as well as the values, right-click the column heading and select **Copy with headers**.

3 In the data grid that you want to paste the column into, select the cell in which you want the column to start.

4 Select **Edit** ⇨ **Paste**. You can also right-click the cell and select **Paste** from the menu that appears.

Note: When you copy and paste data into the data grid, any formats that you assigned to the data are lost.

Note: Changes that you make in the data grid are saved immediately. You cannot use the **Edit** ⇨ **Undo** option for these changes.

Searching the Data Grid

You can search the data grid for a specific value or column name.

Note: You cannot search in the Data Explorer.

To search for text in the data grid

1. Select **Edit** ⇨ **Find**.
2. In the **Find what** box, type the text that you are searching for.
3. You can specify the following options when you search:
   - **Match case** to match the capitalization in the text that you are searching for.
   - **Match whole cell** to require that the text that you are searching for exactly matches the text within a cell. Cells that contain the text as part of a longer word or value are not included in the results.

     Note: The Match whole cell option does not require that the capitalization in the text that you are searching for matches the capitalization in the text within a cell.

   - **Search up** to specify the direction in which you want to search the data. By default, this option is not selected, and the direction is set to down.
4. You can specify the area that you want to search. From the drop-down list, select from the following options:
   - **Whole table** searches the entire data grid but excludes the column headings from the search.

     Note: If you select the **Whole table** option, the search is limited to the entire data grid, excluding the column headings. But if the entire data grid is selected, the **Whole table** option does include the column headings. To select the entire data grid, click the top, left-most cell.

   - **Column only** searches the active or currently selected column. In cases where there is more than one column selected, the column heading or a cell within the column of the active or currently selected column has a bold black outline around it. To select a column, click on the column name.
   - **Selection only** limits the search to the currently selected portion of the grid.

     Note: If the column headings are included in the selection, then they are included in the search. The selected areas must be contiguous. If you have two non-contiguous selected areas, the search is done on the most recently selected area. If you want to select several columns to include in a selection, you must press the SHIFT key while you select the columns. If you press the CTRL key while you select the columns, then the search is limited to the last column that you selected.

- **Column headings only** searches only the column headings. The rest of the data grid is not included in the search.

5 To find the next occurrence of the text, click **Find Next**.

Opening and Viewing Data

If you want to look at the contents of a data table, select **File** ⇨ **Open** and browse to find the file. If you have a project with data, double-click the data in the Project pane (identified in the Project pane by ▦). The file contents appear in a data grid. You can use the **Open Items** pane to view a list of all the open data sets and other items in your current SAS Enterprise Guide session. For more information, see “Viewing and Closing SAS Data Sets” on page 86.

Note: You can open very large data sets in SAS Enterprise Guide because only a small portion of the data is read and displayed in the data grid at a time. To view the total number of rows in the data, click on the vertical scroll bar to the right of the data grid. To view the total number of columns in the data, click on the horizontal scroll bar at the bottom of the data grid.

There is a symbol next to each variable name in the grid that indicates the data type of the variable:

Table 5.2 Variable Symbols

Symbol	Data Type
▲	Character
#	Numeric
📅	Date
🕒	Time
💵	Currency

Here are the default formats and informats for the data types:

Table 5.3 Default Formats and Informats

Data type	Format	Informat
Character	$w.	$w.
Numeric	BESTw.d	w.d
Date	MMDDYYw.d	MMDDYYw.d
Time	TIMEw.d	TIMEw.d
Currency	DOLLARw.d	none

For a list of all the SAS formats and informats, see *SAS Formats and Informats: Reference*.

Viewing and Closing SAS Data Sets

During a SAS Enterprise Guide session, you might open and view the contents of several data tables. From the Open Items tab of the navigation pane, you can view a list of all the SAS data sets and other items that are open in the current session and select one or more items to close. Closing a SAS data set removes the cached data from the current SAS Enterprise Guide session. However, closing a data set does not affect any process flows, SAS tasks, or SAS reports that use these data sets.

To close a data set:

1. Click 🗎 in the navigation pane.
2. Select one or more items to close and click ✕ in the Open Items toolbar.

Subsetting Your Data with a WHERE Expression

Creating a SAS WHERE Expression

If you are viewing a SAS data set that resides on a server, you can subset the data set by specifying a SAS WHERE expression. A WHERE expression enables you to specify a condition for selecting rows of data. A WHERE expression can be as simple as a single variable or constant. A WHERE expression can also include a sequence of operators and operands and use a SAS function.

To specify a SAS WHERE expression:

1 Open the SAS data set that you want to subset and click **Where** on the work area toolbar. The WHERE expression text box opens at the top of the data grid.

Note: If your data provider does not support WHERE subsetting, the **Where** button is disabled.

2 In the WHERE expression text box, enter the SAS WHERE expression using the following syntax: `operand <operator> <operand>`

operand
something to be operated on. An operand can be a variable, a SAS function, or a constant.

operator
a symbol that requests a comparison, logical operation, or arithmetic calculation. All SAS expression operators are valid for a WHERE expression. These expression operators include arithmetic, comparison, logical, minimum and maximum, concatenation, parentheses to control order of evaluation, and prefix operators. In addition, you can use special WHERE expression operators. These expression operators include BETWEEN-AND, CONTAINS, IS NULL or IS MISSING, LIKE, sounds-like, and SAME-AND.

Note: You can join multiple WHERE expressions together by using a logical operator such as AND, AND NOT, OR, or OR NOT: `where-expression-1 <logical-operator where-expression-n>`

3 Press Enter to apply the expression to the data set.

Note: For more information, see *SAS Global Statements: Reference*.

Creating a SAS Local Data Provider WHERE Expression

If you are viewing a SAS data set that resides on your local computer and you do not have SAS installed locally, you can subset the data set by specifying a WHERE expression for the SAS local data provider. A WHERE expression enables you to specify a condition for selecting rows of data.

To specify a SAS Local Data Provider WHERE expression:

1 Open the SAS data set that you want to subset and click **Where** on the work area toolbar. The WHERE expression text box opens at the top of the data grid.

Note: If your data provider does not support WHERE subsetting, the **Where** button is disabled.

2 In the WHERE expression text box, enter the WHERE expression using the following syntax: `column-name operator value`

column-name
must be a valid column name from the source table. The column name cannot contain a square bracket ([ ]). If the name contains spaces, you must enclose the name in square brackets as follows: [FIRST NAME].

operator
must be one of the following: <, >, <=, >=, <>, =, LIKE.

value
a constant, which is a number or a quoted character string that indicates a fixed value, such as 3.1415 or 'chocolate'. The following syntax rules apply to values:

- For character values, use single quotation marks: 'chocolate'
- For single quotation marks that are embedded within a character value, use two single quotation marks: 'O''Malley'
- For numeric values, you can use decimal points and scientific notation or the special null value for missing values: AGE <> NULL
- If the operator is LIKE, the value can include wildcards. Only the asterisk (*) and percent sign (%) are allowed, and they must occur at the beginning or end of the value: '*ship' or 'ba%'.

Note: You can join multiple WHERE expressions by using the logical operators AND and OR: `where-expression-1 <logical-operator where-expression-n>`

These syntax rules apply to logical operators:

- There is no precedence between AND and OR.
- Clauses can be grouped within parentheses.

- You cannot group clauses that are joined by an OR, and then join the group to another clause with an AND.

  For example, the following expression is invalid: `(LastName = 'Smith' OR LastName = 'Jones') AND FirstName = 'John'`

  You must rewrite the expression as follows: `(LastName = 'Smith' AND FirstName = 'John') OR (LastName = 'Jones' AND FirstName = 'John')`

3 Press Enter to apply the expression to the data set.

File Types That SAS Enterprise Guide Reads and Writes

In addition to reading and creating SAS data sets, SAS Enterprise Guide can also read and export several other types of data files. The Import Data wizard enables you to create SAS data sets from text, HTML, or PC-based database files (including Microsoft Excel, Microsoft Access, and other popular formats). You can also import JMP, Stata, and SPSS files. For more information, see "About the Import Data Wizard" on page 106.

Note: In the 64-bit version of SAS Enterprise Guide, you can import Microsoft Access files (*.mdb, *.accdb) only if the 64-bit version of Microsoft Access or the 64-bit version of the Microsoft Office System (ACE) Driver is installed. Similarly, if you are running the 32-bit version of SAS Enterprise Guide, then you must have the 32-bit version of Microsoft Access or the Microsoft Office System (ACE) Driver in order to import Microsoft Access files.

Note: To import Stata or SPSS files, SAS Enterprise Guide uses the SAS/ACCESS interface to PC Files and the IMPORT procedure, so you must have SAS/ACCESS to PC files and Base SAS to import these data files. The IMPORT procedure is supported only on SAS servers that are running on Windows or UNIX.

You can also use SAS Enterprise Guide to connect to Open Database Connectivity (ODBC) and Object Linking and Embedding Database (OLE DB) data sources. For more information, see "Inserting an ODBC Data Source" on page 102 and "Opening a Data Source by Using OLE DB" on page 104. In addition, you can access many other file types by using SAS/ACCESS engines.

Table 5.4 Understanding File Types in SAS Enterprise Guide

Files Types That SAS Enterprise Guide Can Read	File Types That SAS Enterprise Guide Can Write
Microsoft Access 1.x, 2.0, Access 95, Access 97, Access 2000, Access 2002 (XP), Access 2003, Access 2007 and later	
Microsoft Excel 4-7, Excel 97, Excel 2000, Excel 2002 (XP), Excel 2003, Excel 2007 and later	Microsoft Excel (*.xlsx)
HTML	
Text (delimited or fixed column)	Text (*.txt, *.csv, *.tab)
All versions of Stata (*.DTA) under Microsoft Windows	
All versions of SPSS (*.SAV) under Microsoft Windows	
JMP	

Changing File Name Properties

When you create a project, you can run it first against sample data before you run it against your real data. When you are ready to use real data, you can change the file that the data item points to by changing the file name.

Note: To view the properties of the data, open the data in the work area and click ▤ in the toolbar. You can also right-click the data in the Project pane or the process flow and select **Properties**.

To change the file name:

1 On the **General** page of the Properties window, click **Change**. You are prompted to choose the location from which you want to open the data. After you select the location, the Insert Data window opens.

2 Select the file that you want the data item to point to, and click **Open**.

3 Click **OK** in the Properties window to close the window and save your changes.

Note: Changing the file name property does not automatically update the results. You must rerun the tasks in order to update your results.

Viewing Column Properties

You can view the properties of each column in a data grid. You can update the properties only if the data is in Edit mode. For more information, see “Changing the Protection Mode” on page 115. To view the properties of a column, right-click the column in the data grid and select **Properties**.

Note: If you open the data in Read-Only mode, then only a summary page of General properties is displayed. You cannot update any of the properties for a column in a read-only data grid.

Table 5.5 *General Properties*

Name	Specifies the name of the new column. Valid column names are 1 to 32 characters long.
Label	Specifies the label that identifies the column.
Type	Specifies the data type in the column. The value can be Character or Numeric. **Note:** If you change the type from numeric to character, a warning message tells you that the change might affect the integrity of your data. You should change the type only if you are certain that it will not adversely affect your data. If the column that you selected is protected by an integrity constraint, then this option is read-only.
Length	Specifies the length of the column value in bytes. **Note:** If the column that you selected is protected by an integrity constraint, then this option is read-only.
Group	Specifies the data group. The value depends on what you select for **Type**. If the data type is Numeric, then the value of **Group** can be Numeric, Date, Time, or Currency. If the data type is Character, then the value of **Group** can only be Character.

Table 5.6 *Formats Properties*

Categories	Specifies the category of the column format. You can use a standard SAS format or user-written format that you previously defined in the

	Create Format task. For more information, see “About the Create Format Task” in *SAS Tasks in SAS Enterprise Guide and SAS Add-In for Microsoft Office*. For a list of all the SAS formats and informats, see *SAS Formats and Informats: Reference*.
Formats	Displays the name of the column format. Each data type has a default format. For more information, see Table 5.3 on page 86.
Attributes	■ **Overall width** – number of characters a user is allowed to enter for a column value. ■ **Decimal places** – number of decimal places that are allowed in the column. **Note:** When you select attributes for the column, an example of how the column will appear in the data grid is shown in the **Example** box.

Table 5.7 Informats Properties

Categories	Specifies the category of the column informat. You can use a standard SAS informat or user-written informat. For a list of all the SAS formats and informats, see *SAS Formats and Informats: Reference*.
Informats	Specifies the name of the column informat. Each data type has a default informat. For more information, see Table 5.3 on page 86.
Attributes	■ **Overall width** – number of characters a user is allowed to enter for a column value. ■ **Decimal places** – number of decimal places that are allowed in the column. **Note:** When you select attributes for the column, an example of how the column will appear in the data grid is shown in the **Example** box.

Table 5.8 Confirm Options Properties

Prompt for confirmation	Enables you to preview format changes that you have made to the column. When this option is selected and you save a change to the column format on the Formats page of the Properties window, the Confirm Results window opens with a preview of

	the data in the new format. To apply the new format to all of the data in the column, click **Commit changes**. **Note:** This option is not selected by default. You must select this option each time you open the Properties window.
Skip missing values	Omits any missing values from the data preview.
Start with row	Specifies the row number of the first row that you want to display in the data preview.
Number of rows to show	Specifies the number of rows that you want to include in the data preview. **Note:** The preview is limited to the first 30 rows of the data.

Creating Database Tables

You can view the properties of each column in a data grid. You can update the properties only if the data is in Edit mode. To view the properties of a column, right-click the column in the data grid and select **Properties**.

You can create Oracle and DB2 tables by saving a new or existing data source in SAS Enterprise Guide to an Oracle or DB2 library.

Note: You cannot create a database table from a file if the file contains a column name that is a reserved word in the database.

To save a new database table:

1 Right-click the file that you want to save from the Project pane or from the process flow, and select **Share** ⇨ **Export** . The Save window opens.

2 Select the server, library, and database in which you want to create the table.

   Note: Because there is no default server for the database table, you must specify a server. To select a different server from the one that is currently displayed, click **Servers** in the navigation pane on the left.

3 If you want to give the database table a different name, in the **File name** box, type the name of the table.

4 Click **Save** to save the new database table.

Updating Database Tables

You can update some types of database tables, such as Oracle and DB2, from within SAS Enterprise Guide by typing changes directly into the table. You can also delete rows from the tables.

These database tables cannot be updated from within SAS Enterprise Guide:

- Teradata
- Hadoop
- Netezza
- Impala
- Aster
- Greenplum

To update a database table:

1. Open the database table in a SAS Enterprise Guide data grid. Ensure that the table is opened in Edit mode.
2. Type your changes into the database table. Your changes are automatically saved in the unprotected data table.

To delete rows from a database table:

1. Open the database table in a SAS Enterprise Guide data grid. Ensure that the table is opened in Edit mode.
2. Select the row or rows that you want to delete.
3. Right-click anywhere in the row and select **Delete**. You are prompted to confirm that you want to delete the selected row or rows.

Exporting Data Files

By default, data files are exported as SAS Data Files (V7 Long Names) (*.sas7bdat). If you want to export your data as another file type, you must change the file type in the **Save as type** drop-down list.

Note: When you export a data file to a different file type, the data values might not be formatted exactly the same.

To export a data file:

1 Right-click the file that you want to export and select **Share** ⇨ **Export** . You are prompted to choose the location where you want to export the data. After you select the location, the Export window opens.

Note: You can also export a data file by clicking **Share** ⇨ **Export** on the toolbar of the data tab.

2 Select the folder in which you want to save the file and enter the name of the exported file.

3 To change the file type, select a new file type from the **Files of type** drop-down list.

4 Click **Save**.

Note: You can also export data as a step in the process flow of a project.

Exporting Data or Code Files as a Step in a Project

About Exporting Files as a Step in a Project

You can export code, data files, and reports as a step in the process flow of a project. The export step is added to the process flow and the Project pane. The export step is executed when the project is run.

Note: When you export a data file to a different file type, the data values might not be formatted exactly the same.

Step 1: Select the File to Export

1 In the Project pane or process flow, right-click the file that you want to export and select **Share** ⇨ **Export as a step in project**. The Export wizard opens.

Note: If there is an error in creating the file that you are exporting, you can use the **Export even when errors occur in the file** option to choose how you want SAS Enterprise Guide to handle the error. If you select **Export even when errors occur in the file**, SAS Enterprise Guide attempts to export an earlier version of the file with the same name, if one exists. If you do not select this option, SAS Enterprise Guide displays an error during the export step and continues running the rest of the project.

2 From the **File To Export** list, select the file that you want to export.

3 If you have selected a data file, results in SAS Report format, or a log file, click **Next** to select the type of output file.

Otherwise, click **Next** to specify the name and the location of the output file.

Step 2: Select the Output File Type

1 Before you can select the output file type, you must select the file to be exported.

2 Select the file type from the **Output File Type** list.

3 If you have selected a file type other than SAS, click **Next** to modify additional output file options

If you have selected a SAS file type, click **Next** to specify the name and the location of the output file.

Step 3: Modify the Output File Options

Depending on the output file type, you might be able to modify additional output file options.

To modify output file options:

1 Before you can select the output file options, you must select the output file type.

2 To use column labels instead of the column names in the exported file, select **Use labels for column names**.

3 Click **Next** to specify the name and the location of the output file.

Step 4: Specify the Name and Location of the Output File

To specify the name and location of the output file:

1. Select **Local Computer** or **SAS Servers** as the location for the output file. Click **Browse** to change the file name and location.

2. If you have selected a code file to export, select the type of character encoding you want to use from the **Encoding** drop-down list.

3. To overwrite an existing file with the same name, select **Overwrite existing output**. If you do not select this option and a file with the same name already exists, a new file name is created by appending the current date and time to the existing file name.

4. If you have selected a code file to export, click **Next** to preview the code to export.

   Otherwise, click **Next** to create the export file.

Step 5: Preview the Code to Export

To preview the code to export:

1. In the Code available for export list, clear the check boxes next to any code items that you do not want to export.

2. In the Code preview area, you can click **Preview All Selected** to preview all of the selected code items, or you can click **Preview of *code-item*** to view only the selected code item.

3. Select from these options:

   - **Include headers** to include headings that provide descriptive information about the code.

   - **Include generated wrapper code** to include wrapper code that SAS generates.

   - **Include custom SAS code from the Tools --> Options dialog** to include SAS code that is specified in the Options window.

   - **Include library assignment code** to include the code that SAS generates to assign the library.

4. Click **Next** to create the export file.

Step 6: Create the Export File

To create the export file:

1. View the selections that you have made.
2. When you have confirmed your export selections, click **Finish** to export the file as a step in the process flow of the project.

Sending Data

You can send a copy of your SAS Enterprise Guide data to another user through electronic mail. To send data through e-mail, you need to have installed an e-mail application such as Microsoft Outlook that supports the Messaging Application Programming Interface (MAPI).

You can also send a copy of the active data source to Microsoft Excel, Microsoft Word, or JMP.

- To send data to another user, right-click the data, and then select **Share** ⇨ **Email**.
- To send data to Microsoft Excel, right-click the data, and then **Share** ⇨ **Send to Microsoft Excel**. The active data source is inserted into a new Microsoft Excel worksheet.

  Note: If you have applied a user-defined format to a numeric column, then that format is retained when the data is sent to Excel. Numeric columns with user-defined formats cannot be used in numeric calculations after the data is sent to Excel. This applies only to user-defined formats.

- To send data to Microsoft Word, right-click the data, and then select **Share** ⇨ **Send to Microsoft Word**. The active data source is inserted as a table into a new Microsoft Word document.
- To send data to JMP, right-click the data, and then select **Share** ⇨ **Send to JMP**. The active data source is displayed in JMP.

Working with Information Maps

About Information Maps

Information maps are user-friendly metadata definitions of physical data sources that enable you to query data in order to meet specific business needs. Metadata is information about the structure and content of data and the applications that process and manipulate that data. An information map contains data items and filters. A data item can point to a field in the physical data, or the data item can be a calculation. A filter contains criteria for subsetting data.

You can use an information map to import data into a project.

Note: OLAP information maps that were created in SAS Information Map Studio are not supported in SAS Enterprise Guide.

After you select an information map, you can specify the data items to include, apply filters to the data, and specify an output location. When you run the information map, the result is a SAS data set that can be used as input for further tasks and queries. SAS Enterprise Guide also generates the SAS code for the step that imports the data. You can use this import step as a stored process or as part of a larger process flow.

Adding an Information Map

About Adding an Information Map

To add an information map to a project, select **File** ⇨ **Open**. Click **SAS Folders**, and then click **Browse** to browse for the information map.

Note: You can also add an information map by using the SAS Folders section of the navigation pane. From the **Show** drop-down list, select **Information map**. Navigate through the folders to find the information map that you want to open.

Selecting an Information Map

You can select an information map from the Select Information Map window. This window displays the information maps that are available on the server that you selected. Only relational information maps are available. Multidimensional maps are not supported.

Note: SAS Enterprise Guide can open information maps with names that are greater than 32 bytes if you have applied the third maintenance release for SAS 9.2 at your site. If you are running a previous release of SAS, SAS Enterprise Guide cannot open information maps with names that are greater than 32 bytes.

To select an information map:

1 In the Select Information Map window, select the information map that you want to add to the project.

2 Click **OK** to save your selection. After you have selected the information map, you can specify the data items, filter, and output location for the information map.

Combining Filters

In the Filter Combination window, you can group, ungroup, and delete filters, change logical conditions, and change the order of filters for an information map.

To create a group of filters, press CTRL to select the filters that you want to include in the group and click .

To change a single logical operator between filters, click the logical operator and select **AND** or **OR** from the drop-down list.

To change all of the logical operators in a group, click a logical operator in the group and select **Change entire group to AND** or **Change entire group to OR** from the drop-down list.

Open an Information Map

In the Open Information Map window, you can select the data items that you want to include, apply filters to the data, and specify the output location. In an information map, category data items are indicated by , and measure data items are indicated by . By default, no data items are selected. If you do not select a data item, the data item is not included in the data source, but it remains in the information map.

After you have made your selections, click **OK** to run the information map and add it to your project. A prompt might appear if the information map references a stored process, even if you have not selected a filter.

To select the data items to include:

1 Click the **Data Items** tab.

2 Select one or more items in the **Available data items list**, and move them to the **Selected variables** list.

3 To display predefined computed items such as summarizations in the **Available data items** list, select the **Display aggregated values (group by category)** check box.

   - If you select this option, then the inclusion of these items results in an aggregated data set that is summarized according to the information map definition, displaying the measure data items across the categorical data items. Aggregation is the process of grouping data, using an operation that produces a statistic such as a sum, average, minimum, or maximum.
   - If this option is cleared, detail data is displayed in the data set. Detail data is nonsummarized (or partially summarized) factual information that pertains to a single area of interest, such as sales figures, inventory data, or human-resource data.

4 (Optional) Click **Additional data options** to complete these tasks:

   - add the folder information to the label of the data item in the **Selected variables** list
   - override the default column length (32 bytes) for calculated columns
   - rename your data items so that they comply with SAS naming rules

To apply a filter to your data:

1 Click the **Filters** tab. A list of the filters that were defined when the information map was created is displayed.

2 Select a filter and add it to the **Selected filters** list.

   Note: A filter name can contain either single quotation marks or double quotation marks. If a filter name contains both types of quotation marks, the filter is not supported in SAS Enterprise Guide.

3 (Optional) Create a custom filter.

   Note: Filters are often defined when an information map is created. However, you can also create custom filters for an information map. These custom filters are available only in your current session of SAS Enterprise Guide. These filters are not saved with the information map.

4 Specify the logical condition that joins two filters. By default, multiple filters are joined by using AND as the logical condition.

5 To apply a NOT operator to a filter, select the filter in the **Select filters** list and click to display **YES** in the **Not** column for the filter.

To specify the output location:

1 Click the **Output data** tab. The SAS server name is specified in the current profile and is associated with the library that contains the data tables.

2 To save the output to a different output table, click **Browse**.

3 To limit the number of records, select **Limit number of records in output table** to and select a number.

Accessing ODBC-Compliant Data

Inserting an ODBC Data Source

Open database connectivity (ODBC) standards provide a common interface to a variety of databases, including AS/400, dBASE, Microsoft Access, Oracle, Paradox, and SQL Server databases. SAS Enterprise Guide enables you to connect to ODBC data sources and use their data files in your project.

Note: If you need to run a task or a query on a large ODBC data source, your performance can be improved by creating a library that uses the appropriate SAS/ACCESS or database engine on the SAS server to access your data.

In order for SAS Enterprise Guide to process a task or query by using ODBC, it must first copy the data to the SAS Enterprise Guide client computer. For large volumes of data, this can be slow. If you create a library that uses SAS/ACCESS or a database engine on the SAS server, SAS Enterprise Guide does not need to copy the data to the client computer.

In addition to the information that is provided here, you should refer to the documentation that accompanies your ODBC driver. Most ODBC drivers supply a help file that you can access online. In the Windows Control Panel, open the ODBC administrator application. Within the ODBC Data Source Administrator, double-click the data source name (DSN) from the User DSN, System DSN, or File DSN tabbed window. This opens the setup window for this specific ODBC driver. Click Help to get the information that you need to configure the ODBC data source for this driver.

In SAS Enterprise Guide, you can connect to file data sources and machine data sources.

Connecting to File Data Sources

The **File Data Source** tab enables you to connect with a data source that has file DSNs. DSNs are file-based data sources that can be shared among all users who have the same drivers installed, so that they have access to the database. These data sources do not need to be dedicated to a user or local to a computer.

To select a data source:

1. Select **File** ⇨ **Open** ⇨ **ODBC**, and then click **Browse**. Ensure that the **File Data Source** tab is selected.
2. Select the correct folder from the **Look in** drop-down list.
3. Type the name of the DSN in the **DSN Name** field. Click **OK**.
4. If only one data file is associated with the data source, the data file appears in the Project pane and a data grid that contains the file appears. Otherwise, a window opens that enables you to select a data file from the data source. Select the desired file and click **OK**.

Note: Due to limitations of the dBASE ODBC driver, you must close the data grid that contains the data file before running any tasks on a dBASE data file.

Connecting to Machine (User or System) Data Sources

The **Machine Data Source** tab enables you to connect with a data source that has a user DSN or a system DSN. Machine data sources are specific to this machine and cannot be shared. User data sources are specific to a user on this machine. System data sources can be used by all users on this machine or by a system-wide service.

To select a data source:

1. Select **File** ⇨ **Open** ⇨ **ODBC**, and then click **Browse**. Click the **Machine Data Source** tab.
2. Select the data source name.
3. Click **OK**. A window specific to the data source opens. Enter the requested information and click **OK**.

Using SAS/ACCESS to Access ODBC Data

Open database connectivity (ODBC) standards provide a common interface to a variety of databases, including AS/400, dBASE, Microsoft Access, Oracle, Paradox, and SQL Server databases. You can use SAS Enterprise Guide to connect to ODBC data sources, or, if you need to run a task or a query on a large ODBC data source, you can improve your performance by creating a library that uses the appropriate SAS/ACCESS or database engine on the SAS server to access your data.

Note: You must license the appropriate SAS/ACCESS or database engine on the SAS server in order to use this type of library. If you are not sure whether the necessary engine is installed, ask your system administrator.

In order for SAS Enterprise Guide to process a task or query by using ODBC, it must first copy the data to the SAS Enterprise Guide client computer. For large volumes of data, this can be slow. If you create a library that uses SAS/ACCESS or a database engine on the SAS server, SAS Enterprise Guide does not need to copy the data to the client computer.

Note: You can use the SAS Enterprise Guide Explorer to create a library. For information about creating a library, see "Administering SAS Enterprise Guide" in *SAS Intelligence Platform: Desktop Application Administration Guide*.

If you or an administrator defines a library that uses a SAS/ACCESS engine, you can then access data in that library by using the same method that you use to access other server-based data. Select **File** ⇨ **Open** ⇨ **Servers**, and then click **Browse**.

Opening a Data Source by Using OLE DB

OLE DB provides access to a variety of data sources from within Enterprise Guide. You can use OLE DB (Object Linking and Embedding Database) to insert a data source into your SAS Enterprise Guide project.

To open a data source by using OLE DB

1 Select **File** ⇨ **Open** ⇨ **OLEDB**, and then click **Browse**. The Data Link Properties window opens.

2 On the **Provider** tab, select an OLE DB provider from the list of available providers.

3 On the **Connection**, **Advanced**, and **All tabs**, enter information for your provider-specific options. Contact your system administrator for more information.

Accessing Other Types of Data

Opening Other Data

You can use SAS Enterprise Guide to open any file that resides on your local network, a SAS server, or a SAS folder.

1 Select **File** ⇨ **Open**.

2 You can choose to open a file from your local network, a SAS server, or a SAS Folder. The list of servers includes all servers that are known to SAS Enterprise Guide. Select the location that you want to use and click **Browse**.

Note: If want to specify how SAS Enterprise Guide opens a file from your local computer, you can specify an option from the **Open as** drop-down list:

- **Default** opens the file by using the default option for the file type.
- **Text** opens the file as a text file.
- **Text with encoding** opens the Select Encoding window when you select the file to open so that you can select the appropriate character encoding to use when the text file is opened. You can change the encoding from the default encoding only for local files.
- **Program** opens the file as a SAS program file.
- **Program with encoding** opens the Select Encoding window when you select the file to open so that you can select the appropriate character encoding to use when the SAS program file is opened. You can change the encoding from the default encoding only for local files.
- **Log** opens the file as a SAS log file.
- **Log with encoding** opens the Select Encoding window when you select the file to open so that you can select the appropriate character encoding to use when the log file is opened. You can change the encoding from the default encoding only for local files.

3 Select the file that you want to open, and click **Open**.

Importing Text, HTML, or PC-Based Database Files

About the Import Data Wizard

The Import Data wizard enables you to create SAS data sets from text, HTML, or PC-based database files (including Microsoft Excel, Microsoft Access, and other popular formats). When you use the Import Data wizard, you can specify import options for each file that you import.

To import files from Microsoft Access or Microsoft Excel 2007 or later, you must have installed at least one of these applications:

- Microsoft Access or Microsoft Office 2007 or later
- the 2007 Office System Driver or the Microsoft Access Database Engine (ACE) 2010

For more information, see "File Types That SAS Enterprise Guide Reads and Writes" on page 89, "Importing JMP and Stata Files" on page 112, and "Importing SPSS Files" on page 112.

Note: Whether data from another locale imports correctly depends on the encoding value on the SAS server. If the imported data contains characters that are not supported by the encoding value and PROC IMPORT is used to read the data, then the SAS log contains transcoding errors. If the imported data contains characters that are not supported by the encoding value and a DATA step is used to read the data, then no errors appear in the log, but unsupported characters are replaced with question marks in the output data set.

Note: You must have SAS/ACCESS to PC Files to use PROC IMPORT.

Step 1: Specify the Data

When you open the Import Data wizard, the task uses the data source that you selected as the input data source. The Source Data File area lists the location, path, and data type of the source file. You cannot change the input data source after you open the Import Data wizard. To change the input data source, you must close the wizard, select a different data source, and then reopen the wizard.

To specify the data:

1 If you are importing a text file, the following options are available:

- To change the encoding that is associated with a text file, click **Encoding**. In the Select Encoding dialog box, select the encoding that you want to be associated with the text file, and click **Select**. The new encoding appears in the **Data type** field in the **Source Data File** area.
- To control the transfer of large data files between the client and the server, click **Performance** to access the performance options.

2 To save the output SAS data set to a different location, click **Browse**. Specify the server, the library, and the file name for the output SAS data set, and click **Save**. The final output location appears in the **Output SAS Data Set** area.

3 Click **Next** to select the data source.

Step 2: Select the Data Source

The options that are available for selecting the data source depend on your input data source. See the appropriate topic for your input data source.

Importing an HTML File

1 Before you can select the data to import, you must specify the data.

2 In the **Table names** box, select the table that contains the data that you want to import. The data for the selected table appears in the **Selected table's data** box.

3 If your file contains field names that you want to use as column names, select the **Table contains column names on row number** check box. In the box, enter the record number that contains the field names.

4 To specify the row number that contains your first line of data, enter a value in the **Data records start at row number** box.

5 To specify the last row that you want to import, select the **Limit the number of rows read to** check box. In the box, enter the number of the last row that you want to import.

6 Click **Next** to define the field attributes.

Importing a Table or View from Microsoft Access

1 Before you can select the data to import, you must specify the data.

2 Select the table or view to import.

3 In order for the Import Data task to accurately calculate the maximum number of bytes needed to store the data, select the **Examine all text field values in order to determine length** check box. By default, this option is not selected. The Import Data task uses the column length that is defined in the Microsoft Access table to estimate the number of bytes needed to store the column values in a SAS data set. Using the column length that is defined in the Microsoft Access

table could result in extra, unneeded bytes being added to the length of a column.

4 To rename the columns to comply with SAS naming conventions, select the **Rename columns to comply with SAS naming conventions** check box. For more information, see “Creating Variable Names in SAS Enterprise Guide” on page 73.

5 Click **Next** to define the field attributes.

Importing a Text File

1 Before you can select the data to import, you must specify the data.

2 Select the format of the text file. You can choose from these options:

- **Delimited fields** - a character, such as a comma, tab, semicolon, or space, separates the fields in the data file. To specify the delimiter for the imported data, select **Other** and specify the delimiter in the box.

  If your data includes double quotation marks or single quotation marks, you might need to specify a text qualifier. By default, the Import Data task requires fields that contain double quotation marks be enclosed in double quotation marks. An example of a text value is `Susan O'Brien said "Hello"`. If you select double quotation marks as the text qualifier, the Import Data task expects the text value to be `"Susan O'Brien said ""Hello"""`. If you select single quotation marks as the text qualifier, the Import Data task expects the text value to be `'Susan O''Brien said "Hello"'`.

- Fixed columns - the fields are aligned so that each field starts in the same byte position. SAS imports fixed column data at specific byte positions for each. Depending on the value of your encoding, a character could occupy more than one byte, which could affect the alignment of the fields. The ruler at the top of the field break box is measured in number of bytes. The light gray grid in the background of the field break box displays the number of bytes for each character. Sometimes, a single character can take up 4 bytes. If the data does not include end-of-line characters, such as a carriage return or line feed, then you must use the **Record length** option to specify the length of each record.

  In the field break box, you must insert break lines to indicate where each column starts. An arrow marks the location of each break line.

  To specify a break line, click the tick mark in the ruler that is located above the box. You can use the mouse to move a break line to a different location. To remove a break line, double-click the arrow or drag the arrow out of the box.

  Note: There are several keyboard shortcuts that you can use to create these break lines.

3 If your file contains field names that you want to use as column names, select the **File contains field names on record number** check box. In the box, enter the record number that contains the field names.

4 To rename the columns to comply with SAS naming conventions, select the **Rename columns to comply with SAS naming conventions** check box. For more information, see "Creating Variable Names in SAS Enterprise Guide" on page 73.

5 To specify the record number that contains your first line of data, enter a value in the **Data records start at record number** box.

6 To specify the last record that you want to import, select the **Limit the number of records read to** check box. In the box, enter the number of the last record that you want to import.

7 Click **Next** to define the field attributes.

Importing a Worksheet

1 Before you can select the data to import, you must specify the data.

2 Select the data to import. You can choose from these options:

- **Use a worksheet** - imports all of the observations in the first tabular region of the worksheet. By default, SAS Enterprise Guide imports the data from the upper left corner of the worksheet.

  You can change the range of cells to import by selecting the **Use a specific range of cells within the worksheet** check box and specifying the upper left cell and the lower right cell of the range that you want to import. If possible, SAS Enterprise Guide automatically discovers the range of data to import. However, SAS Enterprise Guide cannot automatically determine the values for the upper left cell and the lower right cell for the Excel XLSB file format. After selecting the **Use a specific range of cells within the worksheet** check box, you must specify the range of data that you want to import.

  If the number of rows in your data might change, select the **Expand row range as needed** option. When this option is selected, the Import Data task automatically extends the end of the range to the last row in the data. If you do not select this option, the Import Data task uses the range that you specified.

  Note: If you are importing a Microsoft Excel worksheet and Excel is installed on your computer, SAS Enterprise Guide can automate Excel to read the attributes of the data. If Excel is not installed, some features might not be available.

- **Use a predefined named range** - imports the observations from a predefined named range. You can create a predefined named range in Microsoft Excel.

3 To specify that the first row of the range that you are importing contains column names, select the **First row of range contains field names** check box. The Import Data task automatically determines whether the first row of the range contains field names. If the task determines that the first row contains column names, then this check box is selected by default. If the first row of the range does not contain column names, then you might need to clear this check box.

4. To rename the columns to comply with SAS naming conventions, select the **Rename columns to comply with SAS naming conventions** check box. For more information, see "Creating Variable Names in SAS Enterprise Guide" on page 73.

5. Click **Next** to define the field attributes.

Step 3: Define the Field Attributes

By default, all of the columns are included in the output. To remove a column from the output, clear the check box in the **Inc** column.

To define the field attributes for a column:

1. Before you can define the field attributes, you must select the data source.

2. In the table, select the field that contains the attributes that you want to edit. When a field is highlighted, you can edit the field's attributes.

   Note: To change the attributes for multiple rows, press Shift and select the rows that you want to edit. After you have selected the rows, click **Modify**. In the Field Attributes for Multiple Selections dialog box, specify the attributes that you want to change. You can specify only one value for each attribute. When you click **OK**, the value of that attribute appears in all of the selected fields

3. Specify these attributes:

   - To include the field in the output data set, select the check box in the **Inc** column. By default, all of the columns in the data source are included in the output.
   - Specify the name of the column. If you imported the field names from your data, the column names are the field names that you imported. If you chose not to import the field names, the column names are F`n`, where `n` is an integer value.
   - Specify the text that describes the contents of the column. By default, the label is the same as the column name.
   - Select the data type of the values in the column.
   - Specify the informat to use when reading in (importing) the data in the **Source Informat** column.
   - To specify the number of bytes that are used to store each of the column values in a SAS data set, enter a value in the cell of the **Len** column. By default, the length is 8 bytes for numeric data. For character data, the task determines the maximum length of data for each character field. You can override this value by specifying the width for character-based columns in the **Data > Performance** pane in the Options dialog box.

Note: To import and display an integer value greater than 8 digits, assign an appropriate format and informat.

- Specify the format to use when displaying the data in the **Output Format** column.
- Specify the informat to use when creating the output data set in the **Output Informat** column.

Note: Depending on the data type of the field, you might not be able to edit some attributes.

4 Click **Next** to specify the advanced options.

Step 4: Specify the Advanced Options

1 Before you can specify the advanced options, you must define the field attributes.

2 To include the data in the SAS program as part of a DATALINES statement, select the **Embed the data within the generated SAS code** check box. If you select this option, your output includes a SAS program that you can run again or share with other users at your site. If you do not select this option, the Import Data task accesses the data from the generated SAS code as an external file. If necessary, the data is transferred to the SAS server.

3 To import the data by using the SAS/ACCESS Interface to PC Files, select the **Import the data using SAS/ACCESS Interface to PC Files whenever possible** check box. If you do not select this option, the data is imported using a DATA step.

Note: This option is neither available nor selectable in the following circumstances:

- if you are importing data from a server that is not running Microsoft Windows.
- if this option could result in duplicate column names in the data. When you select this option, all column names are truncated to 32 characters, so depending on the column names in your data, some columns might have duplicate names. Because these column names are no longer unique and the columns are not renamed, using this option could result in the loss of data.

4 To remove any characters that might cause transmission errors, select the **Remove characters that can cause transmission errors from text-based data files** check box. You might select this option if you are importing a file that contains characters from one locale and the server does not support characters for that locale. For example, you are importing a Microsoft Excel file that

contains Japanese characters. Because the locale for the server is French, some Japanese characters are not recognized.

5 If you are importing a text file and you plan to include the code for importing this text file in a stored process, select **Generalize import step to run outside SAS Enterprise Guide**. When the stored process runs, a temporary text file is not created if the server can access the original text file. The delimiter that you specified in the Import task is used in the generated SAS code.

Note: Selecting this option assumes that the text file is in the same encoding as the SAS server. Also, any strings in the text file that include CF or LF characters cannot be enclosed in single or double quotation marks.

6 Click **Finish** to import the data.

Importing JMP and Stata Files

You can import JMP and Stata data files into a SAS Enterprise Guide project. Only the Microsoft Windows versions of Stata (*.DTA) files are supported.

To import these files, SAS Enterprise Guide uses SAS/ACCESS Interface to PC Files and the IMPORT procedure, so you must have SAS/ACCESS Interface to PC Files and Base SAS to import these data files. The IMPORT procedure is supported only on SAS servers that are running on Windows or UNIX.

To import JMP or Stata files:

1 Select **Tasks** ⇨ **Data** ⇨ **Import *data-type* file**, where data-type is JMP or Stata.

2 Select the file that you want to import and click **Open**.

3 Select the SAS server that you want to use.

4 Specify an output library and a name for the imported table. This name is used to identify the table in the project.

5 Click **OK**. The imported table is now available in the project.

Importing SPSS Files

You can import SPSS data files into a SAS Enterprise Guide project. Only the Microsoft Windows versions of SPSS (*.SAV) files are supported.

To import these files, SAS Enterprise Guide uses SAS/ACCESS Interface to PC Files and the IMPORT procedure, so you must have SAS/ACCESS Interface to PC Files and Base SAS to import these data files. The IMPORT procedure is supported only on SAS servers that are running on Windows or UNIX.

SPSS data files can contain lookup values called value labels, which are similar to SAS formats. During the import process, SAS converts these value labels to SAS format definitions. By default, these formats are saved in the Formats catalog in the Work library.

When importing SPSS data, you can create a second table that contains the formatted values instead of the raw data values. This second table does not rely on external format definitions.

Note: In order for SAS to find and use the format definitions, the formats must be stored in a catalog that is in the format search path. This path is controlled by the FMTSEARCH system option.

To import an SPSS file:

1 In the Tasks pane, select **Data** ⇨ **Import SPSS file**.

2 Select the file that you want to import and click **Open**.

3 Select the SAS server that you want to use.

4 Specify an output library and name for the imported table. This name is used to identify the table in the project.

5 (Optional) Specify the SAS library and catalog to store the SAS formats that are created from the SPSS labels. To ignore the value labels in the output, select **<remove value labels from data>** from the drop-down list.

6 (Optional) Specify any additional output options.

- To create a second table with no external formats, select the **Create additional table with formatted data values** check box. Because the formatted values might contain significant content, select the **Compress data file to save disk space** check box.
- To ensure that your imported data set contains variable names that follow the SAS naming rules, select the **Rename variables to comply with SAS naming rules** check box. The SAS log displays any renaming operations that occur during the import process.

7 Click **OK**. The imported table is now available in the project.

Accessing Text Data

You can import data using the Import Data task. To export text data, you need to create a Schema.ini file in addition to including the Text ISAM information in the registry. The Schema.ini file contains the specifics of the text data source, including how the text file is formatted and the default export format for files.

The following example shows a sample Schema.ini file for exporting a tab-delimited file.

If you are exporting data into a delimited text file, specify the format for that file as well. This example refers to a specific export option; you can specify any variation of export options at connect time. This example also corresponds to a data source name (DSN) that can be passed at connect time. All three format sections can be included in the same .ini file.

```
[Export: My Special Export]
ColNameHeader=True
Format=TabDelimited
MaxScanRows=25
CharacterSet=OEM
DateTimeFormat=mm.dd.yy.hh.mm.ss
CurrencySymbol=Dm
CurrencyPosFormat=0
CurrencyDigits=2
CurrencyNegFormat=0
CurrencyThousandSymbol=,
CurrencyDecimalSymbol=.
DecimalSymbol=,
NumberDigits=2
NumberLeadingZeros=0
```

See Also

"Importing Text, HTML, or PC-Based Database Files" on page 106

Accessing ADO Data Sources

SAS Enterprise Guide uses the Microsoft Jet database engine to access data sources for which there is an ADO (ActiveX Data Objects) interface. Examples of these types of data sources include Microsoft Excel files, Microsoft Access database files, text files, and Paradox database files.

Note: ADO data sources are not available if you are connected to a SAS Viya environment or to a server that is running SAS 9.4M8 or later.

When you use SAS Enterprise Guide to access these types of files, be aware of the following situations:

- When you have a column that contains data of mixed data types, the Jet database engine uses the data type of the majority of cells as the data type for the column. Cell values that cannot be converted to the column data type are discarded.
- The Jet database engine does not support Excel 2.1 files.
- The naming conventions for ADO objects include the following: names can be up to 64 characters long and can include any combination of letters, numbers, spaces, and characters except a period (.), an exclamation point (!), a backquote

(`), and brackets ([ ]). File names cannot begin with spaces or control characters (ASCII values 0 through 31).

For more information about the Jet database engine, see the Microsoft website.

Accessing VSAM Files

SAS Enterprise Guide does not currently have an interface for reading raw VSAM files. However, you can access VSAM data in SAS Enterprise Guide in two ways:

- You can convert the VSAM file to a SAS data set on z/OS.
- You can create a permanent DATA step view of the VSAM data. To do this, you might want to create a client library on the z/OS server so that the LIBNAME statement is automatically issued when SAS is invoked on the server. For help on creating client libraries, see your system administrator.

Editing Data

Changing the Protection Mode

You can change the mode in which a data file is opened in your project. A data file can be opened either in Edit mode, which means that you can make changes to it, or in Read-Only mode, which means that you cannot make changes to it.

Note: You can change the protection mode of a read-only data file only if you have the appropriate permissions.

To change the protection mode, click ⋮ in the data grid toolbar and select **Edit data**. You are prompted for confirmation when you change a file from Read-Only mode to Edit mode.

Note: You can also change the protection mode of a read-only data file by typing in it. You are prompted to change the file to Edit mode before you can make any changes to it.

Adding Columns to a Data File

You can add columns to data files that are opened in update mode. If the data file is opened in Read-Only mode, you must change it to Edit mode. For more information, see “Changing the Protection Mode” on page 115.

By default, the new column is added to the right of the column in which the cursor is positioned. You can type or copy data into the column or you can create an expression to compute values for the column.

Note: When you create a column, the default format and informat are assigned based on data type. For a list of all the SAS formats and informats, see *SAS Formats and Informats: Reference*.

To add columns to a data file:

1 Open the data file to which you want to add a column, and position the cursor in a column that is adjacent to where you want to add the new column.

2 Right-click the row heading and select **Insert Column**. The Insert window opens.

3 Click **General** in the selection pane.

4 If you want to insert the new column to the left of the selected column, select **To the left**. To insert the columns to the right of the selected column, select **To the right**.

5 Specify the name, label, type, length and group for the column.

   Remember these guidelines:

   - Valid column names are 1 to 32 characters long.
   - Valid column labels cannot exceed 256 characters.
   - The length in bytes of a numeric column must be between 3 and 8, and the length of a character column must be between 1 and 32,767.

   If you want to compute the new column from other columns or values, type the expression in the **Expression** box. To use the Advanced Expression Builder, click ....

   Note: An expression is applied to the column only one time and is not stored with the data file.

6 Click **Formats** and **Informats** in the selection pane to specify the format and informat for the data type.

   For a list of all the SAS formats and informats, see *SAS Formats and Informats: Reference*.

7 Click **OK** to add the new column.

Using the Advanced Expression Editor

You can use the Advanced Expression Editor to insert a new column into your data table that is computed from other columns or values.

To create an expression:

1. You can build your expression using constant values, columns, mathematical operators, and functions. You can use the Advanced Expression Editor window to add these elements, or you can enter them into the **Expression Text** box at the top of the Advanced Expression Editor window.

   - To add operators, click the buttons beneath the expression box. The operators are added to the expression.

   - To add a function, click the **Functions** tab. Select a category of functions from the **Categories** list, and then select the specific function that you want to use from the **Functions** list. Click **Add to Expression** to add the function to your expression.

     Note: For a list of all SAS functions, see *SAS Functions and CALL Routines: Reference*. The functions in the Data Quality category require SAS Data Quality Server to be licensed and installed on the server.

   - To add a column, click the **Data** tab. In the **Available Variables** box, select the table from which you want to add a column. In the **Variable Values** list, select the column that you want to add. Click **Add to Expression** to add the column to your expression.

   - To add a specific value of a column, click the **Data** tab. In the **Available Variables** box, select the column from which you want to add a value. In the **Variable Values** list, select the column value that you want to add. Click **Add to Expression** to add the value to your expression.

     Note: You can also double-click items to add them to the expression.

2. Click **OK** to save your expression.

Adding Rows to a Data File

You can add one or more rows to data files that are opened in Update mode. By default, the new rows are added below the row in which the cursor is positioned.

Note: You can also append a row to the end of your data file if the data type that you are using does not support the insertion of blank rows or columns.

To add rows to a data file:

1. Open the data file to which you want to add rows and position the cursor in the row adjacent to where you want to add more rows.
2. Right-click the row heading and select **Insert Rows**. The Insert Rows window opens.
3. Specify whether you want to insert the new rows above or below the row that you selected.
4. In the **Number of rows** box, type the number of rows that you want to add.

   Note: You can add up to 999 rows at one time.

5. Click **OK** to add the rows.

Appending a Row to a Data File

You can append a new row to the end of a data file that is opened in Edit mode. If the data file is opened in Read-Only mode, you are prompted to change it to Edit mode.

The appended row is initially displayed as the top row in the data grid so that you can type the data that you want to add. When you finish typing the data, the row is appended to the end of the file.

Note: You can also insert rows in your data file if the data type that you are using supports the insertion of blank rows.

To append a row to a data file:

1. Open the data file to which you want to append a row.
2. Right-click the row heading and select **Append row**.
3. Type in the data that you want to add to the file and press Enter. The new row is appended to the end of the file.

   Note: If you do not type in any data or if you close the data grid before you press Enter, the new row is not appended to the file.

Resizing a Row or Column

You can resize the height of a row or the width of a column. The value for the height and width is measured in pixels and can be any whole number from 1 to 999.

To adjust the row height:

1. Select the rows that you want to resize. To select the entire data grid so that you can adjust the height of multiple rows at a time, click the top, leftmost cell in the data grid. The entire data grid is highlighted.
2. Right-click the row heading and select **Height**. Enter the new value for the row height and click **OK**. If you selected the entire data grid in step 1, then all of the rows, except any rows whose height you changed manually, are changed to the new height.

To adjust the column width:

1. Select the columns that you want to resize. To select the entire data grid so that you can adjust the width of multiple columns at a time, click the top, leftmost cell in the data grid. The entire data grid is highlighted.
2. Right-click the column heading and select **Width**. Enter the new value for the column width and click **OK**. If you selected the entire data grid in step 1, then all of the columns, except any columns whose width you changed manually, are changed to the new width.

6

Working with OLAP Data

Opening an OLAP Cube

You can open cubes with at least two dimensions from any OLE DB for OLAP provider. You can choose to open cubes from the list of OLAP servers that is maintained in the metadata, from a private OLAP server, or from a SAS folder. To specify the members that you want to display in the rows and columns, you can open the cube with options. If you are accessing the OLAP Analyzer from SAS Enterprise Guide, you can insert a cube into your SAS Enterprise Guide project, and then run tasks against different views, or slices, of it.

Note: You can also open an FDB (financial database) cube or a HOLAP (hybrid OLAP) cube.

By default, when you save a project that contains a reference to an OLAP cube, the server credentials are not saved with the project. This means that the credentials must be entered each time the project is accessed. You can use the security options to change how the credentials are saved.

Note: By default, SAS 9.2 servers use Integrated Windows Authentication to securely manage server connections. This means that after you are authenticated you do not need to re-enter your user name and password to connect to a server.

In SAS Enterprise Guide, you can add the same cube to your project twice. This functionality enables you to perform different operations on the same cube and compare the results.

Note: You can also create multiple views of the same cube and desynchronize them.

To open an OLAP cube:

1 Select **File** ⇨ **Open**. Click **Servers**, and then click **Browse** to browse for the OLAP cube. Select whether you want to open a cube from the OLAP Servers list, the Private OLAP Servers list, or a SAS folder.

   Note: You can also open an OLAP cube by using the Servers or SAS Folders sections of the navigation pane. In the SAS Folders section, select **OLAP cube** from the **Show** drop-down list.

2 Select the cube that you want to open and click **Open**. By default, the cube is opened and displayed with its default view.

   Note: These steps might vary depending on the provider that you select.

Running Tasks on a Slice of a Cube

If you are accessing the OLAP Analyzer from within SAS Enterprise Guide, you can create a slice of a cube, and then run SAS tasks on that slice. For more information about specific tasks, see the Help for that task.

To run a task on a slice of a cube in SAS Enterprise Guide:

1 In SAS Enterprise Guide, open the OLAP cube in the OLAP Analyzer and create the slice of the cube that you want to use in the task. For more information, see "Manually Creating a Slice of a Cube" on page 124.

2 Make sure that the cube slice is selected in the **Project** pane or process flow, and then select the task from the **Tasks** section of the navigation pane. For more information, see "Accessing Tasks" on page 518.

Manually Creating a Slice of a Cube

If you are accessing the OLAP Analyzer from within SAS Enterprise Guide, you can manually create a slice of a cube. When you create a slice of a cube, you are creating a SAS data set that is based on the current view of an OLAP cube in the OLAP Analyzer. The cube slice is added to the project, and you can use the slice in tasks, queries, and your own SAS programs.

To create a slice of a cube:

1. In SAS Enterprise Guide, open the OLAP cube in the OLAP Analyzer and navigate in the cube to create the view of the data that you want in the slice.
2. Right-click the cube in the **Project** pane or process flow and select **Create slice**.
3. For the **Label** option, specify the label that you want to be displayed in the **Project** pane and process flow. To change the default label, click the label name and enter the new label.
4. For the **Cube** option, specify the cube from which the slice is created. To change the cube that is used, click the cube name and select another cube. You can choose from among all the cubes in the project.
5. For the **Slice type** option, select the type of slice that you want to create. You can choose from among these options:
   - **Rows, columns, and measures as separate data columns** creates separate data columns for the row headings, the column headings, and each measure value. Use the **Number of rows to scan to estimate column width** option to specify how many rows you want the OLAP Analyzer to read in order to estimate the width of a column. You must enter a value between 1 and 2,147,483,647. The default value is 500. If the width of the rows that occur after the rows that are scanned as an estimate differs from the width of the rows that are scanned, then the data slice might not display all of the columns correctly.
   - **Maintain the same layout as the OLAP table** creates the same rows and columns that are displayed in the current table view.
   - **All rows except measures** moves all of the dimensions except the measure dimensions to the rows, so that only the measure dimensions are displayed in the columns. When you select this option, the slice is generated by using SAS code instead of by using MDX code in SAS Enterprise Guide. This means that a slice that is generated by using this option can be run from within a stored process. In addition, in order to generate this data from within a stored process, you must verify that the OLE DB for OLAP provider is installed on your Workspace Server. For more information, see "Creating Stored Processes" on page 349.

6 To limit the number of rows that are included in the slice, click the **Maximum number of rows in slice** option. In the **Maximum** box, enter the number of rows that you want to include in the slice. To include all of the rows in the slice, select **No maximum**. Click **OK**.

7 For the **Output table** option, specify the name and location of the output table that is created. To change the default value, click the output table name and specify the name and location that you want to use.

8 To view the code that is used to create the slice, click **Preview MDX**.

Note: If you have selected the **All dimensions on rows except measures** option, click the **Preview Code** button to view the code.

9 Click **Run** to create the cube slice. The slice is added to the **Project** pane and process flow.

Drilling through to Detail Data

If you are viewing a cube that has drill-through detailed data enabled, then you can view the detailed data that makes up any tuple, or pairing of items from two dimensions, in that cube. If you are accessing the OLAP Analyzer from within SAS Enterprise Guide, by default, the detailed data is displayed in a separate window, and you can choose to insert the detailed data as an object in your project.

Note: If you choose to add a data item to the project, then a data object is automatically added to the **Project** pane and the process flow.

In the table view of the OLAP Analyzer, right-click the cell for which you want to view detailed data and select **Drill Through Details**.

Note: If you are accessing the OLAP Analyzer from within SAS Enterprise Guide, the first 100 rows of the detail data are displayed by default. You can choose to increase or decrease the maximum number of rows that are displayed by using the **Maximum number of rows to include detailed data** option. To access this option, select **Tools** ⇨ **Options**, and then click **OLAP Data** in the selection pane. Click **Other View Options**, and then click **Data Drill Through**.

Editing a Cube Detail Object

When you insert cube detail data as an object in your project, you can edit information about that object, such as the label, the cube that is used, and the output table. You can also specify whether you want to be able to generate the detail data from within a stored process. For more information about viewing detail data, see "Drilling through to Detail Data" on page 125.

To edit the cube detail object:

1. In the **Project** pane or process flow, right-click the cube detail object and select **Modify**.
2. To change the label, click the **Label** option and enter the new label. Click **OK** to save the new label.
3. To change the cube that is used to create the detail data, click the **Cube** option and select the cube that you want to use from the list of cubes that are in the project.
4. To change where the output data is saved, click the **Output table** option. In the Save File dialog box, browse to find the location that you want to use, and then click **Save**.
5. You can use the **Stored process compatible** option to specify whether you want to be able to generate this data from within a stored process.
   - If you select **Yes** for this option, the detail data is generated by using PROC SQL code. In addition, in order to generate this data from within a stored process, you must verify that the OLE DB for OLAP provider is installed on your SAS Workspace Server.
   - If you select **No** for this option, the detail data is generated by using MDX code in SAS Enterprise Guide.
6. To preview the code that is used to generate the detail data, click **Preview MDX**.

   Note: If you have selected **Yes** for the **Stored process compatible** option, this option is labeled **Preview Code**.

7. To save your changes and create the detail data, click **Run**. To save your changes without creating the data, click **Save**.

Changing the OLAP File Name

When you add an OLAP cube to a project, you might want to run the project first against a sample cube before you run it against your real data. When you are ready to use real data, you can change the OLAP cube that the OLAP data item points to by changing the file name or the server.

Note: The structure of the new cube must be identical to the structure of the prior cube. If the structure of the new cube is different from that of the prior cube, the OLAP data item will no longer work.

To change the OLAP file name:

1. Open the Properties window by right-clicking the OLAP cube in the **Project** pane or the process flow and selecting **Properties**.
2. Click **Advanced** in the selection pane.
3. Click **Change**. If you have any open views of the cube, a message warns you that these views will be closed.
4. Select the OLAP cube that you want the OLAP data item to point to and click **Open**.

   Note: You can also change the server, and then select another OLAP cube on that server.

5. Click **OK** in the Properties window to close the window and save your changes.

Note: Changing the file name property does not automatically update any slices that were created from the cube or update the results from any tasks that were run on the cube. You must re-create the slices and rerun the tasks in order to update your results.

About the OLAP Analyzer

The OLAP Analyzer enables you to view data that is stored in an OLAP (online analytical processing) cube.

Note: You can also use the OLAP Analyzer to view data that is stored in an FDB (financial database) cube or a HOLAP (hybrid OLAP) cube.

An OLAP cube is a specialized data storage facility that stores summarized data in a matrix-like format for fast and easy access. Users can quickly view large amounts of presummarized data at any cross section of business dimensions. A business dimension can be any aspect of the data that makes sense, such as time, geography, or product.

Note: If you are accessing the OLAP Analyzer from within SAS Enterprise Guide, it might be easier to work in the OLAP Analyzer if you maximize the space that it uses within the SAS Enterprise Guide application by selecting **View** ⇨ **Full screen**. To display the minimized windows again, click **Restore** in the upper right of the application window.

Using the OLAP Analyzer Interface

Using the Cube Manager

The OLAP Analyzer consists of the Cube Manager on the left side and the table and graph views of the data on the right side. The Cube Manager includes the following views that you can use to manage the open cube. Click the appropriate icon to open each view. The views that are available in the Cube Manager can change if you are viewing a write-enabled cube, a cube that is associated with an ESRI map, or a Cube Explorer view.

Note: The graph view of the Cube Manager is not available if you are connected to a SAS Viya environment or to a server that is running SAS 9.4M8 or later.

Note: In the Customized Items and Sets, Filters, and Conditional Highlights views, you can select which items you want to display in the views. For more information, see "Customizing the Cube Manager" on page 133.

- **Data Dimensions** – enables you to select members for the cube views by using a tree view of the default hierarchy for each dimension.
- **Customized items and sets** – enables you to add a percentage of the visual total item to the table view, to create and edit your calculated members, calculated measures, and member sets, and to view the MDX code for global calculated measures or calculated members.
- **Filters** – enables you to create and edit filters on the OLAP cube.

- **Conditional highlights** – enables you to create and edit conditional highlights for the OLAP cube.
- **Bookmarks** – enables you to create and edit bookmarks.

The table and graph views display the current view of the cube and enable you to drill in your data and expand and collapse levels.

Using the Data Dimensions View

The Data Dimensions view in the Cube Manager enables you to select members for the cube views by using a tree view of the default hierarchy for each dimension. The tree view displays all of the items that are included in the current query in bold.

Below the tree view, the Data Dimensions view also displays a table of attributes for the selected member, level, or hierarchy. The attributes can include a count of the selected item. Member items display a count of their children, and hierarchies display a count of their descendants.

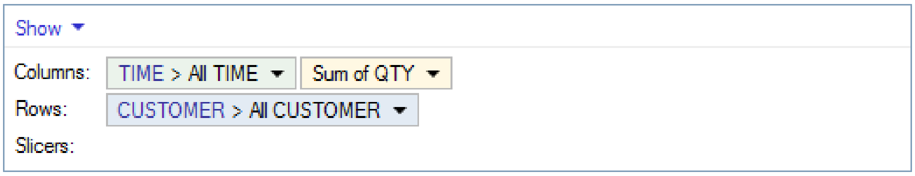
The Data Dimensions view enables you to browse or search for members, levels, and hierarchies.

Note: The search function can find only items that are in a part of the tree view that has been expanded at least once. When you start a search for an item, you cannot cancel the search.

Using the View Details Area

The View Manager is displayed by default at the top of each table, graph, and Cube Explorer view in the OLAP Analyzer. The View Details area can display the dimensions that are being used in the columns and rows, the filters that are applied to the views, and any conditional highlights that are applied to the table view. You can customize the View Details area by changing the information that is displayed in it.

From the View Details area, you can access many features of the OLAP Analyzer, including rearranging the rows and columns in the table and graph views, creating filters, and creating conditional highlights.



To customize the View Details area:

1 Click **Show** at the top of the table view, graph view, or Cube Explorer. The options that are available depend on whether a table, graph, or Cube Explorer is selected in the worksheet.

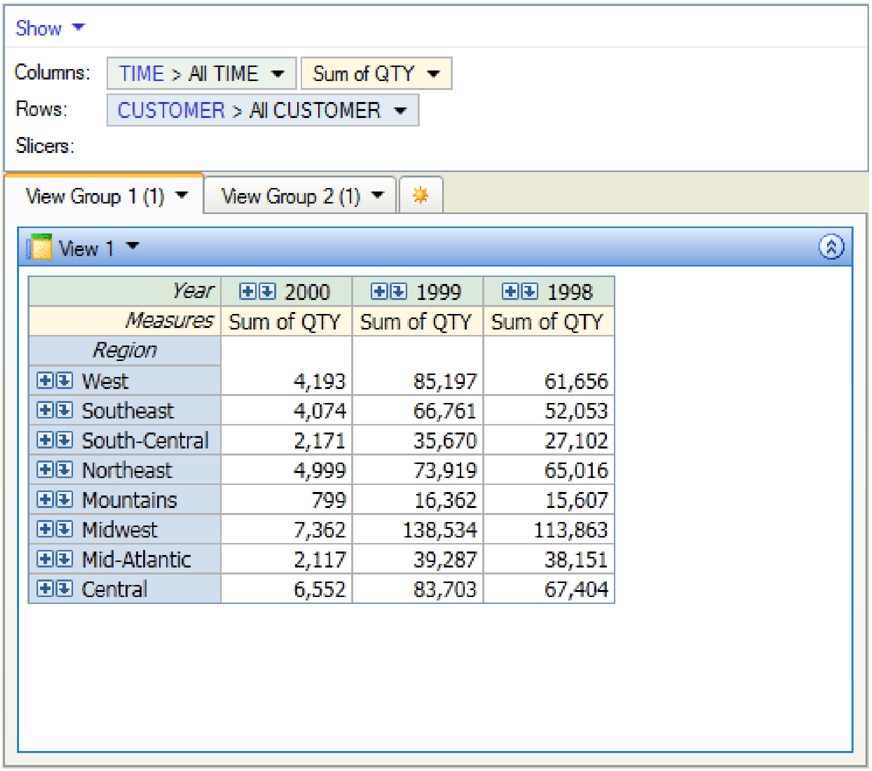
2 Specify the options for the table, graph, or Cube Explorer.

   - If a table is selected, you can display the hierarchies on the column and row axes and display any slicers, filters, and conditional highlighting. You can also display the hidden hierarchies. By default, each hierarchy that is not displayed in the table view as a row or a column is included as a hidden hierarchy. Hidden hierarchies are always set to the default member of the hierarchy, which is usually the [All] member. When you change the value of a hidden hierarchy, the hierarchy becomes a slicer.
   - If a graph or Cube Explorer is selected, you can display the measures used in the graph view, the hierarchies used to group the data, any slicers, and filters. You can also display the hidden hierarchies. By default, each hierarchy that is not displayed in the table view as a row or a column is included as a hidden hierarchy. Hidden hierarchies are always set to the default member of the hierarchy, which is usually the [All] member. When you change the value of a hidden hierarchy, the hierarchy becomes a slicer.

Using the Table View

The OLAP Analyzer consists of the Cube Manager on the left side and table and graph views of the data on the right side. The table and graph views are displayed in view groups that contain one or more views of the data. By default, when you open a cube, there are two view groups. The first view group contains the default table view, and the second view contains the default graph view.

The table view displays the current view of the cube in a table format and enables you to explore your data by drilling in your data and expanding and collapsing levels. You can create multiple table and graph views of the same cube.



Show
Columns: TIME > All TIME | Sum of QTY
Rows: CUSTOMER > All CUSTOMER
Slicers:
View Group 1 (1)
View Group 2 (1)
View 1

Year	2000	1999	1998
Measures	Sum of QTY	Sum of QTY	Sum of QTY
Region			
West	4,193	85,197	61,656
Southeast	4,074	66,761	52,053
South-Central	2,171	35,670	27,102
Northeast	4,999	73,919	65,016
Mountains	799	16,362	15,607
Midwest	7,362	138,534	113,863
Mid-Atlantic	2,117	39,287	38,151
Central	6,552	83,703	67,404

TIP Dimension and level names are displayed in italics.

At the top of the table view, the View Details area displays the measures and dimensions that are displayed in the current view. You can use the View Details area to change to another member in the same dimension and to rearrange the order of the rows and columns. You can also view and edit the filters and highlights that are applied to the view. By default, each dimension that is not displayed in the table view as a row or column is included as a hidden hierarchy.

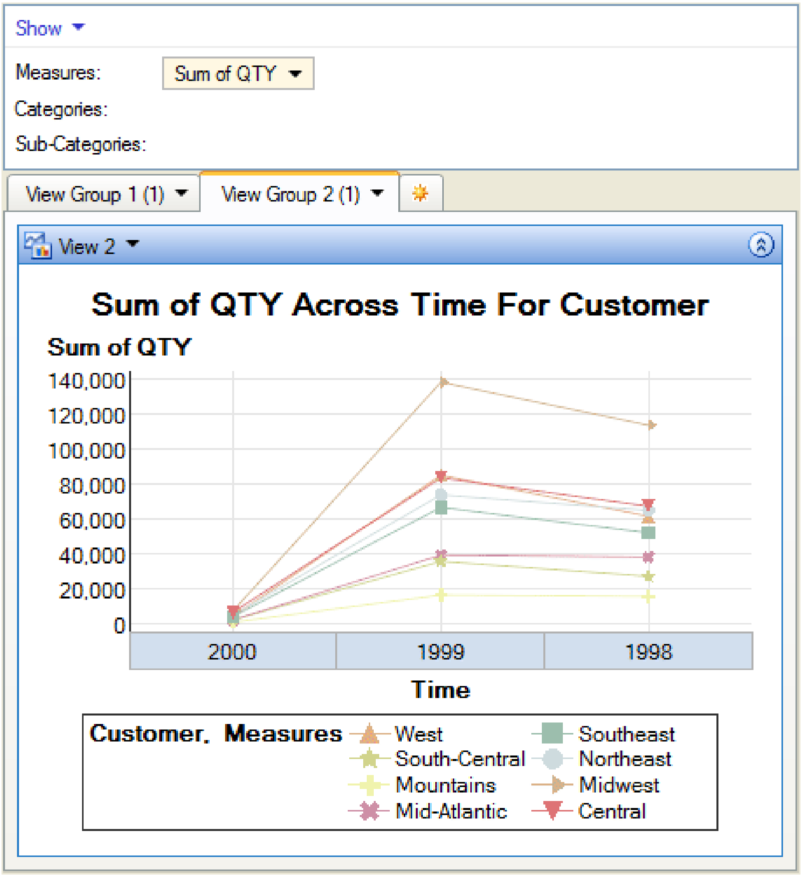
If you have enabled the **Show raw values on data cells** option for the view, you can move the mouse pointer over each data cell to display the raw, unformatted value of the cell. You can also use the Properties dialog box for the view to set other options for the current view of the active cube. Only one table view can be open at a time.

Note: You can set options for the table view of all cubes by using the Options dialog box. These options appear under the **OLAP Data** heading in the selection pane.

Using the Graph View

The OLAP Analyzer consists of the Cube Manager on the left side and table and graph views of the data on the right side. The table and graph views are displayed in view groups that can contain one or more views of the data. By default, when you open a cube, there are two view groups. The first view group contains the default table view, and the second view group contains the default graph view.

Note: The graph view of the Cube Manager is not available if you are connected to a SAS Viya environment or to a server that is running SAS 9.4M8 or later.



You can choose from among many different chart types in the OLAP Analyzer. By default, the initial graph view is created as an automatic chart. To generate an automatic chart, the OLAP Analyzer chooses the best chart for the data based on the number of measures and whether the data includes a time dimension. You can also customize the appearance of the graph view by changing the graph style and by selecting a predefined color scheme. For more information, see "Customize the Appearance of the Graph View" on page 187.

You can also specify preferences for how graph views are created. For more information, see "Specifying Options for Charts" on page 190.

Note: You can set preferences for the graph view of all cubes by using the Options dialog box. These options appear under the **OLAP Data** heading in the selection pane.

Synchronizing the Cube Views

By default, all of the cube views in the OLAP Analyzer are synchronized so that when you make a change to one view, such as expanding, drilling, or filtering, all of the other views are updated as well. If you want to modify one view without affecting any of the other views, you can choose to desynchronize the views. When the views are desynchronized, you can resynchronize them to match the query in a view that you specify.

Note: Cube Explorer views are always desynchronized and cannot be synchronized with another view. For more information, see "Using the Cube Explorer" on page 134.

- On the OLAP Analyzer toolbar, select **Desynchronize Views**. The cube views are desynchronized and can now be updated independently.
- On the OLAP Analyzer toolbar, select **Synchronize Views To**. From the pop-up menu, select the view that you want to use as the basis for synchronizing all of the other views. All of the cube views are synchronized.

Customizing the Cube Manager

In the Customized Items and Sets, Filters, and Conditional Highlights views of the Cube Manager, you can change how items in those views are displayed and grouped, and you can sort them. You can also search for items in each view.

To change how items are displayed:

1. In the upper right corner of the view that you want to change, click the current display type. By default, all items in the group are displayed. For example, in the Customized Items and Sets view, the default display type is **All items and sets** and the items are grouped by type.
2. From the pop-up menu, select the display option and grouping option that you want to use.

Using the Cube Explorer

About the Cube Explorer

In addition to viewing your OLAP data in the table view and graph view, you can also use the Cube Explorer to navigate in your data. The Cube Explorer enables you to select a dimension, expand a node within that dimension, and then change to another level or dimension. You can use this method to explore the data and find paths through the cube that might point to areas that need further analysis.

Creating a Cube Explorer View

You can create a new Cube Explorer view by using the Create New Cube Explorer View Wizard. In the wizard, you can select the dimension from which you want to start and one or more measures to analyze. By default, each dimension that is not displayed in the view is included as a hidden hierarchy. The default value for each dimension is the [All] member, which includes all members of that dimension. You can change the filter to specify a specific filter value that you choose.

When you create a view in the Cube Explorer, it is added as another view in the worksheet. You can print the Cube Explorer view and save it as a JPEG or BMP file.

To create a Cube Explorer view:

1 On the toolbar, select **New** ⇨ **Explorer**. The Create New Cube Explorer View Wizard appears with a list of dimensions that are defined for the current cube.

2 Select the dimension from which you want to start, or expand the list of dimensions to select a specific member from which you want to start.

3 Click **Next**.

4 Select one or more measures to analyze from the list of measures that are defined in the current cube. You can select up to 10 measures to analyze, but one measure must be defined as the primary measure. The cube's default measure is defined as the primary measure and is used to calculate the parent-child relative contribution percentages that are displayed at each node.

5 If you want to change the primary measure, select the measure that you want to use as the primary measure and click **Set as Primary Measure**.

6 Click **Next**.

7 If you want to create a slicer based on the other dimensions in the cube, select the dimension that you want to use in the slicer, and click .... Repeat this step to add slicers for other dimensions.

8 Click **Finish**. The Cube Explorer opens with the data that you selected.

Navigating in the Cube Explorer

In the Cube Explorer, you can expand and collapse nodes and sort nodes by name or value. You can select a node and change to another dimension. This enables you to view a subset of data in another dimension at the level that you choose based on a member in the current dimension.

You can also copy a level in the current view in the Cube Explorer to the table view to use it as a starting point for more complex analysis.

> TIP To see an example of how you can use the Cube Explorer to navigate in your data, see "Exploring Your Data with the Cube Explorer" on page 136.

- To expand a node to a lower level within the same dimension, click the down arrow on the node that you want to expand, or right-click the node and select **Expand.**
- To collapse an expanded node, click the up arrow on the node that you want to collapse, or right-click the node and select **Collapse**.

  > TIP If there is not enough room to display all of the nodes in a row, then you can use the right and left arrow keys or the Home and End keys to navigate among the nodes.

- To sort the nodes in a level, right-click a node in the level that you want to sort and select **Sort.** From the pop-up menu, you can select either of the following options:
  - □ **Name** ⇨ **Ascending** or **Name** ⇨ **Descending** to sort the nodes by member name
  - □ ***measure-name*** **Ascending** or ***measure-name*** **Descending** to sort the nodes by the measure value
- To view data in another level from a level in the current dimension, click a node in the level that you want to use and select **Change to** ⇨ ***dimension-namelevel-name***. A new row of nodes is added to the Cube Explorer for the new level.
- To copy a level to the table view, right-click a node in the level that you want to copy and select **Create a Table for this Level**. Because the other table and graph views that you have created are synchronized, a warning message tells you that all existing table and graph views will be changed to the new table. The new table view opens with the level that you selected. If the table is already open, then the current table view is changed to reflect the selected level.

Modifying a Cube Explorer View

You can modify an existing Cube Explorer view by adding and removing measures. In addition, you can change the parent-child relative contribution percentage that is displayed above each node.

Note: You can also use the View Details area to modify a view after you have created it.

To add a measure to a Cube Explorer view:

1. In the View Details area, make sure that the measures are displayed.
2. Right-click the area to the right of the measures and select **Add Measure *measure-name***. The new measure is added to each node in the view.

To remove a measure from a Cube Explorer view:

1. In the View Details area, make sure that the measures are displayed
2. To remove a measure from the view, click the inverted triangle icon beside the measure that you want to remove and select **Remove from Explorer**. The measure is removed from all of the nodes in the view.

To change the percentage:

1. In the View Details area, ensure that the measures are displayed.
2. Click the inverted triangle icon beside any measure and select **Show Percentage For** ⇨ ***measure-name***. The nodes are updated to display the new percentage.

Exploring Your Data with the Cube Explorer

You can use the Cube Explorer to subset data in one dimension based on a member that you choose from another dimension. In this example, using a sample cube, you look for trends in your sales data and get ideas for more in-depth analyses. You look for the products that sold the best and the regions in which they were sold.

You find that the electronics category of products are the most popular, especially among female customers in the Midwest. And these customers tend to do most of their shopping in the fourth quarter, before the holidays.

Note: You could also, for example, explore the data to find what products sold poorly or exactly what product numbers female buyers were purchasing in the Midwest.

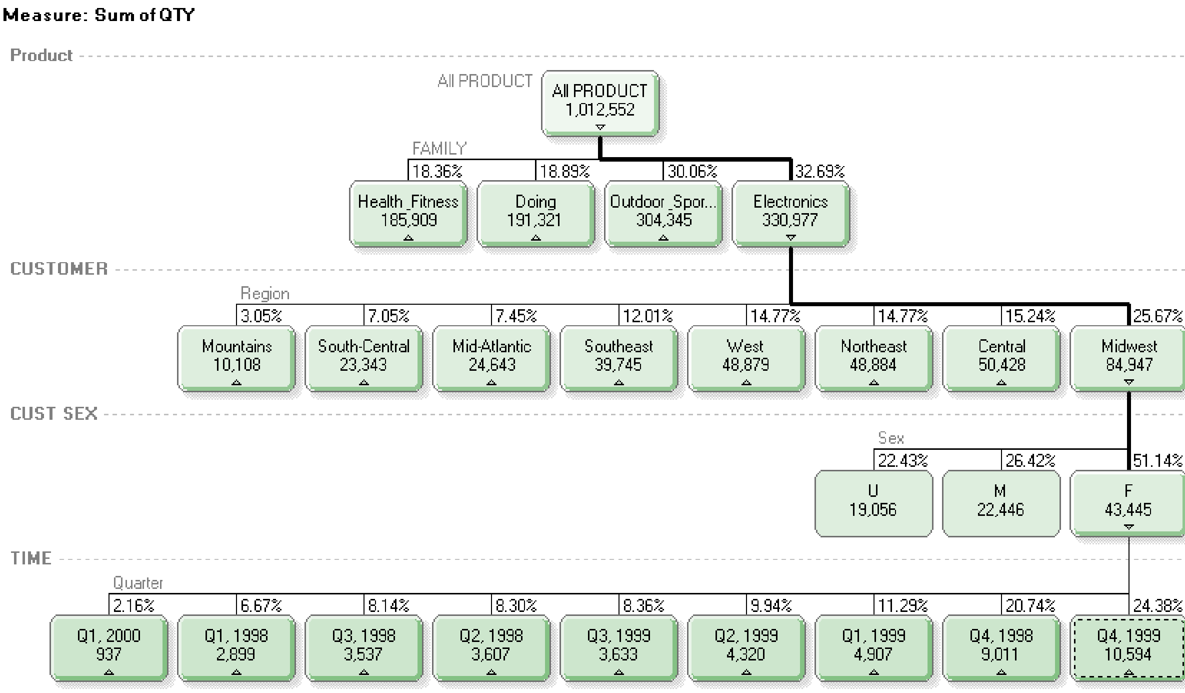
Example: Exploring sales of electronics products

1. Expand the All Product node to display the Family level. There are four families of products. The greatest quantity of products, 32.69%, is sold in the electronics family.

2. From the Electronics node, change to the Region level in the Customer dimension. The greatest number of sales, 25.67%, occur in the Midwest region.

3. From the Midwest node, change to the Sex level in the CustomerSex dimension. Female customers purchase 51.14% of products in this region

4. From the F node, change to the Quarter level in the Time dimension. The greatest number of sales, 24.38%, are made in the fourth quarter.



Customizing the Appearance of the Cube Explorer

You can customize the appearance of the Cube Explorer by changing the labels that are displayed, the look of the parent nodes, and the color shadings.

In the Cube Explorer panel, select the options that you want to change:

- **Show percentage** – displays the relative contribution of each child node to its parent by using the primary measure that was defined when you created the view.

  Note: The percentages might not always add up to 100%, depending on the statistic that is used in the measure and whether calculated members are included in the results. The percentages would not add up to 100% in the following examples:

- The primary measure represents an average value, such as the average cost per unit. Because the values are averages instead of precise values, the total might not be exactly 100.
- The primary measure includes values from a calculated member that aggregates members from other levels. For example, you might have a cube with a Cars dimension that includes levels for several car companies as well as a Red Cars calculated member that includes all red cars. In this case, the total percentage would be greater than 100 because the red cars would be counted twice: once with car company level and again with the Red Car calculated member. Percentages from calculated members are displayed with an asterisk.

- **Show level labels** – displays the name of each level within its dimension.
- **Show calculated members in results** – includes calculated members in the current Cube Explorer view.
- **Orientation** – specifies how you want to display the nodes in the Cube Explorer.
- **3D look for parent nodes** – displays the parent nodes with a three-dimensional appearance.
- **Apply color shadings based on** – determines how color shading is applied. If you select Dimension levels, then nodes are shaded darker the deeper in the hierarchy that they are. If you select Measure values, then nodes are shaded darker as the value increases.

  To change the colors that are used for the shading, select a color from the color drop-down list.

  To change the background color of the Cube Explorer, select a color from the **Background color** drop-down list.

Any changes that you make are immediately applied to the Cube Explorer.

Navigating in an OLAP Cube

Creating a Cube View

The OLAP Analyzer displays the OLAP cube with two default view groups that contain the default table view and the default graph view. You can create additional view groups, and each view group can contain one or more views of the cube. For more information, see “Understanding View Groups” on page 139.

When you create a view, it is automatically added to the current view group, and you can choose from among many different view types. If a view group contains multiple views, the views are displayed vertically within the view group. When you add views to a view group, you can resize the views, move them up and down within the view group, and move them to another view group. You can also minimize a view within a view group.

To create a view:

1 On the OLAP Analyzer toolbar, select **New** ⇨ ***view-type***.

2 Select the type of view.

Note: If you select **Automatic Chart**, the OLAP Analyzer chooses the best chart for the data based on the number of measures and whether the data includes a time dimension.

Understanding View Groups

The OLAP Analyzer displays your OLAP cube in a tabbed interface with two default view groups. Each view group is displayed as a tabbed page and can contain one or more views, such as a table view or a graph view, of the cube. You can create view groups and move existing view groups to the left and right within the OLAP Analyzer.

When you create a view, it is automatically added to the current view group. If a view group contains multiple views, the views are displayed vertically within the view group. When you add views to a view group, you can resize the views, move them up and down within the view group, and move them to another view group.

To create a view group, on the OLAP Analyzer toolbar, select **New** ⇨ **View Group**.

Adding and Rearranging Members in the Table and Graph Views

In the OLAP Analyzer, you can change the members that are displayed in the table and graph views. You can add new members to the views, and you can also move members between the rows and columns.

To add a member to the table and graph views:

1 In the Cube Manager, click to display the Data Dimensions view.

2 Expand the tree as necessary, and select one or more members or member sets that you want to add.

Note: If you select a member of a different type, then all previously selected members are deselected and only the current member remains selected. For example, if you have selected three measures, and you press Ctrl + click to also select a level, then the measures are automatically deselected and only the level is selected

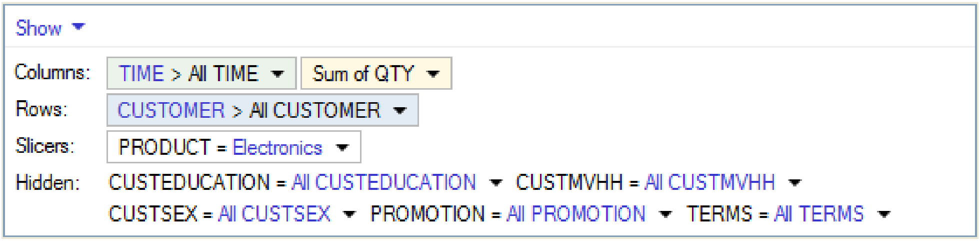
3 Right-click the selected member and add to the row axis, to the column axis, or the slicer dimension.

The table and graph views are updated.

Understanding the Hidden Hierarchies and Slicers

By default, each hierarchy that is not displayed in the table view as a row or a column is called a hidden hierarchy. The default value for each hidden hierarchy is the default member for the hierarchy. (The default member for a hierarchy is usually the [All] member, which includes all members of that hierarchy.)

You can assign another member in the hierarchy to a hidden hierarchy to create a slicer. Slicers restrict the data that is displayed. For example, in a cube that contains sales data, you could display the sales by customer region and year. You could create a slicer to restrict this data by using the Products hierarchy to display only the sales data for electronic products as shown below.



Show

Columns: TIME > All TIME | Sum of QTY

Rows: CUSTOMER > All CUSTOMER

Slicers: PRODUCT = Electronics

Hidden: CUSTEDUCATION = All CUSTEDUCATION | CUSTMVHH = All CUSTMVHH | CUSTSEX = All CUSTSEX | PROMOTION = All PROMOTION | TERMS = All TERMS

You can view the hidden hierarchies and slicers in the View Details area of the OLAP Analyzer.

To create a slicer:

1 To display the hidden hierarchies, click **Show** in the View Details area, and then select **Hidden**. Click **OK**.

2 Click the hidden hierarchy that you want to edit. The Select Members window appears.

3 Use the **Browse**, **Search**, and **Use Filter** tabs to select the members that you want to use.

4 You can add members in the following ways:

- To add only the selected member, click **Add**.
- To add the selected member and all of its children members, click the inverted triangle on the **Add** button and select **Selected and children**.
- To add the selected member and all of its descendants, click the inverted triangle on the **Add** button and select **Selected and all descendants**.
- To add all of the members, click **Add All**.

Note: You cannot select any members that are ancestors of other selected members.

5 Click **Select** to update the hierarchy. The slicer is applied to the cube and added to the list of slicers in the View Details area.

Drilling Up and Down within an OLAP Cube

The OLAP Analyzer displays the OLAP cube in a table view that you can use to drill up and down in your data. You can use drilling to navigate in your data from the most summarized levels to the most detailed levels. You can drill down on all members of a level, or you can drill down on a specific member of the level. You can also drill in the graph view. Because the table view and graph views are synchronized, all views are updated when you drill up or down in one view.

For dimensions with multiple hierarchies, you can change the drill-down path by changing the navigation hierarchy for members or for hierarchies, levels, and measures.

To drill on all members of a level in the table view, right-click on the level heading that you want to drill down on and select **Drill Down**. From the submenu, select the level to which you want to drill down.

To drill on a specific member of a level in the table view, In the row or column heading for the member that you want to drill down on, click ⤓.

To drill in the graph view, right-click the area of the graph where you want to drill, and select **Drill Down on** or **Drill Up to**. A submenu lists the rows and columns that you can drill on. You can also choose to drill on both rows and columns. If you double-click an area of the graph, the default behavior is to drill on all of the rows and columns.

To change the navigation hierarchy for a member:

1 In the table view, right-click on the row or column heading that you want to change and select **Replace With**.

2 From the pop-up menu, you can select a level in the current hierarchy, another hierarchy, or another dimension to use for navigation. The current member is removed from the table view and is included instead as a hidden hierarchy.

To replace the navigation hierarchy for a hierarchy, level, or measure, right-click on a level or measure heading in the table view, or click beside a dimension or measure in the View Manager, and select **Replace With**. From the pop-up menu, you can select a replacement for the current level, hierarchy, or measure.

Note: You cannot change the navigation hierarchy in dimensions that have only one default hierarchy defined.

Isolating and Removing Members

You can isolate one or more members in the table or graph views of the OLAP Analyzer so that no other members are displayed. You can also remove specific members from the view. Because the table and graph views are synchronized, all views are updated when you isolate or remove members in one view.

TIP You can return to previous views of the cube by clicking the **Back** button. You can also restore the default view of the cube.

To isolate or remove a member. right-click the row or column heading in the table view that you want to keep or remove, and select **Keep Only *member-name* in *hierarchy-name*** or **Remove *member-name***. You can also right-click the appropriate member label in the graph view and select **Keep Only *hierarchy-name*, *member-name*** or **Remove *member-name***.

Browsing the Different Views of an OLAP Cube

After you have created different views, or slices, of an OLAP cube, you can browse backward and forward through the different views. You can also reset your view to the default view.

If you want to save a specific view, you can save it as a bookmark. You can also specify a bookmark that you create to be the default view.

To restore the default view:

1. Click 🔖 in the Cube Manager to open the Bookmarks view.

2. Click **Initial view**. The default view is restored.

Setting and Using Bookmarks

You can set a bookmark in an OLAP cube to save a particular view, or slice, of the cube. You can then use the Bookmarks view in the Cube Manager to go to that saved bookmark directly without navigating through the grid again.

You can also edit, rename, and delete bookmarks that you have previously saved as well as restore the default view. To change the default view, specify a bookmark that you create to be the new default view.

Note: If you are accessing the OLAP Analyzer from within SAS Enterprise Guide, bookmarks are saved with your SAS Enterprise Guide project.

Note: If you create a bookmark that includes an item such as a filter, percent of visual total, conditional highlight, calculated measure, or calculated member, then that bookmark is automatically deleted if the item is deleted.

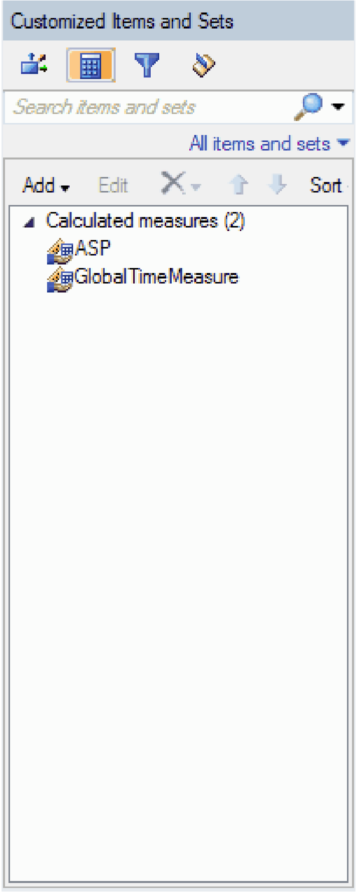
Analyzing the Data in an OLAP Cube

About Customized Items and Sets

The Customized Items and Sets view in the Cube Manager enables you to customize your cube views with items that are derived from other values within the cube. You can also create groups of members from the same dimension.

Here are the customized items and sets that you can create:

- calculated measures
- calculated members
- percent of visual total items
- member sets



Creating Calculated Measures

What Is a Calculated Measure?

You can add a calculated measure to a dimension in an OLAP cube. A calculated measure is a special dimension that usually represents numeric data values that are analyzed. Calculated measures can be computed from other values and must return either a string or a numeric value. However, most OLE DB for OLAP providers support only numeric values.

Note: This option is available only for OLAP cubes that were created with SAS 9 or that were created for other OLE DB for OLAP providers that support calculated measures.

Count Analysis

You can use this type of analysis to calculate the discount count of a measure.

1. If you are using the OLAP Analyzer, click ▦ in the Cube Manager. In the Customized Items and Sets view, click **Add Calculated Member**.

   If you are working in a PivotTable, click **OLAP Options** on the **SAS** tab, and select **Calculated Members** ⇨ **New**.

   The Add Calculated Member wizard opens.

2. In the **Name** box, enter the name of the new calculated measure.

3. For the type of analysis, select **Special analysis**, and from the drop-down list, select **Count analysis**. Click **Next**.

4. Select the type of calculation for the new measure:

   - **Member Count** - counts the number of members in the hierarchy or counts the number of members on a specified level.
   - **Unique Item Count (Discount Count)** - counts the number of nonempty values for a level in a hierarchy.

5. Select the hierarchy to use in the calculation. You can also select the level of the hierarchy to use. Selecting a level is usually recommended. If you are calculating a distinct count, then select the measure to use in the calculation.

   Note: If you are working with an SAP cube, then you must select a level.

   Click **Next**.

6. For a unique item count, select the measure to use.

7. Select the format that you want to apply to the new calculated measure. If you are using a SQL server cube, then the formats in this list are similar to formats in Microsoft Excel. If you are using a SAS cube, then the formats in the list are SAS formats. You can also specify a custom format.

8. Specify how the new measure is available:

   - **Locally, to me** - the calculated measure is valid for as long as you are using the OLAP cube in your current session. After the calculated measure has been created, it can be used by all queries in the same session. A calculated measure with this scope is available only to the person who created it.
   - **Publicly, to all users at all times** - the calculated measure is available to anyone who connects to that cube. This option is available only on a SAS OLAP server. If you are not using a SAS OLAP server, then this option is disabled and the value is set to Locally, to me. You must also have WriteMetadata permission on the metadata server to use this option. If you do not have this permission, then an error message is displayed. For more information, contact your SAS administrator.

Note: If the calculation for the new measure includes a measure with local scope, then the scope option is disabled and the value is set to **Locally, to me**. When a measure with local scope is used to create another calculated measure, the new calculated measure must be created with local scope as well.

Note: You cannot specify the availability of a calculated measure for an OLAP cube that is opened in a PivotTable.

9 In the **Solve order** box, enter the number that specifies the priority for evaluating the calculated member or calculated measure. A higher solve order results in the calculated member or calculated measure being evaluated later in the expression. The solve order usually is not needed, but it can be used when there is more than one calculated member or calculated measure in a query.

Click **Next**.

10 Review a summary of your selections and the MDX expression. Click **Finish** to create the new calculated measure.

Note: If you are working with an SAP cube, you cannot specify the format, availability, and solve order options. Any calculated measures that you create are available only to you.

Custom Calculation

To create a calculated measure for a custom calculation:.

1 If you are using the OLAP Analyzer, click 🖩 in the Cube Manager. In the Customized Items and Sets view, click **Add Calculated Member**.

If you are working in a PivotTable, click **OLAP Options** on the **SAS** tab, and select **Calculated Members** ⇨ **New**.

The Add Calculated Member wizard opens.

2 In the **Name** box, enter the name of the new calculated measure.

3 For the type of analysis, select **Use MDX statement to create custom analysis**. Click **Next**.

4 Specify the formula that you want to apply to the new calculated measure. If you are using a SQL server cube, then the formats in this list are similar to formats in Microsoft Excel. If you are using a SAS cube, then the formats in the list are SAS formats. You can also specify a custom format.

5 Select the format that you want to apply to the new calculated measure. If you are using a SQL server cube, then the formats in this list are similar to formats in Microsoft Excel. If you are using a SAS cube, then the formats in the list are SAS formats. You can also specify a custom format.

6 Specify how the new measure is available:

- **Locally, to me** - the calculated measure is valid for as long as you are using the OLAP cube in your current session. After the calculated measure has been created, it can be used by all queries in the same session. A calculated measure with this scope is available only to the person who created it.
- **Publicly, to all users at all times** - the calculated measure is available to anyone who connects to that cube. This option is available only on a SAS OLAP server. If you are not using a SAS OLAP server, then this option is disabled and the value is set to Locally, to me. You must also have WriteMetadata permission on the metadata server to use this option. If you do not have this permission, then an error message is displayed. For more information, contact your SAS administrator.

Note: If the calculation for the new measure includes a measure with local scope, then the scope option is disabled and the value is set to **Locally, to me**. When a measure with local scope is used to create another calculated measure, the new calculated measure must be created with local scope as well.

Note: You cannot specify the availability of a calculated measure for an OLAP cube that is opened in a PivotTable.

7 In the **Solve order** box, enter the number that specifies the priority for evaluating the calculated member or calculated measure. A higher solve order results in the calculated member or calculated measure being evaluated later in the expression. The solve order usually is not needed, but it can be used when there is more than one calculated member or calculated measure in a query.

Click **Next**.

8 Review a summary of your selections and the MDX expression. Click **Finish** to create the new calculated measure.

Note: If you are working with an SAP cube, you cannot specify the format, availability, and solve order options. Any calculated measures that you create are available only to you.

Relative Contribution Analysis

You can use this type of analysis to determine how much the value of a measure contributes to the total value or to the value of a parent dimension. You can choose to display this calculation as a percentage.

To create a new measure for relative contribution analysis:

1 If you are using the OLAP Analyzer, click ▦ in the Cube Manager. In the Customized Items and Sets view, click **Add Calculated Member**.

If you are working in a PivotTable, click **OLAP Options** on the **SAS** tab, and select **Calculated Members** ⇨ **New**.

The Add Calculated Member wizard opens.

2 In the **Name** box, enter the name of the new calculated measure.

3 For the type of analysis, select **Special analysis**, and from the drop-down list, select **Relative Contribution analysis**. Click **Next**.

4 Select the type of calculation for the new measure:

- **Member contribution relative to a selected member** - how much the member contributes to a selected value. For example, you might need to know how much the sales of a product contributed to your company's yearly sales.
- **Member contribution relative to its parent** - how much the member contributes to the value of the parent member. For example, you might need to know how much the sale for a state contributed to the quarterly sales for a geographic region.

5 Select the hierarchy to use in the calculation. You can also select the level of the hierarchy to use. Selecting a level is usually recommended. If you are calculating a distinct count, then select the measure to use in the calculation.

Note: If you are working with an SAP cube, then you must select a level.

Click **Next**.

6 If you are calculating a member's contribution relative to a selected value, then select the member that contains the value to use in the comparison.

7 Select the measure to use in the calculation. Click **Next**.

8 Select the format that you want to apply to the new calculated measure. If you are using a SQL server cube, then the formats in this list are similar to formats in Microsoft Excel. If you are using a SAS cube, then the formats in the list are SAS formats. You can also specify a custom format.

9 Specify how the new measure is available:

- **Locally, to me** - the calculated measure is valid for as long as you are using the OLAP cube in your current session. After the calculated measure has been created, it can be used by all queries in the same session. A calculated measure with this scope is available only to the person who created it.
- **Publicly, to all users at all times** - the calculated measure is available to anyone who connects to that cube. This option is available only on a SAS OLAP server. If you are not using a SAS OLAP server, then this option is disabled and the value is set to Locally, to me. You must also have WriteMetadata permission on the metadata server to use this option. If you do not have this permission, then an error message is displayed. For more information, contact your SAS administrator.

Note: If the calculation for the new measure includes a measure with local scope, then the scope option is disabled and the value is set to **Locally, to me**. When a measure with local scope is used to create another calculated measure, the new calculated measure must be created with local scope as well.

Note: You cannot specify the availability of a calculated measure for an OLAP cube that is opened in a PivotTable.

10 In the **Solve order** box, enter the number that specifies the priority for evaluating the calculated member or calculated measure. A higher solve order results in the calculated member or calculated measure being evaluated later in the expression. The solve order usually is not needed, but it can be used when there is more than one calculated member or calculated measure in a query.

Click **Next**.

11 Review a summary of your selections and the MDX expression. Click **Finish** to create the new calculated measure.

Note: If you are working with an SAP cube, you cannot specify the format, availability, and solve order options. Any calculated measures that you create are available only to you.

Basic Analysis

You can use this type of analysis to calculate the sum, the difference, the product, the ratio, and the percent change of two measures.

Note: This option is available only for OLAP cubes that were created with SAS®9 or that were created for other OLE DB for OLAP providers that support calculated members.

1 If you are using the OLAP Analyzer, click 🖩 in the Cube Manager. In the Customized Items and Sets view, click **Add Calculated Member**.

If you are working in a PivotTable, click **OLAP Options** on the **SAS** tab, and select **Calculated Members** ⇨ **New**.

The Add Calculated Member wizard opens.

2 In the **Name** box, enter the name of the new calculated measure.

3 For the type of analysis, select **Basic analysis**. Click **Next**.

4 Select the type of calculation for the calculated measure:

- **Sum** - the combined value of two measures.
- **Difference** - the difference between two measures.

- **Product** - the multiplied value of two measures.
- **Ratio** - the relationship between two measures.
- **Percent change** - the percent change between two measures.

After you have selected the calculation type, use the drop-down lists in the **Calculations** area to specify the measures to use in the calculation. Click **Next**.

5 Select the format that you want to apply to the new calculated measure. If you are using a SQL server cube, then the formats in this list are similar to formats in Microsoft Excel. If you are using a SAS cube, then the formats in the list are SAS formats. You can also specify a custom format.

6 Specify how the new measure is available:

- **Locally, to me** - the calculated measure is valid for as long as you are using the OLAP cube in your current session. After the calculated measure has been created, it can be used by all queries in the same session. A calculated measure with this scope is available only to the person who created it.
- **Publicly, to all users at all times** - the calculated measure is available to anyone who connects to that cube. This option is available only on a SAS OLAP server. If you are not using a SAS OLAP server, then this option is disabled and the value is set to Locally, to me. You must also have WriteMetadata permission on the metadata server to use this option. If you do not have this permission, then an error message is displayed. For more information, contact your SAS administrator.

Note: If the calculation for the new measure includes a measure with local scope, then the scope option is disabled and the value is set to **Locally, to me**. When a measure with local scope is used to create another calculated measure, the new calculated measure must be created with local scope as well.

Note: You cannot specify the availability of a calculated measure for an OLAP cube that is opened in a PivotTable.

7 In the **Solve order** box, enter the number that specifies the priority for evaluating the calculated member or calculated measure. A higher solve order results in the calculated member or calculated measure being evaluated later in the expression. The solve order usually is not needed, but it can be used when there is more than one calculated member or calculated measure in a query.

Click **Next**.

8 Review a summary of your selections and the MDX expression. Click **Finish** to create the new calculated measure.

Note: If you are working with an SAP cube, you cannot specify the format, availability, and solve order options. Any calculated measures that you create are available only to you.

Time Series Analysis

You can use this type of calculated item to calculate the sum or average of a measure during a specific time period, and to compare activity between time periods, the amount of growth between time periods, and balance calculations.

Note: If your OLAP cube does not contain a time dimension, this analysis is not available.

To create a calculated member for time series analysis:

1 If you are using the OLAP Analyzer, click ▦ in the Cube Manager. In the Customized Items and Sets view, click **Add Calculated Member**.

  If you are working in a PivotTable, click **OLAP Options** on the **SAS** tab, and select **Calculated Members** ⇨ **New**.

  The Add Calculated Member wizard opens.

2 In the **Name** box, enter the name of the new calculated measure.

3 For the type of analysis, select **Special analysis**, and from the drop-down list, select **Time series analysis**. Click **Next**.

4 Select the type of calculation for the new measure:

  - **Rolling totals** - the cumulative total of a specified measure for a specified time period. For example, you could calculate the cumulative sales of a product from the beginning of the year to the current date.
  - **Rolling averages** - the cumulative average of a specified measure for a specified time period. For example, you could calculate the average sales of a product from the beginning of the year to the current date.
  - **Totals to date** - the cumulative totals of a measure for a specified time period. For example, you could calculate the cumulative total sales for a product from the beginning of the year to the current date.
  - **Average to date** - the cumulative average of a measure for a specified time period. For example, you could calculate the cumulative average sales for a product from the beginning of the year to the current date.
  - **Growth, comparing parallel periods** - a comparison of the value of a measure during parallel time periods. For example, you could compare product sales from Q1 2013 to product sales from Q1 2012.
  - **Growth, comparing parallel period totals to date** - a comparison of the totals during parallel time periods. For example, you could compare total product sales from January through the current month to the total product sales from January through the same month last year.
  - **Growth, comparing consecutive periods** - a comparison of the value of a measure for a time period versus the preceding period. For example, you could compare product sales from one quarter to the next.

- **Opening balance** - the value of a measure at the beginning of a time period. For example, this value could be the amount of inventory that you have for a product at the beginning of the month.
- **Closing balance** - the value of a measure at the end of a time period. For example, this value could be the amount of inventory that you have for a product at the end of the month.

5 Select the hierarchy to use in the calculation. You can also select the level of the hierarchy to use. Selecting a level is usually recommended. If you are calculating a distinct count, then select the measure to use in the calculation.

Note: If you are working with an SAP cube, then you must select a level.

Click **Next**.

6 From the **In** drop-down list, select the measure to use in the calculation.

Note: This option is available only if you selected a growth calculation.

7 From the **With measure** drop-down list, select the measure to use in the calculation.

8 Specify the time period of the data. For some calculations, you might need to specify the number of periods for the analysis. For example, you might want to specify how many time periods (such as months or years) that you want to include in a rolling calculation, or you might want to compare data between two periods (such as sales in 2007 compared with sales in 2006).

Click **Next**.

9 Select the format that you want to apply to the new calculated measure. If you are using a SQL server cube, then the formats in this list are similar to formats in Microsoft Excel. If you are using a SAS cube, then the formats in the list are SAS formats. You can also specify a custom format.

10 Specify how the new measure is available:

- **Locally, to me** - the calculated measure is valid for as long as you are using the OLAP cube in your current session. After the calculated measure has been created, it can be used by all queries in the same session. A calculated measure with this scope is available only to the person who created it.
- **Publicly, to all users at all times** - the calculated measure is available to anyone who connects to that cube. This option is available only on a SAS OLAP server. If you are not using a SAS OLAP server, then this option is disabled and the value is set to Locally, to me. You must also have WriteMetadata permission on the metadata server to use this option. If you do not have this permission, then an error message is displayed. For more information, contact your SAS administrator.

Note: If the calculation for the new measure includes a measure with local scope, then the scope option is disabled and the value is set to **Locally, to me**.

When a measure with local scope is used to create another calculated measure, the new calculated measure must be created with local scope as well.

Note: You cannot specify the availability of a calculated measure for an OLAP cube that is opened in a PivotTable.

11 In the **Solve order** box, enter the number that specifies the priority for evaluating the calculated member or calculated measure. A higher solve order results in the calculated member or calculated measure being evaluated later in the expression. The solve order usually is not needed, but it can be used when there is more than one calculated member or calculated measure in a query.

Click **Next**.

12 Review a summary of your selections and the MDX expression. Click **Finish** to create the new calculated measure.

Note: If you are working with an SAP cube, you cannot specify the format, availability, and solve order options. Any calculated measures that you create are available only to you.

Trends and Forecasting Analysis

You can use this type of calculated item to calculate the correlation or the covariance, or to perform a linear regression analysis.

Note: If your OLAP cube does not contain a time dimension or your provider does not support the statistics for trend and forecasting analysis, this analysis is not available.

To create a calculated member for trends and forecasting analysis:

1 If you are using the OLAP Analyzer, click ▦ in the Cube Manager. In the Customized Items and Sets view, click **Add Calculated Member**.

If you are working in a PivotTable, click **OLAP Options** on the **SAS** tab, and select **Calculated Members** ⇨ **New**.

The Add Calculated Member wizard opens.

2 In the **Name** box, enter the name of the new calculated measure.

3 For the type of analysis, select **Special analysis**, and from the drop-down list, select **Trends and Forecasting analysis**. Click **Next**.

4 Select the type of calculation for the new member:

- **Correlation** - statistic that measures the strength of the linear relationship between two series of values. The values of correlation coefficients range from –1 to 1.

- **Covariance** - the tendency of two measures to increase or decrease together at the same time.
- **Linear Regression Intercept** - the intercept of the line that is created by an attempt to assign a linear function to your data by using the least squares method.
- **Linear Regression Point** - a predicted value on the line that is created by an attempt to assign a linear function to your data by using the least squares method.
- **Linear Regression Slope** - the slope of the line that is created by an attempt to assign a linear function to your data by using the least squares method.
- **Linear Regression Variance** - the average of the total squared dispersion between each observation and the line that is created by an attempt to assign a linear function to your data by using the least squares method and the sample mean.

5 Select the measure or measures to use in the calculation.

6 Select the time hierarchy.

7 If you are calculating the linear regression point, you can also select the time period to use in the calculation. By default, the trend is calculated across all time periods.

8 Select the format that you want to apply to the new calculated measure. If you are using a SQL server cube, then the formats in this list are similar to formats in Microsoft Excel. If you are using a SAS cube, then the formats in the list are SAS formats. You can also specify a custom format.

9 Specify how the new measure is available:

- **Locally, to me** - the calculated measure is valid for as long as you are using the OLAP cube in your current session. After the calculated measure has been created, it can be used by all queries in the same session. A calculated measure with this scope is available only to the person who created it.
- **Publicly, to all users at all times** - the calculated measure is available to anyone who connects to that cube. This option is available only on a SAS OLAP server. If you are not using a SAS OLAP server, then this option is disabled and the value is set to Locally, to me. You must also have WriteMetadata permission on the metadata server to use this option. If you do not have this permission, then an error message is displayed. For more information, contact your SAS administrator.

Note: If the calculation for the new measure includes a measure with local scope, then the scope option is disabled and the value is set to **Locally, to me**. When a measure with local scope is used to create another calculated measure, the new calculated measure must be created with local scope as well.

Note: You cannot specify the availability of a calculated measure for an OLAP cube that is opened in a PivotTable.

10 In the **Solve order** box, enter the number that specifies the priority for evaluating the calculated member or calculated measure. A higher solve order results in the calculated member or calculated measure being evaluated later in the expression. The solve order usually is not needed, but it can be used when there is more than one calculated member or calculated measure in a query.

Click **Next**.

11 Review a summary of your selections and the MDX expression. Click **Finish** to create the new calculated measure.

Note: If you are working with an SAP cube, you cannot specify the format, availability, and solve order options. Any calculated measures that you create are available only to you.

Changing the Format of a Measure

You can change the format of measures that are displayed in the OLAP Analyzer. If you change the format of a globally calculated measure or a measure that was defined with the cube, a new measure is created with the format that you selected. The new measure is used to replace the existing measure in the view. If you change the format of a calculated measure that you have created, then the definition of the calculated measure is updated with the new format.

To change the format of a globally calculated measure or a measure that was defined with the cube:

1 In the table view, right-click the row or column heading of the measure that you want to reformat and select**Change Format To** ⇨ ***format-name***.

2 In the warning dialog box, click **Replace**. A new measure named *measure-name*_formatted is created with the format that you selected. The original measure is replaced in the view with the new measure.

To change the format of a calculated measure in the table view, right-click the row or column heading of the measure that you want to reformat and select **Edit *measure-name format-name***. The measure values are reformatted, and the format is saved with the calculated measure definition.

Creating Calculated Members

About Calculated Members

You can add a calculated member to a dimension in an OLAP cube. A calculated member is a member whose value is derived from the values of other members in the same dimension. A calculated member must be associated with a parent dimension. By default, the member is added to the top level of the parent dimension, which in most cases is the All level. However, you can specify the level to use.

Note: This option is available only for OLAP cubes that were created with SAS®9 or that were created for other OLE DB for OLAP providers that support calculated members.

Note: Calculated members that were created in SAS Information Map Studio are not supported in SAS Add-In for Microsoft Office.

Aggregate Across Members

To create a calculated member that is aggregated across members:

1 If you are using the OLAP Analyzer, click ▦ in the Cube Manager. In the Customized Items and Sets view, click **Add Calculated Member**.

If you are working in a PivotTable, click **OLAP Options** on the **SAS** tab, and select **Calculated Members** ⇨ **New**.

The Add Calculated Member wizard opens.

2 In the **Name** box, enter the name of the new calculated member.

3 For the type of analysis, select **Aggregate across members**.

4 Select the parent dimension. The member is added to the top member of the parent dimension, which in most cases is the All member. To change the member, click **All *member-name***.

The selected member appears in the Add Calculated Member wizard. Click **Next**.

5 If you are aggregating members for a parent, then from the **Select a member** pane, select a member. The children of that member appear in the **Check members to aggregate** pane.

If you are aggregating members on a level, then from the **Select a member** pane, select a level. The members of that level appear in the **Check members to aggregate** pane.

6 In the **Check members to aggregate** pane, select the check boxes for the members that you want to aggregate. Click **Next**.

7 Specify how the new measure is available:

- **Locally, to me** - the calculated measure is valid for as long as you are using the OLAP cube in your current session. After the calculated measure has been created, it can be used by all queries in the same session. A calculated measure with this scope is available only to the person who created it.

- **Publicly, to all users at all times** - the calculated measure is available to anyone who connects to that cube. This option is available only on a SAS OLAP server. If you are not using a SAS OLAP server, then this option is disabled and the value is set to Locally, to me. You must also have WriteMetadata permission on the metadata server to use this option. If you do not have this permission, then an error message is displayed. For more information, contact your SAS administrator.

Note: If the calculation for the new measure includes a measure with local scope, then the scope option is disabled and the value is set to **Locally, to me**. When a measure with local scope is used to create another calculated measure, the new calculated measure must be created with local scope as well.

Note: You cannot specify the availability of a calculated measure for an OLAP cube that is opened in a PivotTable.

8 In the **Solve order** box, enter the number that specifies the priority for evaluating the calculated member or calculated measure. A higher solve order results in the calculated member or calculated measure being evaluated later in the expression. The solve order usually is not needed, but it can be used when there is more than one calculated member or calculated measure in a query.

Click **Next**.

9 Review a summary of your selections and the MDX expression. Click **Finish** to create the new calculated measure.

Note: If you are working with an SAP cube, you cannot specify the format, availability, and solve order options. Any calculated measures that you create are available only to you.

Custom Calculation

To create a calculated member for a custom calculation:

1 If you are using the OLAP Analyzer, click 🖩 in the Cube Manager. In the Customized Items and Sets view, click **Add Calculated Member**.

If you are working in a PivotTable, click **OLAP Options** on the **SAS** tab, and select **Calculated Members** ⇨ **New**.

The Add Calculated Member wizard opens.

2 In the **Name** box, enter the name of the new calculated member.

3 For the type of analysis, select **Use MDX statement to create custom analysis**. Click **Next**.

4 From the **Parent hierarchy** drop-down list, select the parent dimension. The member is added to the top member of the parent dimension, which in most cases is the All member.

5 Specify the formula that you want to use for the new calculated member.

After you have created and verified your expression, click **Next**.

6 Specify how the new measure is available:

- **Locally, to me** - the calculated measure is valid for as long as you are using the OLAP cube in your current session. After the calculated measure has been created, it can be used by all queries in the same session. A calculated measure with this scope is available only to the person who created it.
- **Publicly, to all users at all times** - the calculated measure is available to anyone who connects to that cube. This option is available only on a SAS OLAP server. If you are not using a SAS OLAP server, then this option is disabled and the value is set to Locally, to me. You must also have WriteMetadata permission on the metadata server to use this option. If you do not have this permission, then an error message is displayed. For more information, contact your SAS administrator.

Note: If the calculation for the new measure includes a measure with local scope, then the scope option is disabled and the value is set to **Locally, to me**. When a measure with local scope is used to create another calculated measure, the new calculated measure must be created with local scope as well.

Note: You cannot specify the availability of a calculated measure for an OLAP cube that is opened in a PivotTable.

7 In the **Solve order** box, enter the number that specifies the priority for evaluating the calculated member or calculated measure. A higher solve order results in the calculated member or calculated measure being evaluated later in the expression. The solve order usually is not needed, but it can be used when there is more than one calculated member or calculated measure in a query.

Click **Next**.

8 Review a summary of your selections and the MDX expression. Click **Finish** to create the new calculated measure.

Note: If you are working with an SAP cube, you cannot specify the format, availability, and solve order options. Any calculated measures that you create are available only to you.

Editing Calculated Measures and Calculated Members

Edit a Calculated Measure or Calculated Member

Before you edit a calculated measure or calculated member, review these restrictions:

- You cannot add, edit, or delete a calculated measure or calculated member that is publicly available and is associated with an OLAP cube that is opened in a PivotTable.
- When you create a calculated measure or calculated member using the OLAP Analyzer, you must specify how the calculated item is available. Calculated items can be available locally (only to you) or globally (publicly available to everyone at your site). You cannot edit calculated items that are publicly available and were not created in the current session. Global calculated items that were created in the current session can be edited.

1. Click ▦ in the Cube Manager to open the Customized Items and Sets view.
2. In the Customized Items and Sets view, select the calculated measure or calculated member that you want to edit. Click **Edit**. The Add Calculated Measure wizard or the Add Calculated Member wizard appears.
3. Using the options in the wizard, modify the calculated measure or calculated member, and click **Finish** to save your changes.

Note: You cannot change the hierarchy for an existing calculated member, so when you edit a calculated member, the **Parent hierarchy** drop-down list is disabled.

Viewing the MDX Code for a Generated Calculated Measure or Calculated Member

When you create a calculated measure or calculated member, you must specify how the calculated item is available. Calculated items can be available locally (only to you) or globally (publicly available to everyone at your site). You cannot edit calculated items that are publicly available. However, you can view and copy the MDX code for these global calculated measures or calculated members.

To view the MDX code for a global calculated measure or calculated member:

1. Click ▦ in the Cube Manager to open the Customized Items and Sets view.

2 In the Customized Items and Sets view, select the calculated measure or calculated member and click **View**. The View MDX Statements dialog box appears.

Note: The **View** button is available only for calculated measures or calculated members that you cannot edit. If you can edit the calculated measure or calculated member, then the **Edit** button is available instead.

3 To copy the MDX code, click **Copy to Clipboard**.

Using the Advanced Expression Editor for a Calculated Measure or Calculated Member

You can use the Advanced Expression Editor to create the formula for a calculated measure or calculated member. To create an expression:

1 You can build your expression by using constant values, existing members, mathematical operators, and MDX functions. You can use the Advanced Expression Editor window to add these elements, or you can enter them in the expression box at the top of the Advanced Expression Editor window.

- To add operators, click the buttons below the expression box.
- To add an MDX function, select a category of functions from the **MDX Functions** list, and then select the specific function that you want to use.

  Note: The functions are categorized according to their return type. To view all of the functions in alphabetical order, select the All category.

- To add a specific dimension or level to the query, click the tab that corresponds to the dimension. Select the appropriate dimension or level and click to add it to your expression.
- To add metadata to the query, select the metadata item and click to add it to your expression. You can also double-click items in the MDX Functions and Metadata boxes to add the items to the expression.

2 Click **Verify** to validate your expression.

3 Click **OK**. If your MDX syntax is not valid, you will not be able to save the expression.

Setting Options for the MDX Editor

You can customize these options for the Enhanced Editor that is available in the MDX Editor.

Table 6.1 General Editor Options

Option Name	Description
Allow cursor movement past end of line	enables you to move the cursor past the end of the line in the program window. If you do not select this option, the cursor jumps down to the next line when you move it past a carriage return, and only the text on each line is highlighted during selection.
Drag and drop text editing	enables drag-and-drop editing to cut, copy, or paste text.
Show line numbers	displays line numbers in leftmost column of program windows. You cannot select line numbers.
Strip Ctrl+Z characters	removes any Ctrl-Z characters from the file. Ctrl-Z characters can be an end-of-file marker in some files.

Table 6.2 Changing the Options for Different File Types

Option Name	Description
File type	displays the type of file whose general settings you are editing. To change the settings of another file type, select the file type from the **File type** drop-down list.
Tab settings	■ **Tab size** - 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. ■ **Insert spaces for tabs** - inserts the number of spaces listed in the **Tab size** box instead of a single tab character. ■ **Replaces tabs with spaces on file open** - automatically replaces all tab characters with the number of spaces listed in the **Tab size** box when you open the file.
Indention	specifies the indention for a new line. ■ **None** - turns off all automatic indention. ■ **Automatic** - automatically indents a new line by the amount of space that the previous line is indented.

Creating and Using Member Sets

In the OLAP Analyzer, you can create member sets, which are groups of members from the same dimension. You can use member sets in the Data Dimensions view to add multiple members to the rows and columns at one time.

To create a new member set:

1. Click ▦ in the Cube Manager to open the Customized Items and Sets view.
2. Select **Add** ⇨ **Member Set**.
3. Specify a name for the member set.
4. Select the dimension from which you want to select members.
5. Use the **Browse**, **Search**, or **Use Filter** tabs to select the members that you want to include in the member set.
6. Click **OK** to create the member set.

Creating and Using Percent of Visual Total Items

In the OLAP Analyzer, you can create another type of measure that computes the percent that each value is of a visual row, column, or grand total. Visual totals display only the total of the values that are displayed in the cube view. In order to create a percent of visual total item, you must display visual totals in the cube view.

Note: Percent of visual total items cannot be used in conditional highlights, calculated measures or members, or as filter criteria.

To create a new percent of visual total item:

1. Click ▦ in the Cube Manager to open the Customized Items and Sets view.
2. Select **New** ⇨ **Percent of Visual Total**.
3. From the **Percent of** drop-down list, select the total that you want to use.
   - **Grand Total** - calculates the percent that each value is of the grand total of all row and column totals
   - **Row Total** - calculates the percent that each value is of the total for the row.
   - **Row Subtotals** - calculates the percent that each value is of the subtotal for each row. The row subtotal is the row total for each level.
   - **Column Total** - calculates the percent that each value is of the total for the column.

- **Column Subtotals** - calculates the percent that each value is of the subtotal for each column. The column subtotal is the column total for each level.

4 Select the measure that you want to use in the calculation.

5 Enter the name of the percent of total item. The name is displayed in the list of items in the Customized Items and Sets view.

6 Click **OK** to create the percent of total item.

Note: If the cube totals that are displayed are not visual totals, then you are prompted to set the totals to visual totals. If you do not set the totals to visual totals, then the percent of total item is created, but it cannot be added to the view until the totals are set to visual totals.

Adjusting Values in a Write-Enabled Cube

If you are using an OLAP server that supports write-enabled cubes, such as Microsoft SQL Server Analysis Services, then you can use the OLAP Analyzer to adjust values in the cube. You might want to adjust values in a cube in order to conduct a what-if analysis and determine how a change to one value might affect other values in the cube. For example, in a cube that contains budget data, you could adjust the budgeted values for personnel expenditures to see how total profit is affected.

When you adjust a value in a cube, the OLAP Analyzer automatically makes proportional adjustments to all of the other leaf values in the hierarchy. The adjusted values are stored in a separate file that is saved with your project and automatically applied to the cube each time you open it in that project. The original values in the cube are not updated. If you want to view the cube in its original form again, you can clear the cube changes or reopen the cube in your project.

You can adjust a value either by specifying a new value or by specifying the percentage by which you want to increase or decrease the current value.

Note: Depending on the size of your cube, the process of adjusting all of the leaf values might take a few minutes. You cannot stop the process once it has started.

To adjust a value in a cube in the table view, right-click the value that you want to change and click **Adjust Value**. Select one of the following options:

- **Enter Value** - enables you to specify a new value in the cube. The percentage of change is calculated and applied to all of the leaf values in the hierarchy.

  Note: You can also specify a new value by double-clicking the value that you want to change and then entering the new value.

- **Increase Value by 10%** - increases the current value by 10%. All of the leaf values in the hierarchy are also increased by 10%.

- **Increase Value by 5%** - increases the current value by 5%. All of the leaf values in the hierarchy are also increased by 5%.
- **Decrease Value by 10%** - decreases the current value by 10%. All of the leaf values in the hierarchy are also decreased by 10%.
- **Decrease Value by 5%** - decreases the current value by 5%. All of the leaf values in the hierarchy are also decreased by 5%.
- **Change Value by Other Percent** - opens the Adjust Value By Percent dialog box so that you can specify the percentage by which to adjust the value. Select whether you want to increase or decrease the values, and then specify the percentage. All of the leaf values in the hierarchy are also adjusted by the same percentage.

Note: To view a list of all the changes that you have made to the cube, click in the Cube Manager.

To clear the cube changes and return to the original cube values:

1. Click in the Cube Manager to open the Cube Change History view.
2. Click **Clear All**, and then click **OK** in the confirmation dialog box.

Viewing Member Properties

You can use the OLAP Analyzer to view member properties that might be stored with your cube. Member properties are additional attributes that can be associated with dimension members. For example, in a cube that stores sales data for retail stores, you might also have member properties that include each store's size in square feet, the store address, and the current manager.

You can view one or more properties at a time, and you can choose to display member properties instead of the member caption.

To view member properties:

1. In the table view, right-click on the level heading for which you want to view member properties and select **Show Member Property**.
2. In the **Level** box, select the level from which you want to select member properties.
3. In the **Member properties for *level-name*** box, select one or more member properties that you want to include in the table view.
4. If you want to turn off the display of the property name and show only the property value, clear the **Show property name for *level-name*** check box.
5. If you want to turn off the display of the member captions and show only the member properties, clear the **Show member caption for *level-name*** check box. You can clear the **Show member caption for *level-name*** check box only if you have selected at least one member property to display

6 Click **OK** to display the member properties in the table view.

Edit an Existing Cube View

The OLAP Analyzer displays the OLAP cube with a default table view. You can use the View Editor to edit any existing table view by adding and removing members from the columns and rows. You can also rearrange the order of the columns and rows. When you edit a view, no changes are made to the table until you save them.

You can also delete existing views.

To edit an existing view:

1 Right-click on the table view and select **Table View** ⇨ **Edit View** ⇨ **Edit with View Editor**. The Edit window appears.

2 Click the **Browse** tab if it is not already selected, and expand the tree as necessary to view the different levels in a dimension. By default, the dimensions are displayed in a parent-child hierarchy.

> **TIP** The tree view displays all of the items that are included in the current query in bold.

> **TIP** To view the levels inside the parent-child hierarchy, click **Show Levels**. The hierarchical tree view is replaced with a view that displays all of the levels for a dimension without regard for their parent-child relationship. You can use the **Show Levels** option to view lower-level members that might otherwise be hidden if their parents are hidden due to security restrictions.

3 To add a member to the columns or rows, select the member that you want to add. You can drag the member to the list of columns or rows, or you can choose to add a member as follows:

- To add only the selected item to the list of columns or rows, click the arrow beside the **Columns** or **Rows** area.
- To add the selected item and all of its descendants, click the inverted triangle on the arrow button and select **Selected and all descendants**.

  Note: This option is relevant only if the selected item is a member rather than a dimension.

4 To rearrange the order of the columns or rows, select the member that you want to move up or down and click the appropriate or button

5 Click **OK**.

Using the MDX Editor

By default, the query that the OLAP Analyzer has created for the current view of the OLAP cube is displayed in the MDX Editor. You can either edit that query, or clear the query and write your own.

For details about MDX syntax, refer to the *Microsoft Data Access Components Software Developer's Kit*.

To use the MDX Editor:

1 Right-click on the table view and select **Edit View** ⇨ **Edit with MDX Editor**.

2 Select the members and functions that you want to use in your query.

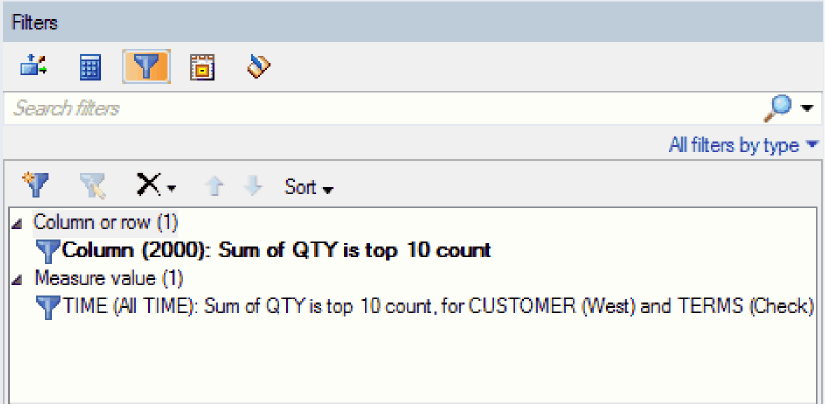
3 When you are ready to run your query, click **OK**.

Filtering and Sorting Data in an OLAP Cube

About Filtering Data in an OLAP Cube

When you view OLAP data in the OLAP Analyzer, you might want to retrieve only data that meets certain criteria, based on values in the data. The process of telling the OLAP Analyzer which data to retrieve is called creating a filter and is performed in the Filters view of the Cube Manager.

When you apply a filter to your data, the filter can usually remain applied while you expand, collapse, or drill in your data. However, if you change what is displayed in the table or graph views so that the filter can no longer be applied, then the filter is automatically removed from the view.



Column or row (1)

Column (2000): Sum of QTY is top 10 count

Measure value (1)

TIME (All TIME): Sum of QTY is top 10 count, for CUSTOMER (West) and TERMS (Check)

Two Types of Filters: Rank Values and Range of Values

You can filter your OLAP data by specifying criteria to limit the values in a column that is currently displayed in the table. You can filter the data by ranking the measure values of the column or by specifying a range of the measure values to view.

You can create two types of filters:

- **Rank values** - filters the data by ranking measure values.
- **Range of values** - filters the data by specifying a range of measure values to view.

When you create a rank filter, you can choose whether you want to view the top or bottom rankings, and you can choose to rank the data by count, by percentage of count or total value, or by sum. For example, you could choose to filter a column of profit data to display only the top 10 profits in that column.

When you create a range filter, you can specify the range of values by selecting a comparison operator and the value that you want to be compared in the filter. For example, you could choose to filter a column of profit data to display only values that are below $20,000.

Create a Rank Values Filter for a Column

To create a column filter based on rank:

1. Click in the Cube Manager to open the Filters view.
2. Click in the Filters view. The New Filter dialog box appears.

3 For the **Limit members on column** option, select the column on which you want the filter to be based. By default, the first column in the table is selected. To select another column, click the column name.

4 For the **Based on** option, select **Rank**.

5 To create the filter, you must specify the measure that you want to use. This option is available only if there are measures displayed in the columns of the table view.

6 Select whether you want the top (largest) or bottom (smallest) values returned in the ranking. By default, the top values are returned.

7 Specify the number of values that you want to return in the ranking. The default value is `10`.

8 Specify how you want the rank to be determined:

- **Count** - returns the top or bottom *n* members, where **n** is the numeric value that you specified. For example, if you specified **Bottom** and a numeric value of `5`, then the filter would return the bottom, or smallest, five members.
- **Percent by Count** - returns the top or bottom *n* percent of members, where *n* is the numeric value that you specified. For example, if you specified **Top** and a numeric value of `10`, then the filter would return the top, or largest, 10% of the members. If there were 500 members in the set, the top 50 members would be returned.
- **Percent by Value** - returns the top or bottom *n* percent of the sum of the values of the selected measure, where *n* is the numeric value that you specified. For example, if you specified **Top** and a numeric value of `90`, then the sum of the members that are returned would equal at least 90% of the total sum of all of the measure values. If the member values that are being filtered include 6, 3, and 1, which add up to 10, then the members that are returned would be 6 and 3, which add up to 9, or 90% of 10.
- **Sum** - returns the top or bottom members until their sum is at least the numeric value that you specified. For example, if you specified **Top**, a numeric value of `10`, and the member values to filter include 5, 3, 2, and 1, then the members that are returned would be 5, 3, and 2 because their sum is 10.

9 Specify whether you want to include ties and missing values in your filter results. By default, ties are included. Click the ties and missing values option to display the check boxes.

- **Include ties** - includes multiple members that meet the criteria for the last member to return in your filter results. If you want to include only one arbitrarily chosen last member, then clear the **Include ties** check box.
- **Include missing values** - includes missing values in the values that are ranked. This option is not selected by default.

10 Specify a name for the filter. By default, the filter name is **Column (columnname)**.

11 Click **OK** to create the new filter. The new filter is applied to the cube and added to the list of filters in the Filters view.

Create a Range of Values Filter for a Column

To create a column filter based on range:

1. Click in the Cube Manager to open the Filters view.
2. Click in the Filters view. The New Filter dialog box appears.
3. From the **Select a type of filter** box, select **Filter values on a column in the table**.
4. For the **Limit members on column** option, select the column on which you want the filter to be based. By default, the first column in the table is selected. To select another column, click the column name.
5. For the **Based on** option, select **Range**.
6. To create the filter, you must specify the measure that you want to use. This option is available only if there are measures displayed in the columns of the table view.
7. Select the comparison operator that you want to use. The default value is **Is greater than**. To select another operator, click the currently selected operator, and then make a selection from the box. If you want to negate the comparison operator, click the currently selected operator, and then select **Not** and the comparison operator that you want to use.

   > **TIP** You can create a filter to display only measure data with missing values by using the **Is missing** operator. The **Is missing** operator returns all data in which the value is missing.

8. Specify the value that you want to be compared in the filter. The default value is `0`. You can also select another measure as the comparison value by clicking the currently selected comparison value, and then selecting a measure from the **Enter a value** drop-down list.
9. Specify the name of the filter. By default, the filter name is **Column (*columnname*)**.
10. Click **OK**. The new filter is applied to the cube and added to the list of filters in the Filters view.

Create a Rank Values Filter for a Row

You can filter your OLAP data by specifying criteria to limit the values in a row that is currently displayed in the table. You can filter the data by ranking the measure values of the row or by specifying a range of the measure values to view.

You can create two types of filters:

- **Rank values** - filters the data by ranking measure values.
- **Range of values** - filters the data by specifying a range of measure values to view.

When you create a rank filter, you can choose whether you want to view the top or bottom rankings, and you can choose to rank the data by count, by percentage of count or total value, or by sum. For example, you could choose to filter a row of sales data to display only the columns of regions in the top 10%.

When you create a range filter, you can specify the range of values by selecting a comparison operator and the value that you want to be compared in the filter. For example, you could choose to filter a row of sales data to display only columns of regions with sales over $25,000.

To create a column filter based on rank:

1. Click in the Cube Manager to open the Filters view.
2. Click in the Filters view. The New Filter dialog box appears.
3. From the **Select a type of filter** box, select **Filter values on a row in the table**.
4. For the **Limit members on row**option, select the row on which you want the filter to be based. By default, the first row in the table is selected. To select another row, click the column name.
5. For the **Based on** option, select **Rank**.
6. To create the filter, you must specify the measure that you want to use. This option is available only if there are measures displayed in the rows of the table view.
7. Select whether you want the top (largest) or bottom (smallest) values returned in the ranking. By default, the top values are returned.
8. Specify the number of values that you want to return in the ranking. The default value is `10`.
9. Specify how you want the rank to be determined:
   - **Count** - returns the top or bottom *n* members, where **n** is the numeric value that you specified. For example, if you specified **Bottom** and a numeric value of `5`, then the filter would return the bottom, or smallest, five members.
   - **Percent by Count** - returns the top or bottom *n* percent of members, where *n* is the numeric value that you specified. For example, if you specified **Top** and

a numeric value of `10`, then the filter would return the top, or largest, 10% of the members. If there were 500 members in the set, the top 50 members would be returned.

- **Percent by Value** - returns the top or bottom *n* percent of the sum of the values of the selected measure, where *n* is the numeric value that you specified. For example, if you specified **Top** and a numeric value of `90`, then the sum of the members that are returned would equal at least 90% of the total sum of all of the measure values. If the member values that are being filtered include 6, 3, and 1, which add up to 10, then the members that are returned would be 6 and 3, which add up to 9, or 90% of 10.
- **Sum** - returns the top or bottom members until their sum is at least the numeric value that you specified. For example, if you specified **Top**, a numeric value of `10`, and the member values to filter include 5, 3, 2, and 1, then the members that are returned would be 5, 3, and 2 because their sum is 10.

10 Specify whether you want to include ties and missing values in your filter results. By default, ties are included. Click the ties and missing values option to display the check boxes.

- **Include ties** - includes multiple members that meet the criteria for the last member to return in your filter results. If you want to include only one arbitrarily chosen last member, then clear the **Include ties** check box.
- **Include missing values** - includes missing values in the values that are ranked. This option is not selected by default.

11 Specify a name for the filter. By default, the filter name is **Row (*row-name*)**.

12 Click **OK** to create the new filter. The new filter is applied to the cube and added to the list of filters in the Filters view.

Filtering Members of a Hierarchy Based on Their Values

You can filter your OLAP data by specifying criteria to limit the members of a data hierarchy based on measure values, regardless of whether the values that are being used as filtering criteria are displayed in the table view. For example, in a table view that displays sales data by quarter for each sales region, you could create a filter to view the top three sales regions for the entire year without displaying the annual total in the table.

You can create two types of filters:

- **Rank values** - filters the data by ranking measure values.
- **Range of values** - filters the data by specifying a range of measure values to view.

When you create a rank filter, you can choose whether you want to view the top or bottom rankings, and you can choose to rank the data by count, by percentage of count or total value, or by sum. For example, you could choose to filter the Beverage members of the Products hierarchy to display only the beverages with a total profit in the bottom 10%.

When you create a range filter, you can specify the range of values by selecting a comparison operator and the value that you want to be compared in the filter. For example, you could choose to filter the Beverage members of the Products hierarchy to display only the beverages with an annual marketing expense less than $30,000.

To create a hierarchy values filter based on rank:

1 Click in the Cube Manager to open the Filters view.

2 Click in the Filters view. The New Filter dialog box appears.

3 From the **Select a type of filter** box, select **Filter members of a hierarchy based on their values**.

4 For the **Hierarchy** option, select the hierarchy whose members you want to filter.

   Note: If you select a hierarchy that is not currently displayed in the table view, then you cannot apply the filter until the hierarchy is added to the table view.

5 The default value in the **Limit members of** drop-down list is either the parent member if all the members in the query share the same parent or the highest level of all the members in the query if the members do not share the same parent. To browse for the member or level that is associated with the member property that you want to use, click the member name to open the Member or Level dialog box.

   - To browse for a specific member, select **A member** from the **Select** drop-down list.
   - To browse for a specific level, select **A level** from the **Select** drop-down list.

   Note: If you choose to filter on a specific member, you are filtering on the children of the selected member. If you choose to filter on a specific level, you are filtering on all of the members of the selected level.

6 For the **Based on** option, select **Rank**.

7 Select the measure whose values you want to be ranked. By default, the measure that is displayed first in the table is selected. To select another measure, click the measure name. The measure that you select does not have to be displayed in the table view.

8 Select whether you want the top (largest) or bottom (smallest) values returned in the ranking. By default, the top values are returned.

9 Specify the number of values that you want to return in the ranking. The default value is `10`.

10 Specify how you want the rank to be determined:

- **Count** - returns the top or bottom *n* members, where **n** is the numeric value that you specified. For example, if you specified **Bottom** and a numeric value of `5`, then the filter would return the bottom, or smallest, five members.

- **Percent by Count** - returns the top or bottom *n* percent of members, where *n* is the numeric value that you specified. For example, if you specified **Top** and a numeric value of `10`, then the filter would return the top, or largest, 10% of the members. If there were 500 members in the set, the top 50 members would be returned.

- **Percent by Value** - returns the top or bottom *n* percent of the sum of the values of the selected measure, where *n* is the numeric value that you specified. For example, if you specified **Top** and a numeric value of `90`, then the sum of the members that are returned would equal at least 90% of the total sum of all of the measure values. If the member values that are being filtered include 6, 3, and 1, which add up to 10, then the members that are returned would be 6 and 3, which add up to 9, or 90% of 10.

- **Sum** - returns the top or bottom members until their sum is at least the numeric value that you specified. For example, if you specified **Top**, a numeric value of `10`, and the member values to filter include 5, 3, 2, and 1, then the members that are returned would be 5, 3, and 2 because their sum is 10.

11 Specify whether you want to include ties and missing values in your filter results. By default, ties are included. Click the ties and missing values option to display the check boxes.

- **Include ties** - includes multiple members that meet the criteria for the last member to return in your filter results. If you want to include only one arbitrarily chosen last member, then clear the **Include ties** check box.

- **Include missing values** - includes missing values in the values that are ranked. This option is not selected by default.

12 For the **Apply filter to** option, select one or more other dimensions to filter the selected dimension. For example, you could limit your filter of top sales to the year 2017 by setting the **Apply To** option to the member [2017] of the Time dimension. To browse for dimensions, click the dimension name to open the Apply To dialog box.

- To apply the filter to the default members of all of the other dimensions, select **The default members of all other dimensions** and click **OK**.

- To apply the filter to members that you select from other dimensions, select **Selected members from dimensions**, and then select the dimension that you want to use from the **Dimension** drop-down list. To specify the individual members that you want to use, select **Specify members**. To specify a filter to use instead of individual members, select **Use filter**. Only filters that are associated with the dimension that you have selected are displayed. The filter that you select is evaluated when the range filter is evaluated, and the members that are returned by that filter are used as the selected members. The filter or members that you select are displayed in the **Selected** area.

13 Specify a name for the filter. By default, the filter name is **Hierarchy (*membername*)**.

14 Click **OK** to create the new filter.

To create a hierarchy values filter based on range:

1 Click in the Cube Manager to open the Filters view.

2 Click in the Filters view. The New Filter dialog box appears.

3 From the **Select a type of filter** box, select **Filter members of a hierarchy based on their values**.

4 For the **Hierarchy** option, select the hierarchy whose members you want to filter.

Note: If you select a hierarchy that is not currently displayed in the table view, then you cannot apply the filter until the hierarchy is added to the table view.

5 For the **Limit members of** option, select the member on which you want the filter to be based. By default, the [ALL] member is selected. To select another member, click the member name.

6 For the **Based on** option, select **Range**.

7 Select the measure whose values you want to be compared in the filter. By default, the measure that is displayed first in the table is selected. To select another measure, click the measure name and select the measure that you want to use. The measure that you select does not have to be displayed in the table view.

8 Select the comparison operator that you want to use. The default value is **Is equal to**. To select another operator, click the currently selected operator, and then make a selection from the box. If you want to negate the comparison operator, click the currently selected operator, and then select **Not** and the operator that you want to use.

TIP You can create a filter to display only measure data with missing values by using the **Is missing** operator. The **Is missing** operator returns all data in which the value is missing.

9 Specify the value that you want to be compared in the filter. The default value is 0. You can also select another measure as the comparison value by clicking the comparison value and selecting a measure from the **Enter a value** drop-down list.

10 For the **Apply filter to** option, select one or more other dimensions to filter the selected dimension. For example, you could limit your filter of top sales to the year 2017 by setting the **Apply To** option to the member [2017] of the Time dimension. To browse for dimensions, click the dimension name to open the Apply To dialog box.

- To apply the filter to the default members of all of the other dimensions, select **The default members of all other dimensions** and click **OK**.

- To apply the filter to members that you select from other dimensions, select **Selected members from dimensions**, and then select the dimension that you want to use from the **Dimension** drop-down list. To specify the individual members that you want to use, select **Specify members**. To specify a filter to use instead of individual members, select **Use filter**. Only filters that are associated with the dimension that you have selected are displayed. The filter that you select is evaluated when the range filter is evaluated, and the members that are returned by that filter are used as the selected members. The filter or members that you select are displayed in the **Selected** area.

11 Specify a name for the filter. By default, the filter name is **Hierarchy (*membername*)**.

12 Click **OK** to create the new filter.

Filtering Members of a Hierarchy Based on Their Labels

You can filter your OLAP data by specifying criteria to limit the members of a data hierarchy based on the labels that are displayed in the table headings. For example, in a table view that displays sales data by quarter, you could create a filter to view only sales in the East region.

To create a hierarchy values filter based on labels:

1 Click in the Cube Manager to open the Filters view.

2 Click in the Filters view. The New Filter dialog box appears.

3 From the **Select a type of filter** box, select **Filter members of a hierarchy based on their labels**.

4 For the **Hierarchy** option, select the hierarchy whose members you want to filter.

> **Note:** If you select a hierarchy that is not currently displayed in the table view, then you cannot apply the filter until the hierarchy is added to the table view.

5 The default value in the **Limit members of** drop-down list is either the parent member if all the members in the query share the same parent or the highest level of all the members in the query if the members do not share the same parent. To browse for the member or level that is associated with the member property that you want to use, click the member name to open the Member or Level dialog box.

- To browse for a specific member, select **A member** from the **Select** drop-down list.
- To browse for a specific level, select **A level** from the **Select** drop-down list.

Note: If you choose to filter on a specific member, you are filtering on the children of the selected member. If you choose to filter on a specific level, you are filtering on all of the members of the selected level.

6 For the operator option, select the comparison operator that you want to use. The default value is **Contains**. If you want to negate the comparison operator, click the operator to open the operator dialog box. Select the **Not** check box and the operator that you want to use.

> **TIP** You can create a filter to display only labels that are blank by using the **Is blank** operator. The **Is blank** operator returns all data that is associated with a blank label value.

7 Click the **Enter a value** option to enter or select the value that you want to be compared in the filter. If you select **Exactly matches** as the operator, click ... to select the value in the Select Values dialog box.

8 Specify a name for the filter. By default, the filter name is **Hierarchy (*membername*)**.

9 Click **OK** to create the new filter.

Filtering Members of a Hierarchy Based on Their Properties

You can filter your OLAP data by specifying criteria to limit the members of a data hierarchy based on their property labels. For example, suppose you have a cube that contains sales data for retail stores. The cube also has a member property that indicates whether a store has street-front access. You could create a filter to display only the stores without street-front access.

To create a hierarchy values filter based on properties:

1 Click ▼ in the Cube Manager to open the Filters view.

2 Click ✳▼ in the Filters view. The New Filter dialog box appears.

3 From the **Select a type of filter** box, select **Filter members of a hierarchy based on their properties**.

4 For the **Hierarchy** option, select the hierarchy whose members you want to filter.

Note: If you select a hierarchy that is not currently displayed in the table view, then you cannot apply the filter until the hierarchy is added to the table view.

5. The default value in the **Limit members of** drop-down list is either the parent member if all the members in the query share the same parent or the highest level of all the members in the query if the members do not share the same parent. To browse for the member or level that is associated with the member property that you want to use, click the member name to open the Member or Level dialog box.

   - To browse for a specific member, select **A member** from the **Select** drop-down list.
   - To browse for a specific level, select **A level** from the **Select** drop-down list.

   Note: If you choose to filter on a specific member, you are filtering on the children of the selected member. If you choose to filter on a specific level, you are filtering on all of the members of the selected level.

6. For the member property option, select the member property that you want to use in the filter.

7. For the operator option, select the comparison operator that you want to use. The default value is **Contains**. If you want to negate the comparison operator, click the operator to open the operator dialog box. Select the **Not** check box and the operator that you want to use.

   > **TIP** You can create a filter to display only labels that are blank by using the **Is blank** operator. The **Is blank** operator returns all data that is associated with a blank label value.

8. Click the **Enter a value** option to enter or select the value that you want to be compared in the filter.

9. Specify a name for the filter. By default, the filter name is **Hierarchy (*membername*)**.

10. Click **OK** to create the new filter.

Filtering Time Periods of a Time Hierarchy

You can filter your OLAP data by specifying criteria to limit the members of a time hierarchy to a certain time period. For example, you could create a filter to display only the first six months of the year 2018.

To create a time hierarchy filter:

1. Click in the Cube Manager to open the Filters view.

2. Click in the Filters view. The New Filter dialog box appears.

3. From the **Select a type of filter** box, select **Filter time periods of a time hierarchy**.

4 For the **Hierarchy** option, select the hierarchy whose members you want to filter.

Note: If you select a hierarchy that is not currently displayed in the table view, then you cannot apply the filter until the hierarchy is added to the table view.

5 The default value in the **Limit members of** drop-down list is either the parent member if all the members in the query share the same parent or the highest level of all the members in the query if the members do not share the same parent. To browse for the member or level that is associated with the member property that you want to use, click the member name to open the Member or Level dialog box.

- To browse for a specific member, select **A member** from the **Select** drop-down list.
- To browse for a specific level, select **A level** from the **Select** drop-down list.

Note: If you choose to filter on a specific member, you are filtering on the children of the selected member. If you choose to filter on a specific level, you are filtering on all of the members of the selected level.

6 Select the time period that you want to use to limit the data.

7 For the **Starting** option, specify the starting time period for the filter.

8 For the **Ending** option, specify the ending time period for the filter.

9 Specify a name for the filter. By default, the filter name is **Hierarchy (*membername*)**.

10 Click **OK** to create the new filter.

Sorting Data in an OLAP Cube

You can sort the rows in the table view of the OLAP Analyzer by measure value, by member caption, or by member property. The column or row on which the data is sorted is identified by an upward triangle an ascending sort or an inverted triangle for a descending sort in the column heading.

Note: The sorting functionality is available only when you access the OLAP cube by using an OLE DB for OLAP Provider.

To sort by using the table view, right-click on the heading in the table view on which you want to sort your data, and select **Sort**.

Note: The options to sort within or across a hierarchy are disabled when the totals that are displayed in the cube are visual totals.

Note: The sort is performed on the measure and is evaluated using the default member of the dimension on the column, even if the default member is not what is displayed in the table view.

Viewing an ESRI Map

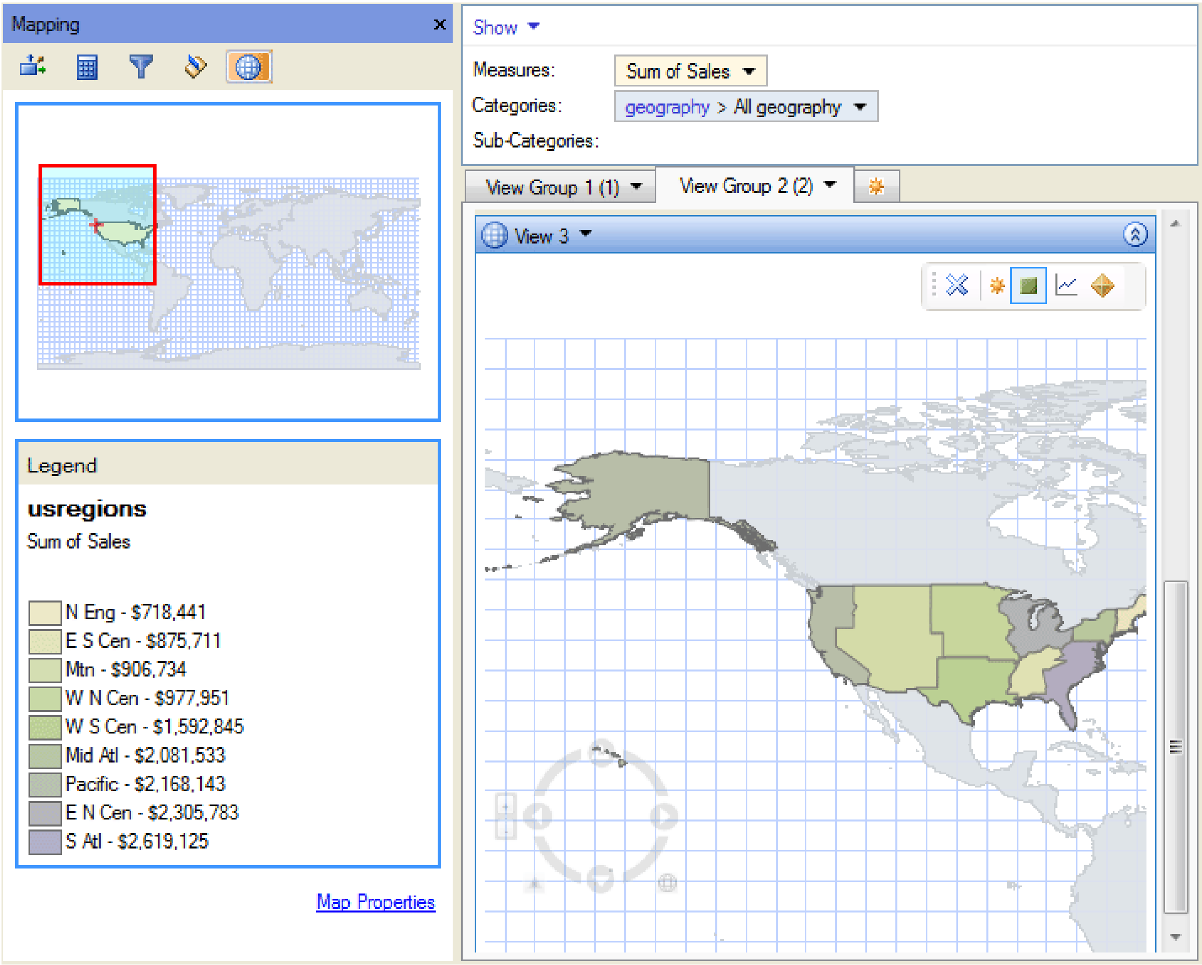
About ESRI Maps

Multidimensional data often contains a geographic dimension that allows analysis based on varying locations. In many cases, these locations are political boundaries such as country, state, province, or city. In other cases, the locations, or regions, are defined based on business rules such as sales territory, wireless coverage plans, or population-based designated market areas.

If ArcGIS software has been installed on your computer, you can view your data in the context of ESRI maps. In an ESRI map, you can drill down and navigate through cube data, display data, set options, filter data, and export the map as an image.

Note: ArcGIS Engine Run-time is required to work with ESRI maps. If you cannot open an ESRI map, contact your site administrator.

When you select an OLAP cube that has been associated with an ESRI map, the OLAP Analyzer displays the map as an additional view of the data.



Measures: Sum of Sales
Categories: geography > All geography
Sub-Categories:

usregions
Sum of Sales

■ N Eng - $718,441
■ E S Cen - $875,711
■ Mtn - $906,734
■ W N Cen - $977,951
■ W S Cen - $1,592,845
■ Mid Atl - $2,081,533
■ Pacific - $2,168,143
■ E N Cen - $2,305,783
■ S Atl - $2,619,125

Open an ESRI Map

If an ESRI map has been associated with the data in an OLAP cube, you can display the map after you open the cube.

To open an ESRI map:

1 Open the OLAP cube that is associated with the ESRI map that you want to view.

2 In the **OLAP Analyzer** group on the **SAS** tab in the Ribbon, select **Insert View** ⇨ **ESRI Map**. The Choose Location dialog box appears.

3 Specify the location for the new view and click **OK.** A new view with the ESRI map is created.

Displaying Data in an ESRI Map

When you have opened an OLAP cube that is associated with an ESRI map, you can navigate through the table view in the same way as with any other OLAP cube. You can use the map view to select specific geographic locations that you want to analyze. You can select locations on the map by using the map legend on the left or by using the drawing tools in the toolbar. After you have selected a location, you can choose from many options, including drilling, expanding, or isolating the location as well as displaying data about the location.

You can select a location on the map either by using the **Legend** area or by using the drawing tools. Any locations on the map that are partially or wholly included in the area that you have drawn are selected in the legend and highlighted in the map view.

To isolate an area in the map view:

1. Select an area on the map view.
2. Right-click the area of the map that you have selected and select **Keep Only *level-name* in *hierarchy-name***. Only the selected level is displayed in the map.

To identify a particular location and display the data that is associated with it, right-click a location on the map and select **Identify**.

Filter ESRI Map Data

You can use an ESRI map to filter your data. For example, you might choose to display only the data that pertains to a particular city.

To filter ESRI map data:

1. Select an area on the map view.
2. Right-click the area of the map that you have selected and select **Filter**. From the pop-up menu, select the filter that you want to use.
3. Click **OK** to create the filter.

Export an ESRI Map

You can export an ESRI map as an image file.

To export an ESRI map:

1. Right-click the ESRI map that you want to export and select **Map View** ⇨ **Export Image**. The Save As dialog box appears
2. Select a file location and specify a filename.
3. Select an image type from the **Save as type** drop-down list. You can save the map as a BMP, JPEG, or PNG image
4. Click **Save**.

Setting ESRI Map Options

Specifying Visibility Options

To set ESRI map options, right-click the map and select **Map Properties** to display the Options window. Click the **Display** tab.

- **Show map legend** - displays the name of the selected area and its associated data along the right side of the map.
- **Show feature tips** - displays the name of the location on the map and its associated data as you move the mouse pointer over the map.
- **Resize on navigate** - resizes the map to zoom in on the current level of data as you navigate in the map. If you do not select this option, then the map does not automatically zoom to display the current level of data and instead remains at the last zoom level. This option is selected by default.

Specifying Layers and Viewing the Location of the Map Document

To set ESRI map options, right-click the map and select **Map Properties** to display the Options window. Click the **General** tab.

- **Map Layers** - enables you to select the map layers that you want to display. Clear the check boxes for the layers that you want to conceal from view.
- **Location** - displays the location of the map document.

Setting the Symbology Options

To set ESRI map options, right-click the map and select **Map Properties** to display the Options window. Click the **Symbology** tab.

Setting the color theme

Theme - enables you to select a theme. The theme colors are displayed in the map and in the scale of values in the Cube View Manager.

Setting the classification method

- **Method** - specifies the classification method for displaying map data.
- **Classes** - specifies the number of classes into which the data is divided.

Setting patterns for missing values

- **Style** - specifies a pattern that is displayed for any missing values.
- **Foreground** - specifies the color and hue for the foreground.
- **Background** - specifies the color and hue for the background.

Customizing the OLAP Analyzer

Working with Conditional Highlighting

Creating Conditional Highlights

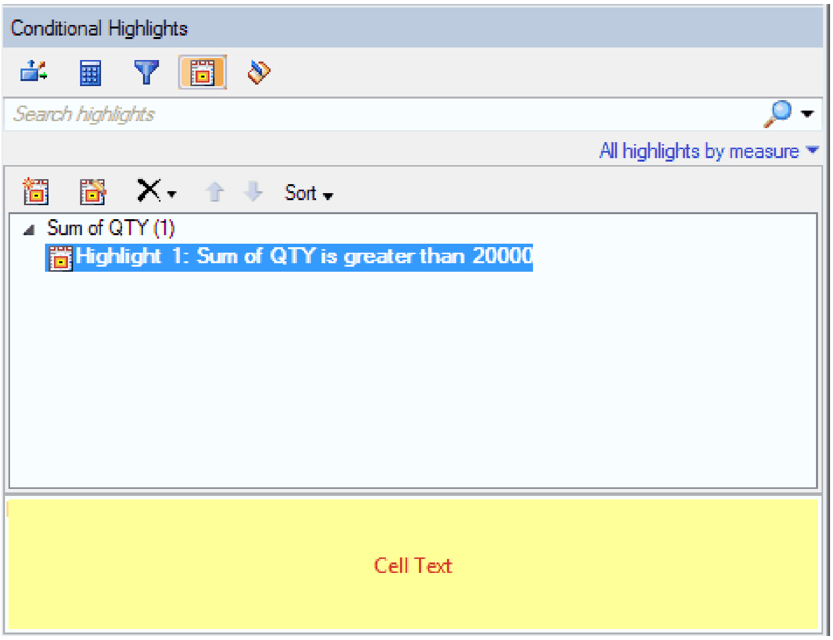
You can create a conditional highlight in an OLAP cube by applying special formatting to values that meet conditions that you specify. For example, you could display sales figures that are less than or equal to a specified value in red with a yellow background, and display sales figures above that value in black. Any conditional highlights that you create are saved and can be used again each time you open the OLAP cube.

When you create a conditional highlight, you can set the condition under which you want to highlight exceptions, and you can define the formatting that you want to apply when the condition is met.

Note: Row and column totals are not included in a conditional highlight.

If you create more than one conditional highlight, they are applied in the order in which you create them.

When you create conditional highlights, they are displayed in the Highlights section of the View Manager and in the Conditional Highlights view of the Cube Manager.


Conditional Highlights
Search highlights
All highlights by measure
Sort
Sum of QTY (1)
Highlight 1: Sum of QTY is greater than 20000
Cell Text

1 To create a new conditional highlight:

Click in the Cube Manager to open the Conditional Highlights view.

2 Click in the Conditional Highlights view. The **Add Conditional Highlight** dialog box appears.

3 If you want to create a condition that is based on rank, then select **Rank** in the **Rule** area and perform the following steps:

a Specify the measure that you want to use. The list of measures includes all measures that are available in the cube.

b Select whether you want the top (largest) or bottom (smallest) values returned in the ranking. By default, the top values are returned.

c Enter the numeric value for the ranking. The default value is `10`.

d Specify how you want the rank to be determined:

- **Count** - returns the top or bottom *n* members, where **n** is the numeric value that you specified. For example, if you specified **Bottom** and a numeric value of `5`, then the filter would return the bottom, or smallest, five members.
- **Percent by Count** - returns the top or bottom *n* percent of members, where *n* is the numeric value that you specified. For example, if you specified **Top** and a numeric value of `10`, then the filter would return the top, or largest, 10% of the members. If there were 500 members in the set, the top 50 members would be returned.
- **Percent by Value** - returns the top or bottom *n* percent of the sum of the values of the selected measure, where *n* is the numeric value that you specified. For example, if you specified **Top** and a numeric value of `90`, then the sum of the members that are returned would equal at least 90% of the total sum of all of the measure values. If the member values that are being filtered include 6, 3, and 1, which add up to 10, then the

members that are returned would be 6 and 3, which add up to 9, or 90% of 10.

- **Sum** - returns the top or bottom members until their sum is at least the numeric value that you specified. For example, if you specified **Top**, a numeric value of `10`, and the member values to filter include 5, 3, 2, and 1, then the members that are returned would be 5, 3, and 2 because their sum is 10.

e Specify whether you want to include ties and missing values in your filter results. By default, ties are included. Click the ties and missing values option to display the check boxes.

- **Include ties** - includes multiple members that meet the criteria for the last member to return in your filter results. If you want to include only one arbitrarily chosen last member, then clear the **Include ties** check box.
- **Include missing values** - includes missing values in the values that are ranked. This option is not selected by default.

4 If you want to create a condition that is based on range, select **Range** in the Rule area and perform the following steps:

a To create a conditional highlight, you must select the measure that you want to use. The list of measures includes all measures that are available in the cube.

b Select the comparison operator that you want to use. The default operator is **Is equal to**. To negate the comparison expression that you create, click the comparison operator and select **Not**. This option is not selected by default.

c In the value box, enter the value that you want to compare or select a measure from the drop-down list.

5 Use the **Highlight** area to define the formatting that you want to apply to values that meet the criteria that you specified in the **Rule** area. You can specify options for the text font, text size, text color, and text style. You can also specify a color for the cell background and select an icon to display in the cell. A preview of the formatting options that you choose is displayed in the **Preview** box

6 Select whether you want to apply the conditional highlight to all levels or only to levels that you select. Click the **Apply conditional highlight to** option to open the dialog box.

- If you want to apply the conditional highlight to all levels, select **All Levels**.
- If you want to apply the conditional highlight only to specific levels, select **Selected Levels**. Expand the hierarchies and select the levels to which the highlight should be applied. Click **OK**.

7 Enter the name of the highlight.

8 Click **OK** to save the highlight.

Applying and Editing Conditional Highlights

If you create more than one conditional highlight, they are applied in the order in which you create them. The highlighting that you see in the table view can change depending on the order in which the highlights are applied. You can change the order in which highlights are applied in the Conditional Highlights view by moving them up or down the list of highlights.

For example, you create two conditional highlights:

```
Highlight A   Sum of Sales > 100,000   values displayed in red
Highlight B   Sum of Sales > 400,000   values displayed in blue
```

If you apply Highlight A first and then apply Highlight B, then cells with values that are greater than 400,000 are displayed in blue, and cells with values that are between 100,000 and 400,000 are displayed in red. If you apply Highlight B first and then apply Highlight A, then all cells with values that are greater than 100,000 are displayed in red.

Moving the highlights in the View Manager is the only way to change the order in which the highlights are applied to the table view. Moving the highlights up and down the list of highlights in the Conditional Highlights pane of the Cube Manager does not change the order in which they are applied. You can also edit existing highlights and remove and reapply them to the table view.

Moving the Rows and Columns

You can move the rows and columns in the OLAP Analyzer in several ways. You can pivot the table view by swapping the rows and columns. You can also move measures and hierarchies between the row and column axes as well as within the current axis. The OLAP Analyzer remembers any changes that you make each time you open it.

To swap rows and columns, click **Pivot**. The table and graph views are updated.

To move measures and hierarchies in the table view, right-click the measure or level heading that you want to move and select the appropriate option.

- If you have selected a single measure heading, the **Move** option moves only the selected measure.
- If you have selected a measure level heading, the **Move Measures To** option moves all measures.
- If you have selected a level heading, the **Move *hierarchy-name* To** moves the entire hierarchy.

Customize the Appearance of the Graph View

Note: The graph view of the Cube Manager is not available if you are connected to a SAS Viya environment or to a server that is running SAS 9.4M8 or later.

You can customize the appearance of the graph view of the OLAP Analyzer by changing the graph appearance and the graph type and by selecting a predefined color scheme. You can also create a customized scheme by modifying one of the predefined schemes. By default, a title for the graph is automatically generated by using the dimensions and measures in the graph. You can change the default title, and you can also add additional lines to the default title.

To change the chart type, right-click on the background area of the graph in the graph view, and select **Chart Type**. If you select **Automatic Chart**, then the OLAP Analyzer chooses the best chart for the data based on the number of measures and whether the data includes a time dimension.

To set style and graph options:

1. Right-click on the background area of the graph in the graph view, and select **Graph Properties**. The Properties dialog box appears.

2. On the **Graph** tab, select the style that you want to use from the **Style** drop-down list. To make changes to the selected style, click **Edit Style**.

3. Select the type of view that you want to use for the graph.

4. If the chart type that you have chosen uses grid lines, you can select whether to display the grid lines and how to place them.

5. To display annotations, select **Annotations**. When they are available, annotations contain data details about graph areas.

6. To specify advanced settings for mouse preferences, Microsoft PowerPoint sessions, scroll bars, graph rendering, and lighting, click **Advanced**. The Advanced Settings dialog box appears. Select the options that you want to use, and then click **OK** to return to the Properties dialog box.

7. To specify options for specific elements of the chart type that you have chosen and for legends and titles, click the appropriate tab.

8. Click **OK** to save your changes and return to the OLAP Analyzer.

To change the graph title:

1. Right-click on the background area of the graph in the graph view, and select **Graph Properties**. The Properties dialog box appears.

2. Click the **Titles** tab.

3 To change the default title, select **Title 1** from the titles drop-down list and enter the new title that you want to use in the title box. To change the title back to the default title, delete the text in the title box for Title 1 and click **OK**.

4 To add additional lines of text to the title, select **Title 2** from the titles drop-down list and enter the text that you want to use in the title box. You can repeat this step for Title 3 and Title 4.

5 To change the position of a title, select the title that you want to move from the titles drop-down list and then select the position from the **Position** drop-down list. By default, all titles are centered over the graph.

6 To turn off the display of a title, select the title that you want to remove from the titles drop-down list and clear the **Show title** check box.

7 Click **OK** to save your changes.

Setting the OLAP Viewer Options

Specifying Options for Parent Totals and Visual Totals

To access the parent totals and visual totals options, select **Tools** ⇨ **Options** and then click **OLAP Data** in the selection pane. Click **Other View Options**, and then click **Parent Totals and Visual Totals** in the selection pane.

- **Parent totals** – displays the total of all children of the parent level. This total includes all children members, regardless of whether they are displayed in the table view.
- **Visual totals** – displays the total of the values that are displayed in the table view. Because these visual totals change dynamically as you add and remove members from the table view, they can take longer to process.
- **On bottom (right)** – displays the totals below the values that are being totaled. This option is not available in all situations.

  Note: You can display visual totals on the bottom only if you are using a SAS 9.2 OLAP server.

- **On top (left)** – displays the totals above the values that are being totaled.
- **Itemize total and parent values across hierarchies** – displays all of the possible crossings, or summarizations of the unique combinations of variables, and totals. This option applies only when there is a cross-join between dimensions, not including the Measures dimension. If you do not select this option, then the

intermediate totals are removed and some crossings are removed when the totals are visual and there is more than one dimension on an axis. This option is selected by default.

Specifying Options for Raw Values and Unique IDs

To access these options, select **Tools** ⇨ **Options**, and then click **OLAP Data** in the selection pane. Click **Other View Options**, and then click **Raw Values and Unique IDs** in the selection pane.

- **Show unique member IDs for table headings** – displays the unique member identifiers, or keys, in the table headings instead of the captions.
- **Show raw values on data cells** – displays the raw value for each value in the table. To see the raw value, place the mouse pointer over the small blue triangle in the upper right corner of the table cell. In some instances, the raw value might differ from the displayed value.

Specifying Options for Optimizing Results

To access the optimizing results options, select **Tools** ⇨ **Options**, and then click **OLAP Data** in the selection pane. Click **Other View Options**, and then click **Optimize Results** in the selection pane.

- **Show empty rows in results** – includes the empty rows in the table and graph. This option is selected by default.
- **Show empty columns in results** – includes the empty columns in the table and graph. This option is selected by default.
- **Show 'ALL' member in results** – includes the [All] member in all of the table views of the cube. The [All] member includes all members of a dimension. This option is selected by default.

  Note: This option is available only for table views. This option does not affect the graph views or the Cube Explorer.

- **Show calculated members in results** – includes calculated members and calculated measures in the table and graph views when you navigate in the cube. This option is not selected by default.

Specifying Options for Charts

To access the chart options, select **Tools** ⇨ **Options**, and then click **OLAP Data** in the selection pane. Click **Other View Options**, and then click **Chart** in the selection pane.

- **Chart type** – sets the type of graph that is displayed.

  Note: If you select **Automatic Chart**, the OLAP Analyzer chooses the best chart for the data based on the number of measures and whether the data includes a time dimension.

- **Group by** – groups the data in the graph by the column values or the row values. By default, the data is grouped by rows.
- **Put groups on** – specifies how the groups are displayed in the graph. The options that are available depend on the chart type.

  Note: This option is available only for table views. This option does not affect the graph views or the Cube Explorer.

- **Show calculated members in results** – includes calculated members and calculated measures in the table and graph views when you navigate in the cube. This option is not selected by default.

Specifying Options for Styles

To access the style options, select **Tools** ⇨ **Options**, and then click **OLAP Data** in the selection pane. Click **Other View Options**, and then click **Styles** in the selection pane.

- **Style** – specifies the predefined color scheme for all of the table and graph views.
- **Row header** – specifies the color of the cells for the row headings.
- **Column header** – specifies the color of the cells for the column headings.
- **Measure header** – specifies the color of the cells for the measure headings.
- **Graph style** – specifies the predefined color scheme for the graph view.
- **Show shadings based on levels** – shades the cells of the table view based on the dimension levels. The deeper in the dimension a cell is, the darker it is shaded.

Specifying Options for Optimizing Performance

To access the optimize performance options, select **Tools** ⇨ **Options**, and then click **OLAP Data** in the selection pane. Click **Other View Options**, and then click **Optimize Performance** in the selection pane.

- **Maximum number of rows or columns to process** – specifies the maximum number of rows or columns to process from the current MDX query results. The default is 10,000, which means that up to 10,000 rows and 10,000 columns can be processed. The maximum value is 2,147,483,647.
- **Number of rows or columns to retrieve at one time** – specifies the number of rows or columns that you want to retrieve at one time when you are displaying data in the OLAP Analyzer. The default value is 100. This value must be less than the value that is specified for the **Maximum number of rows or columns to process** option.

Specifying Options for Data Drill-Through

To access the data drill-through options, select **Tools** ⇨ **Options**, and then click **OLAP Data** in the selection pane. Click **Other View Options**, and then click **Data Drill Through** in the selection pane.

Maximum number of rows to include detailed data specifies the maximum number of rows to display when you are drilling through to detail data. The default value is 100. The maximum value is 2,147,483,647.

Specifying Options for Data Slice

To access the data slice options, select **Tools** ⇨ **Options**, and then click **OLAP Data** in the selection pane. Click **Other View Options**, and then click **Data Slice** in the selection pane.

The type of slice determines how the data in a slice table is arranged.

- **Rows, columns, and measures as separate data columns** – creates separate data columns for the row headings, the column headings, and each measure value. Use the **Number of rows to scan to estimate column width** option to specify how many rows you want the OLAP Analyzer to read in order to estimate the width of a column. You must enter a value between 1 and 2,147,483,647. The default value is 500. If the width of the rows that occur after the rows that are scanned as an estimate differs from the width of the rows that are scanned, then the data slice might not display all of the columns correctly.

- **Maintain the same layout as the OLAP table** – creates the same rows and columns that are displayed in the current table view.

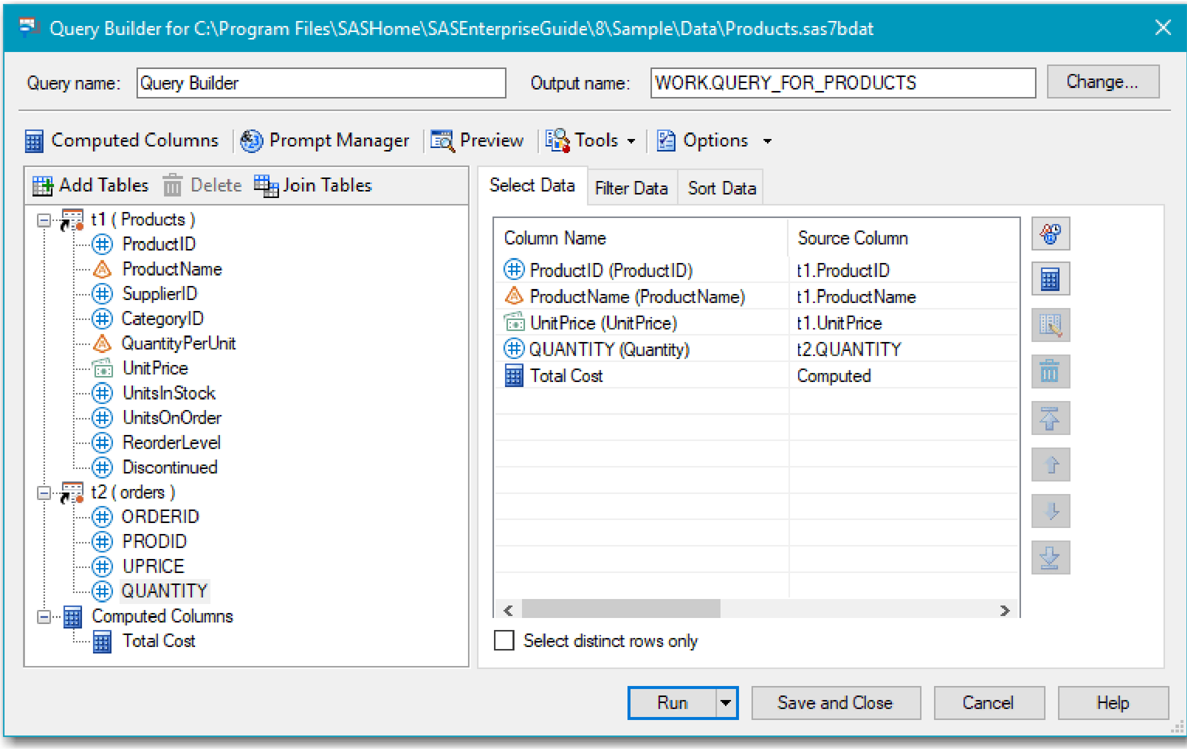
7

Building Queries

About the Query Builder

A query enables you to extract data from one or more tables according to criteria that you specify. You can use the Query Builder to create a query that is based on only one table, or you can join tables together. In the background, the Query Builder generates Structured Query Language (SQL) code, which you can view and use as the basis for a new program.

To use the Query Builder, you must create or open a project. For more information, see “Creating a Project” on page 29. In the Project pane or process flow, right-click the table that you want to query and select Query Builder from the pop-up menu. You can also access the Query Builder by selecting **Data** ⇨ **Query Builder** from the Tasks pane.



Query Builder for C:\Program Files\SASHome\SASEnterpriseGuide\8\Sample\Data\Products.sas7bdat

Query name: Query Builder

Output name: WORK.QUERY_FOR_PRODUCTS

Column Name	Source Column
ProductID (ProductID)	t1.ProductID
ProductName (ProductName)	t1.ProductName
UnitPrice (UnitPrice)	t1.UnitPrice
QUANTITY (Quantity)	t2.QUANTITY
Total Cost	Computed

☐ Select distinct rows only

Note: You must have an open project in order to use the Query Builder. For more information, see Chapter 4, "Working with Projects," on page 27.

TIP If you are creating a simple query that involves filtering or sorting a single table, you can also use the Filter and Sort window to create your query. For more information, see "About Filtering and Sorting Data" on page 279.

Setting Options for the Current Query

You can set results, server, prompt, and title options for the current query by using the Query Options window. To open this window, click **Options** ⇨ **Options for This Query** from within the main Query Builder window.

Setting Results Options

Override the corresponding default settings

Overrides the format setting for query results that you have specified in the Options window.

in Tools->Options

Data table

Saves query results as a static data table against which you can run SAS Enterprise Guide tasks. For more detailed information about the data table format, see “Choosing the Results Format” on page 275.

Data view

Saves query results as a dynamic data view against which you can run SAS Enterprise Guide tasks. For more detailed information about the data view format, see “Choosing the Results Format” on page 275.

Note: Query results cannot be saved as a data view for queries in which one or more of the tables is a Microsoft Excel file, a slice of an OLAP cube, or a file that is stored in a temporary library. For those queries, the query results are automatically saved as a data table.

Report

Saves the query results as one or more reports in the formats that you have specified on the Results General page of the Options window: HTML, SAS Report, Listing, Excel, PDF, PowerPoint, or RTF. Query results in this format are not updated until you rerun the query. You cannot run SAS Enterprise Guide tasks against query results in this format. For more detailed information about the report format, see “Choosing the Results Format” on page 275.

Name

Specifies the library and file name of the results. To change the library or file name, click **Change**. To save the output data set in a different library, click **Servers** in the Save File window, and then navigate to the library that you want to use. You can type a new file name in the **File name** box. Click **Save**.

Note: To reset your selection back to the previously saved location, click **Reset**.

Label

Specifies the label for the results.

Options

Enables you to specify a SAS data set or SAS/ACCESS data set option for the output data. This option is available only if the results format is a data table or a data view. Here are a couple of examples:

- For a SAS data set, you could specify options such as DROP=, KEEP=, or OBS=. For more information about SAS data set options, see "Data Set Options".
- For a database table, you could specify options such as DBFORCE=, DBNULL=, or DBSASTYPE=. For more information about SAS/ACCESS data set options, see "SAS/ACCESS Data Set Options" in SAS/ACCESS Software for Relational Databases: Reference. For more information, see “About the

Data Set Options for Relational Databases” in *SAS/ACCESS for Relational Databases*.

Limit number of matching rows to process

Restricts the number of rows that your query processes from each source table. You can select any number between 1 and 100,000,000, inclusive. This option corresponds to the INOBS= option in PROC SQL and is useful when you want to reduce processing time in order to test a query. For more detailed information about limiting input rows, see “Specifying Limits for Input Processing” on page 274.

Limit number of rows to save in output

Restricts the number of rows that your query saves in the output. You can select any number between 0 and 100,000,000, inclusive. This option corresponds to the OUTOBS= option in PROC SQL and is useful when you want to restrict the size of your output for testing purposes. For more detailed information about limiting output rows, see “Specifying Limits for Output Processing” on page 275.

Setting Server Options

Server

Specifies the server on which you want to execute the query. By default, SAS Enterprise Guide executes the query on the server on which the first table that was added to the query resides.

Note: Changing the server could affect performance. Any input data in the query that does not reside on the selected server is automatically copied to the server before the query can run.

Setting Pass-through Options

Generate source for explicit pass-through

Adds the CONNECT TO statement to the automatically generated SQL code so that it can be processed by the database that was used to define the tables in the query. This option should be used only when you want the query to be processed by the database instead of by SAS. When you select this option, you must also specify the appropriate connection information. This option is available only when your query includes at least one database table.

Note: Using explicit pass-through mode is an option for advanced users with database expertise. This option might require you to create a program with the SQL statements that are automatically generated by SAS Enterprise Guide and to make modifications to the code. If you do not select this option, then SAS Enterprise Guide uses the SAS/ACCESS database engine to create code specifically for the target database.

Note: For some examples of how to specify the options to connect to different databases, see "Running a Query in Explicit Pass-Through Mode" on page 205.

Database

Specifies the database engine that you want to use. You must select or enter a value for this option in order for the pass-through to work. For example, if you select ORACLE to specify the Oracle database, then the CONNECT TO statement would appear as follows:

```
CONNECT TO ORACLE as con1
```

Options

Specifies the connection options that are used when you connect to the database. You can use the default options that SAS Enterprise Guide provides, or you can enter your own options. For information about specific options, see the PROC SQL documentation for your database. For example, if you specified MYDBOPTIONS=here with an Oracle connection, then the CONNECT TO statement would appear as follows:

```
CONNECT TO ORACLE as con1
   (MYDBOPTIONS=here);
```

Use AuthDomain Option

Uses the AUTHDOMAIN= option with the CONNECT TO statement to retrieve the connection credentials from a SAS Metadata Server by using the **Name** value that you specify.

- **Name** – Specifies the name of the authentication domain to use with the AUTHDOMAIN= option. For example, if you have selected the **Use AuthDomain Option** option and you specified *MyOracleAuth* as the name, then the CONNECT TO statement would appear as follows:

  ```
  CONNECT TO ORACLE as con1
  (MYDBOPTIONS=here authdomain=MyOracleAuth);
  ```

Use Specified Credentials

Enables you to specify a user name and password to be used in the CONNECT TO statement. The password is encrypted after you enter it so that it does not appear in plain text in the code or log windows.

- **User** – Specifies the user name that is used to connect to the database.
- **Password** – Specifies the password that is used to connect to the database.

For example, if you have selected the **Use Specified Credentials** option, and you specified a user name of *MyTestUser* and a password of *MyTestPassword*, then the CONNECT TO statement might appear as follows after the password is encrypted:

```
CONNECT TO ORACLE as con1
  ( user="MyTestUser"
    pw="{sas002}B02D643D42B9CB8B3F5E1C6B28DDE842474750B031733D8E");
```

Keep results entirely in database when possible

Creates the output data in the database library. If the output data is very large, this option can improve your performance because the data is not copied back to the SAS server.

Note: This option is available only when both the input and output data is stored in the same database library.

Setting Prompt Options

Project prompts used

Displays the SAS name, display name, and data type of each prompt that you have associated with the query. You can add and remove prompts from the query and create new prompts by using the Prompt Manager. For more information, see "Using Prompts in a Query" on page 250.

- To associate existing project prompts with the query, click **Add** and select the prompt that you want to add.
- To remove a prompt from the query, select the prompt that you want to remove and click **Remove**.
- To move the prompts up or down the list, select the prompt that you want to move and click ⇧ or ⇩. The order of the prompts in the Project prompts used list determines the order in which the prompts are displayed to the user in the prompt window when the query is run.
- To create a prompt, click **Prompt Manager** and then click **Add**. For more information, see "About Creating Prompts" on page 385.

Setting Title Options

Title text for query output

Displays the default title text for the query report output. Select **Override the corresponding default settings in Tools->Options** and type a new title to override the default title that is specified on the Tasks General page of the Options window. For more information, see "Setting General Task Options" on page 504.

Footnote text for query output

Displays the default footnote text for the query report output. Select **Override the corresponding default settings in Tools->Options** and type a new footnote to override the default footnote that is specified on the Tasks General page of the Options window. For more information, see "Setting General Task Options" on page 504.

Setting Other Options

Automatically show or hide summarized data sections	Displays the options for summarized data on the Select Data and Filter Data tabs even if there is no summarized data in the query. This option is selected by default. You might need to clear this check box if you have a query from a migrated project that uses a GROUP BY or HAVING clause without any summarized data in the SELECT statement of the query. For more information, see “Using Summary Functions” on page 252.
Automatically include computed columns in group by	Includes computed columns in the list of automatically selected columns by which you can group the data for a summarized column. The list of automatically selected columns is located on the Select Data tab and is enabled only when you have a summarized column or a computed column that contains a summary function. This option is selected by default and applies only when your query includes both a summarized column and a computed column.

Customizing Data Display in the Query Builder

By default, input data is displayed in the Query Builder in the order in which it occurs in the input table, and columns are displayed by name. You can change how data is sorted and identified in the Query Builder in any areas that display the input data in a tree view, such as the main Query Builder window, the New Filter wizard, and the Tables and Joins window. These changes do not affect how the output appears; they affect only how the data is displayed within the Query Builder.

To sort all data within the Query Builder

- In the Query Builder, select **Options** ⇨ **Sort Columns**, and then select one of the following options:
  - **by Data Order** – displays the columns in the order in which they appear in the data table.
  - **by Ascending Order** – displays the columns in ascending alphabetical order.
  - **by Descending Order** – displays the columns in descending alphabetical order.

The sort order of the columns is immediately updated.

Note: To change the sort order of the data in only one particular area of the Query Builder, right-click on the white space around the columns in that area, and then select **Sort Columns** and the option that you want to use. Only the columns in that area are updated.

To display columns by label

- In the Query Builder, select **Options** ⇨ **Show Columns by Label**. The column display is immediately updated.

  To display the columns by name, select **Options**, and then clear the **Show Columns by Label** option.

Viewing Query Properties

You can view and change the properties for each query item in the project.

To view the properties for a query, right-click the query item in the Project pane or the query object in the process flow and select **Properties**.

General Properties

Label	Name of the query. You can update the name of the query.
Last Execution Time	Displays how long it took for the query to run the last time it was executed. This time is rounded to the nearest second and does not include the time it takes to connect to the server.
Server	Specifies the server on which you want to execute the query. You can select a new server for your query only from within the Query Builder window.
Input tables	Displays the full pathname of the tables that are used in the query and the server where the table is saved.

Results Properties

Result format	■ **Override the corresponding default settings in Tools > Options** – Overrides the format setting for query results that you have specified in the Options window.

- **Data table** – Saves query results as a static data table against which you can run SAS Enterprise Guide tasks.
- **Data view** – Saves query results as a dynamic data view against which you can run SAS Enterprise Guide tasks.
- **Report** – Saves the query results as one or more reports in the formats that you have specified on the Results General page of the Options window: HTML, SAS Report, Listing, Excel, PDF, PowerPoint, or RTF. Query results in this format are not updated until you rerun the query. You cannot run SAS Enterprise Guide tasks against query results in this format.

Note: For more information, see “Choosing the Results Format” on page 275.

Output data

- **Name** – Specifies the library and file name of the results. To change the library or file name, click **Change**.
- **Label** – Specifies the label for the results.

  Note: This option is not available if you have selected Report as your results format.

Query limits

- **Limit number of matching rows to process** – Restricts the number of rows that your query processes from each source table. You can select any number between 1 and 100,000,000, inclusive. This option corresponds to the INOBS= option in PROC SQL and is useful when you want to reduce processing time in order to test a query. For more information, see “Specifying Limits for Input Processing” on page 274.

  Note: This option is not available if you have selected Data view as your results format.

- **Limit number of rows to save in output** – Restricts the number of rows that your query saves in the output. You can select any number between 1 and 100,000,000, inclusive. This option corresponds to the OUTOBS= option in PROC SQL and is useful when you want to restrict the size of your output for testing purposes. For more information, see “Specifying Limits for Output Processing” on page 275.

Code Submission Properties

Use preferences from project properties

Specifies that the code submission settings for the query should use the settings that are specified in the Code Submission section of the Properties for *project-name* window. For more information, see “Viewing Project Properties” on page 37.

Customize code submission options

Specifies that you want to customize the code submission options for this query. You can choose from the following options:

- **Use grid if available** – Enables SAS Enterprise Guide to run tasks in a SAS grid computing environment. For an overview of SAS grid support in SAS Enterprise Guide, see “Using SAS Enterprise Guide with a SAS Grid” on page 12.
- **Allow parallel execution on the same server** – Enables SAS Enterprise Guide to create one or more additional workspace server connections so that parallel process flow paths can be run in parallel on the same server. For more information, see “Using SAS Enterprise Guide with Parallel Execution” on page 13.

  Note: Because there might be more than one workspace server connection, any changes that you make in one workspace session, such as library assignments or changes to macro variables or options, might not be available to tasks that are running in another parallel workspace session.

Action to take on error in the execution of this query

Specifies how to handle an error in the execution of this query.

- **Stop all items** – Stops the execution of all subsequent items in the project or process flow.
- **Stop current branch** – Stops the execution of all subsequent items in the current branch of the process flow. This is the default value.
- **Continue execution** – Continues the execution of all subsequent items in the project or process flow.

Prompts Properties

Project prompts used

Displays the SAS name, display name, and data type of each prompt that you have associated with the query. You can add and remove prompts from the query and create new prompts by using the Prompt Manager. For more information, see “Using Prompts in a Query” on page 250.

- To associate existing project prompts with the query, click **Add** and select the prompt that you want to add.
- To remove a prompt from the query, select the prompt that you want to remove and click **Remove**.
- To move the prompts up or down the list, select the prompt that you want to move and click ⇧ or ⇩. The order of the prompts in the **Project prompts used** list determines the order in which the prompts are displayed to the user in the prompt window when the query is run.

- To create a prompt, click **Prompt Manager** and then click **Add**. For more information, see "About Creating Prompts" on page 385.

Summary Properties

This is a summary of all the properties that are associated with the query. These properties are for informational purposes only; you are not able to edit any of these properties from this location.

Extracting Code from a Query

You can extract a copy of the SQL code that is generated automatically when you create a query in the Query Builder. After you extract the code, you can make changes to the code and then run it.

To extract the code from a query

- In the Project pane or the process flow, right-click the query whose code you want to use and select **Extract code**. A new code item named `Code for query-name` is added to the project. You can make and save changes to the new code.

Note: The extracted code contains a copy of the query code and is no longer associated with the original query. Editing this code does not affect the query, and modifying the original query does not affect the contents of the extracted code.

Replacing a Column

You can replace the name of a column in an existing query with another column name (for example, if the name of a column in the input data has changed). You can choose to replace all occurrences of the name in the query, or you can replace the name only in specified areas, such as filters or computed columns.

To replace the name of a column

1. In the Query Builder, select **Tools** ⇨ **Replace Column**. The Replace Column window opens.
2. In the **Find What** box, type the column name that you want to replace

3. Beside the **Replace with** box, click ... to browse for a replacement column. The Add Column window opens and displays a list of all of the columns in the query.

4. Select the replacement column and click **OK**.

5. From the **Look in** drop-down list, select the area of the query in which you want to replace the specified column. You can choose from among the following areas:

   - **All Occurrences** – replaces all occurrences of the column name in the query.
   - **Select Data** – replaces the occurrences of the column name on the **Select Data** tab.
   - **Filter Data** – replaces the occurrences of the column name on the **Filter Data** tab.
   - **Sort Data** – replaces the occurrences of the column name on the **Sort Data** tab.
   - **Join Order** – replaces the occurrences of the column name on the Join Order pane of the Tables and Joins window.
   - **Computed Columns** – replaces the occurrences of the column name in all of the computed columns in the query.

6. Click **Replace** to replace the column name.

Running a Query in Explicit Pass-Through Mode

About Explicit Pass-Through Mode

When a query uses data from an external database, you can use explicit pass-through mode to send the SQL statements to the database to be processed. If the data files are very large, this can improve your performance because the files do not have to be copied to the SAS server for processing. Once the statements are processed, the results are sent back to SAS Enterprise Guide.

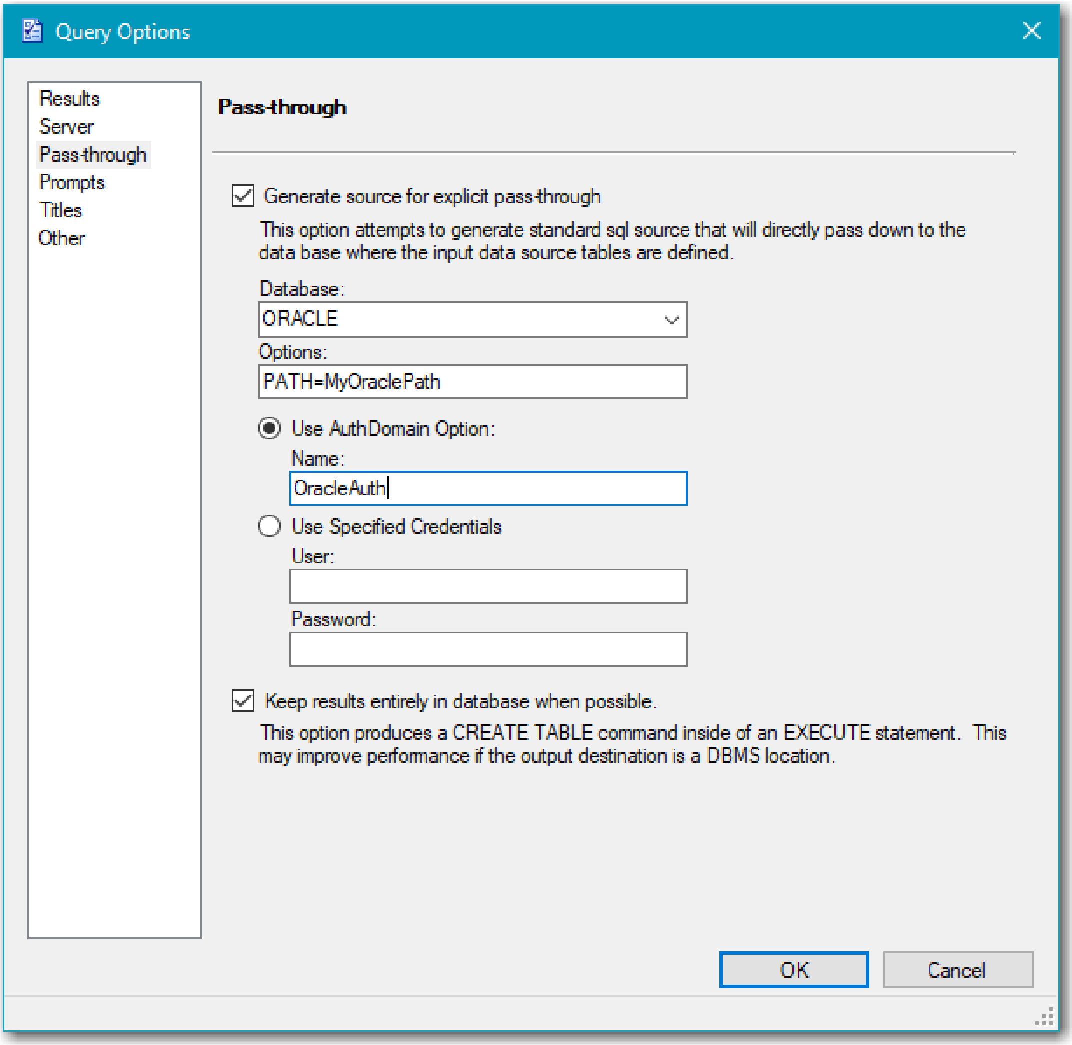
Note: Using explicit pass-through mode is an option for advanced users with database expertise. This option might require you to create a program with the SQL statements that are automatically generated by SAS Enterprise Guide and to make modifications to the code. If you do not select this option, then SAS Enterprise Guide uses the SAS/ACCESS database engine to create code specifically for the target database.

In order to use explicit pass-through mode, you must select the appropriate options in the Query Builder. For more information, see "Setting Pass-through Options" on page 197. There are three different ways to provide the database credentials that you need to use explicit pass-through mode:

- Use the AuthDomain option.
- Use specified credentials
- Include additional required credentials in the **Options** field.

Using the AuthDomain Option

When you use the AuthDomain option, the connection credentials are retrieved from a SAS Metadata Server by using a name that you specify. For example, you could specify options in order to connect to an Oracle database by using the AuthDomain option. In the following example, the database is specified as ORACLE, the connection option specifies the Oracle connection path, and the AuthDomain option is specified as OracleAuth



Pass-through

☑ Generate source for explicit pass-through

This option attempts to generate standard sql source that will directly pass down to the data base where the input data source tables are defined.

Database: ORACLE

Options: PATH=MyOraclePath

● Use AuthDomain Option:

Name: OracleAuth

○ Use Specified Credentials

User:

Password:

☑ Keep results entirely in database when possible.

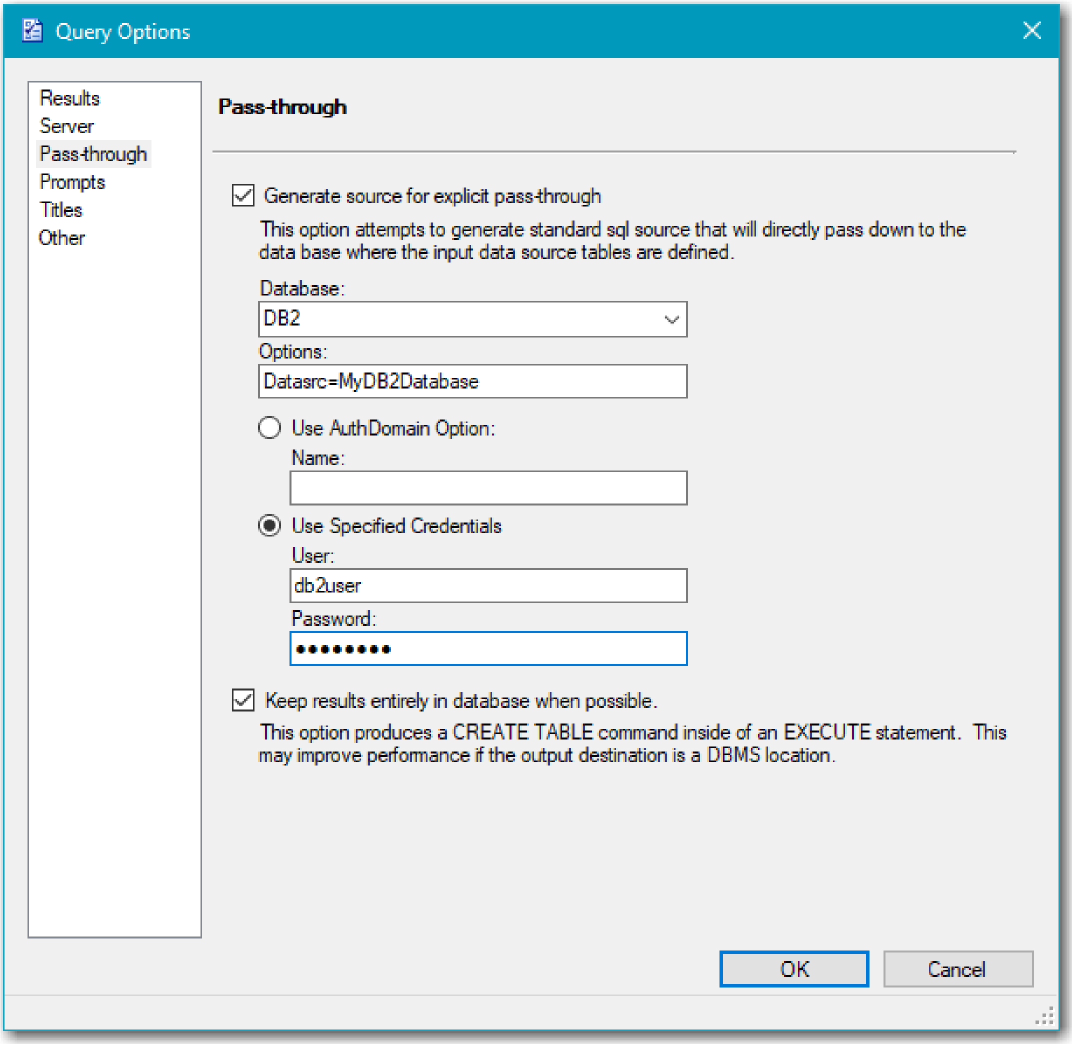
This option produces a CREATE TABLE command inside of an EXECUTE statement. This may improve performance if the output destination is a DBMS location.

When you specify these options, the automatically generated code appears as follows:

```
CONNECT TO ORACLE as con1
    (path=MyOraclePathGoesHere authdomain=OracleAuth);
```

Using Specified Credentials

When you use the specified credentials option, you specify the user name and password that are used to connect to the database. For example, you could specify options in order to connect to a DB2 database by specifying the credentials. In the following example, the database is specified as DB2, the connection option specifies the DB2 connection path, and a sample user name and password are specified.



Pass-through

☑ Generate source for explicit pass-through

This option attempts to generate standard sql source that will directly pass down to the data base where the input data source tables are defined.

Database: DB2

Options: Datasrc=MyDB2Database

○ Use AuthDomain Option:

Name:

● Use Specified Credentials

User: db2user

Password: ••••••••

☑ Keep results entirely in database when possible.

This option produces a CREATE TABLE command inside of an EXECUTE statement. This may improve performance if the output destination is a DBMS location.

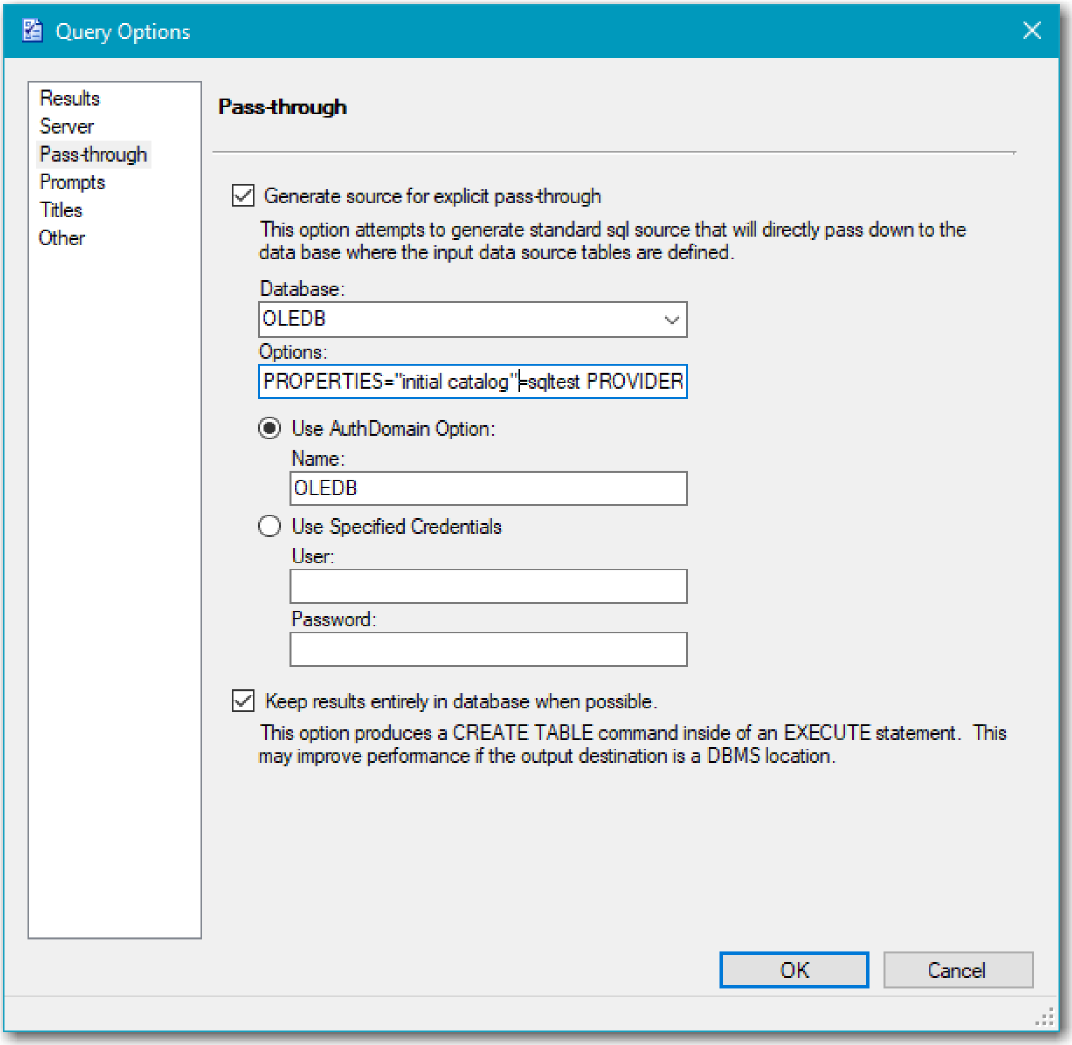
When you specify these options, the automatically generated code might appear as follows after the password is encrypted:

```
CONNECT TO DB2 as con1
```

```
(Datasrc=MyDB2Database user="db2user"
 pw="{SAS002}65CC6A184655B9EC00B682B8481CD329");
```

Connecting to an OLEDB Provider by Using the Options Field

Connections to some database types, such as OLEDB and ODBC, sometimes require several database options in addition to the AuthDomain name or specified credentials. You can enter the additional options in the **Options** field, and they are added to the CONNECT TO statement. For example, you could specify options in order to connect to an OLEDB source, which is linked to a Microsoft SQL Server data source. In the following example, the database is specified as OLEDB, the Options field is used to specify OLEDB connection options, such as Properties and Provider. The AuthDomain option is specified as OLEDB.



Pass-through

☑ Generate source for explicit pass-through

This option attempts to generate standard sql source that will directly pass down to the data base where the input data source tables are defined.

Database:
OLEDB

Options:
PROPERTIES="initial catalog"=sqltest PROVIDER

● Use AuthDomain Option:

Name:
OLEDB

○ Use Specified Credentials

User:

Password:

☑ Keep results entirely in database when possible.

This option produces a CREATE TABLE command inside of an EXECUTE statement. This may improve performance if the output destination is a DBMS location.

When you specify these options, the automatically generated code appears as follows:

```
CONNECT TO OLEDB as con1
    (PROPERTIES="initial catalog"=sqltest PROVIDER=sqlncli
     DATASOURCE=sqlserversource PROMPT=NO authdomain=OLEDB);
```

Accessing the Query Builder

About Accessing the Query Builder

You can use the Query Builder to create a query or to modify an existing query. You can also create a query based on an existing query.

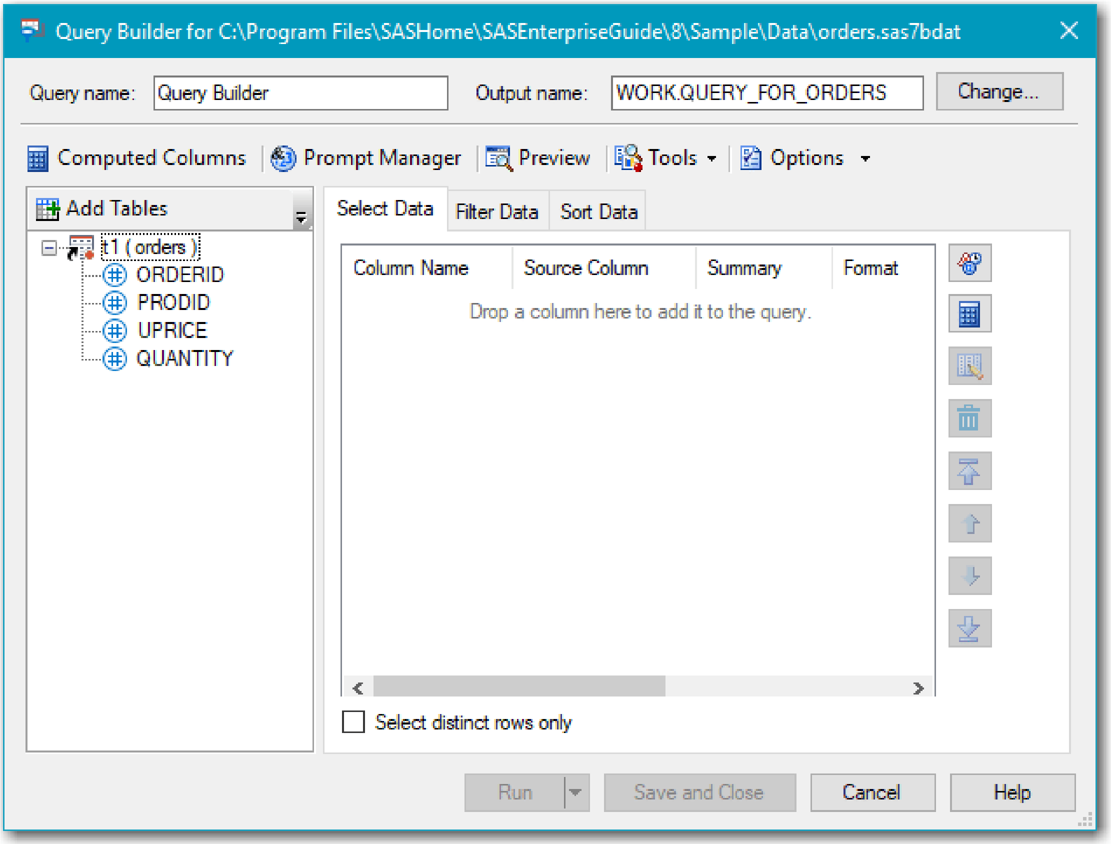
Creating a Query

You can create a query based on data in one or mores tables in your project. The query can be based on an open table, or you can select one or more tables in the project to use in your query. If there is no active data when you open the Query Builder, you are prompted to open the table that you want to use.

Note: You must have an open project in order to use the Query Builder. If you do not have an open project, you are prompted to create one. For more information, see Chapter 4, “Working with Projects,” on page 27.

To create a query based on open data

- Open the data that you want to query and click **Query Builder** on the toolbar. The Query Builder window opens.



Query Builder for C:\Program Files\SASHome\SASEnterpriseGuide\8\Sample\Data\orders.sas7bdat

Query name: Query Builder

Output name: WORK.QUERY_FOR_ORDERS

t1 (orders)
ORDERID
PRODID
UPRICE
QUANTITY

Column Name	Source Column	Summary	Format

Drop a column here to add it to the query.

□ Select distinct rows only

To create a query based on selected tables

- Select one or more tables in the Project pane or process flow and then select **Query Builder** from the Data section of the Tasks pane. The Query Builder window opens. If the selected tables have columns with matching names and data types, the Query Builder automatically joins the tables for you. If no column name and type matches are found, then you are prompted to specify a join for the tables manually, if necessary. For more information, see “Joining Tables Manually” on page 215.

To create a query without open data:

1 Select **Query Builder** from the Tasks pane. The Open window opens so that you can select the table that you want to use.

2 Select the table that you want to use and click **Open**. The Query Builder window opens and the selected table is added to your SAS Enterprise Guide project.

Modifying an Existing Query

To modify an existing query:

- In the Project pane or process flow, right-click the query that you want to modify and select **Modify**. The Query Builder window opens.

Creating a Query Based on an Existing Query

You can use the results of an existing query as the basis for a new query.

Note: You can create a query based on an existing query only if the results of the existing query are saved as a data set or a data view.

To create a query based on an existing query:

1. In the Project pane or the process flow, select the query results that you want to use in your query.
2. Double-click **Query Builder** from the Data section of the Tasks pane. The Query Builder window opens.

Joining Tables

About Joining Tables

The Query Builder enables you to join data from up to 256 different tables. The tables can reside on different servers. By default, if the tables reside on different servers, then SAS Enterprise Guide executes the query on the server where the first table in the query resides. The tables that reside on other servers are temporarily copied to the first server while the query is executed.

Selecting Tables to Join

The tables that you have selected for your query are listed on the left side of the Query Builder window. You can select a table before you invoke the Query Builder, and you can add tables to your query from within the Query Builder.

To add a table to your query

1. In the Query Builder window, click **Add Tables**. The Open Data window appears.
2. To add a table that was already added to your project, click **Project** in the selection pane. Select the table that you want to add to your query and click **Open**.

To add a table that is not already in the project, navigate to the location where the table resides. Select the table that you want to add and click **Open**.

Note: For more information, see "Adding Files to a Project" on page 33.

The Query Builder attempts to join tables by columns that have the same name and type. If no column name and type matches are found, then a warning message appears that tells you to join the columns manually. For more information, see "Joining Tables Manually" on page 215.

After you add the table, a list of the columns in the table is added to the left pane.

Note: By default, the columns in the list are displayed by name instead of label, and they are listed in the order in which they appear in the data table. You can change how the columns are displayed in the Query Builder. For more information, see "Customizing Data Display in the Query Builder" on page 200. Changing how the columns are displayed does not affect how the SQL query is processed or how the columns appear in the output table.

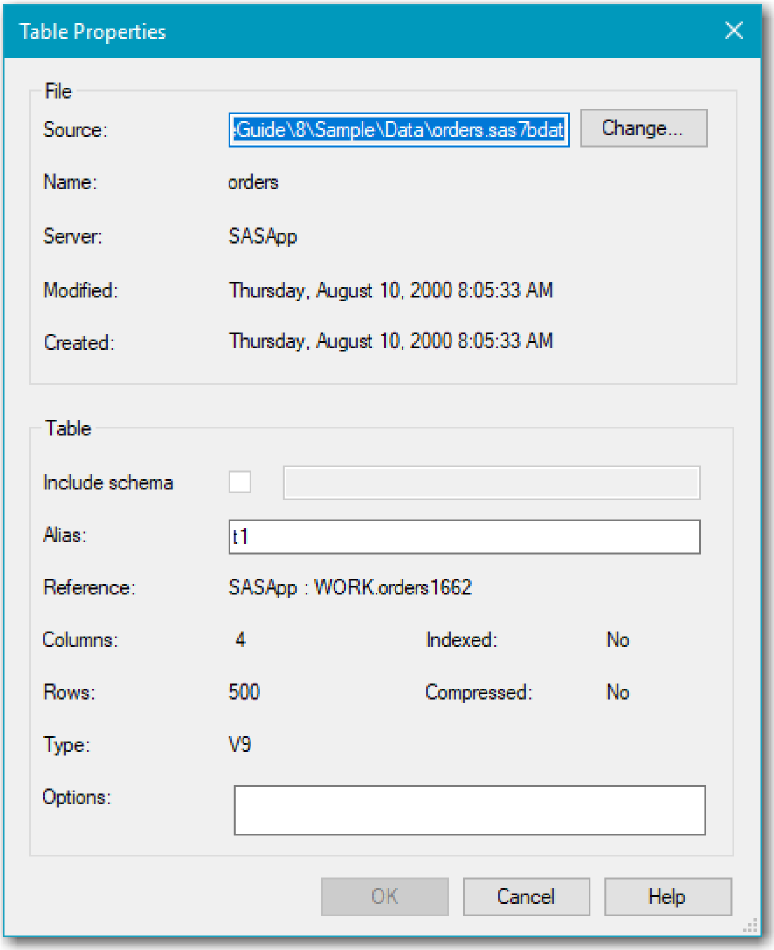
Editing Table Properties in the Query Builder

When you create a query, you can change the file that an input data item points to by changing the file name. You might want to do this so that you can run your query first against sample data before you run it against your real data.

You can also create an alias for the table in the Query Builder, and advanced users can specify input data options. For example, you could specify an option to limit the number of observations or to specify a SAS data type for a column.

To change the file name:

1 In the left pane of the Query Builder, right-click the table that you want to change and select **Properties**. The Table Properties window opens.



Source: :Guide\8\Sample\Data\orders.sas7bdat

Name: orders

Server: SASApp

Modified: Thursday, August 10, 2000 8:05:33 AM

Created: Thursday, August 10, 2000 8:05:33 AM

Include schema □

Alias: t1

Reference: SASApp : WORK.orders1662

Columns: 4 Indexed: No

Rows: 500 Compressed: No

Type: V9

Options:

2 To change the file name, click **Change**. The Open Data window opens. Select the file that you want the data item to point to and click **Open**.

Note: Changing the file name property does not automatically update the query results. You must rerun the query in order to update your results.

3 To specify an alias, enter the alias for the table in the **Alias** box.

4 To specify input data options, enter the options that you want to use in the **Options** box. For information about some of the available options, see Data Set Options.

5 Click **OK** in the Table Properties window to close the window and save your changes.

Selecting the Server

The Query Builder enables you to join data from up to 256 different tables. The tables can reside on different servers. You can select the server on which you want to execute the query. By default, SAS Enterprise Guide executes the query on the server on which the first table that was added to the query resides.

To select a server:

1 In the Query Builder window, select **Options** ⇨ **Options for This Query**. The Query Options window opens.

2 In the selection pane, click **Server**. From the **Server** drop-down list, select the server on which you want to execute the query. The server list includes all of the servers that you can access.

Note: You can choose to execute the query on any server that you can access. However, if there are any input tables in the query that do not reside on the server that you select, those tables are copied to the server before the query can run. Depending on the size of the data that is copied, this can affect performance.

Understanding the Types of Joins

The Query Builder supports nine different types of joins. You can join data from up to 256 different tables. You can select the type of join that you want by modifying an existing join. For more information, see “Modifying an Existing Join” on page 218.

You can select the join option that you want to use in the Join Properties window. For more information, see “Modifying an Existing Join” on page 218.

SAS Enterprise Guide Join Option	Join Type	Description
Matching rows only given a condition	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.
All rows from the left table given a condition	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.
All rows from the right table given a condition	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.
All rows from both tables given a condition	Full outer join	The output rows include all rows from both tables in which the joining criterion is met.
The Cartesian product	Cross join	The output rows include each row from the first table combined with each row from the second

SAS Enterprise Guide Join Option	Join Type	Description
		table. The total number of output rows is the product of the number of rows in each table. For example, if Table A, which contains 9 rows, is combined with Table B, which contains 13 rows, then the number of output rows is 9 * 13, or 117.
Matching rows only with equal common columns	Natural inner join	The output rows include all rows in which the common columns in the two tables contain values that are equal. Common columns are those that have the same name and same data type.
All rows from the left table with equal common columns	Natural left join	The output rows include all rows from the first table and the rows from the second table in which the common columns in the two tables contain values that are equal. Common columns are those that have the same name and same data type.
All rows from the right table with equal common columns	Natural right join	The output rows include all rows from the second table and the rows from the first table in which the common columns in the two tables contain values that are equal. Common columns are those that have the same name and same data type
All rows from both tables with equal common columns	Natural full outer join	The output rows include all rows from both tables. Rows that do not have a match are filled in with missing values.

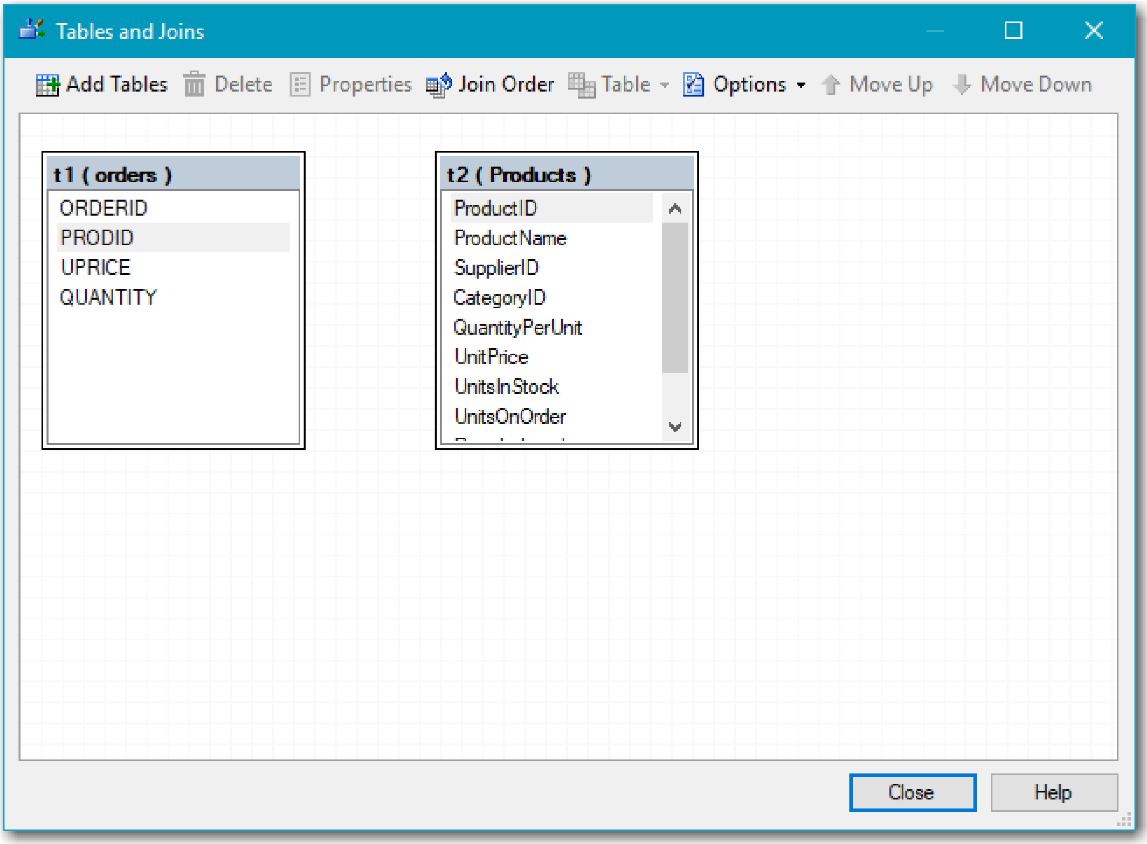
Joining Tables Manually

By default, when you add a table to a query that already has at least one table, the Query Builder looks for columns in two tables that have the same name and data type and assumes that you are joining on equal values of the joining columns. If no column name and type matches are found, then you receive a warning that you should specify the join manually.

You can also choose to apply a filter to the data before the join is processed. This can improve the performance of the join.

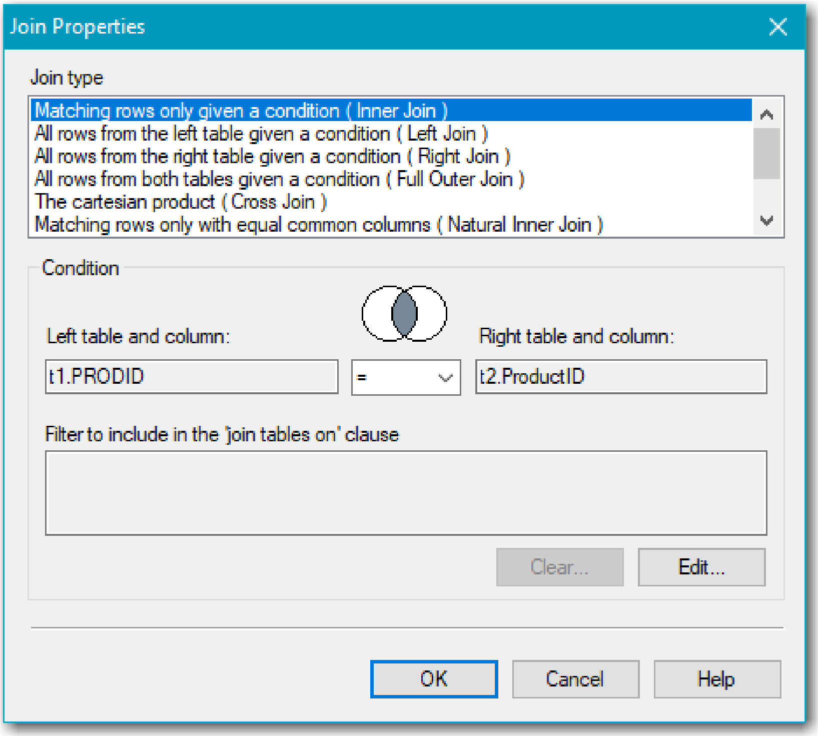
To join two tables manually

1 If the Tables and Joins window is not already open, click **Join Tables** in the Query Builder window. Each table is displayed with a list of columns.



Note: By default, the columns in the list are displayed by name instead of label, and they are listed in the order in which they appear in the data table. You can change how the columns are displayed in the Query Builder. For more information, see "Customizing Data Display in the Query Builder" on page 200. Changing how the columns are displayed does not affect how the SQL query is processed or how the columns appear in the output table. For more information, see "Customizing Data Display in the Query Builder" on page 200.

2 Drag the column name from the table that you want to join to the corresponding column in the table to which you want to join it. The Join Properties window opens.



Join type

- Matching rows only given a condition (Inner Join)
- All rows from the left table given a condition (Left Join)
- All rows from the right table given a condition (Right Join)
- All rows from both tables given a condition (Full Outer Join)
- The cartesian product (Cross Join)
- Matching rows only with equal common columns (Natural Inner Join)

Left table and column: t1.PRODID

=

Right table and column: t2.ProductID

Filter to include in the 'join tables on' clause

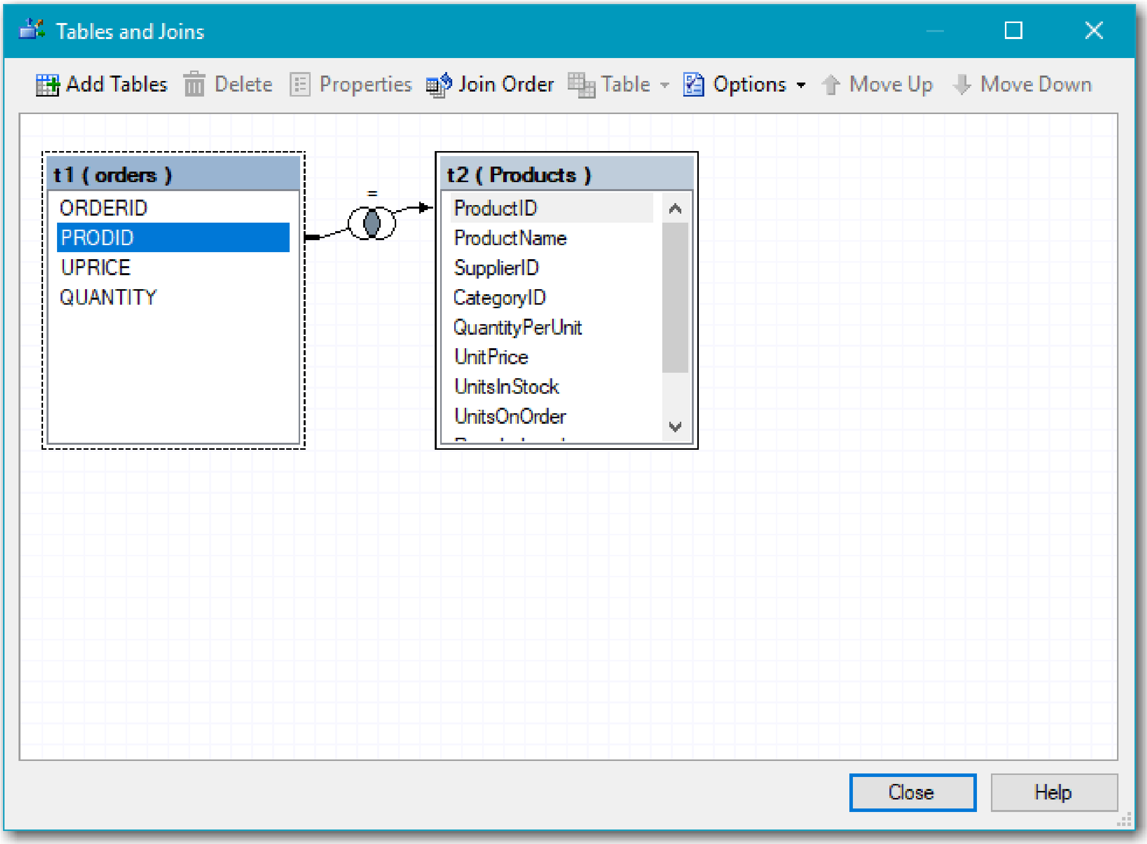
3 Select the join type that you want to use. The default value is **Matching rows only given a condition (Inner Join)**. For more information, see "Understanding the Types of Joins" on page 214.

4 If you have selected a conditional join type, select the comparison operator that is used to compare the columns that are being joined. The default comparison operator is =.

5 If you have selected a conditional join type and you want to add a filter that is processed before the join operation is processed, click **Edit**. The Advanced Filter Builder window opens so that you can create the filter. The filter expression that you create is added to the ON clause in the SQL query.

Note: To remove the filter from the join operation, click **Clear**.

6 Click **OK** to return to the Tables and Joins window. A line appears that links the two column names with a join indicator to represent the type of join and the comparison operator.



Note: After you create the join, you can modify the join by selecting a different comparison operator or type of join or by changing the order of the tables in the join.

7 Click **Close** to save the join.

Note: You can join multiple columns between tables by repeating these steps for any other desired column pairs. When you have multiple joins, the join type is the same for all of the joins between the two tables.

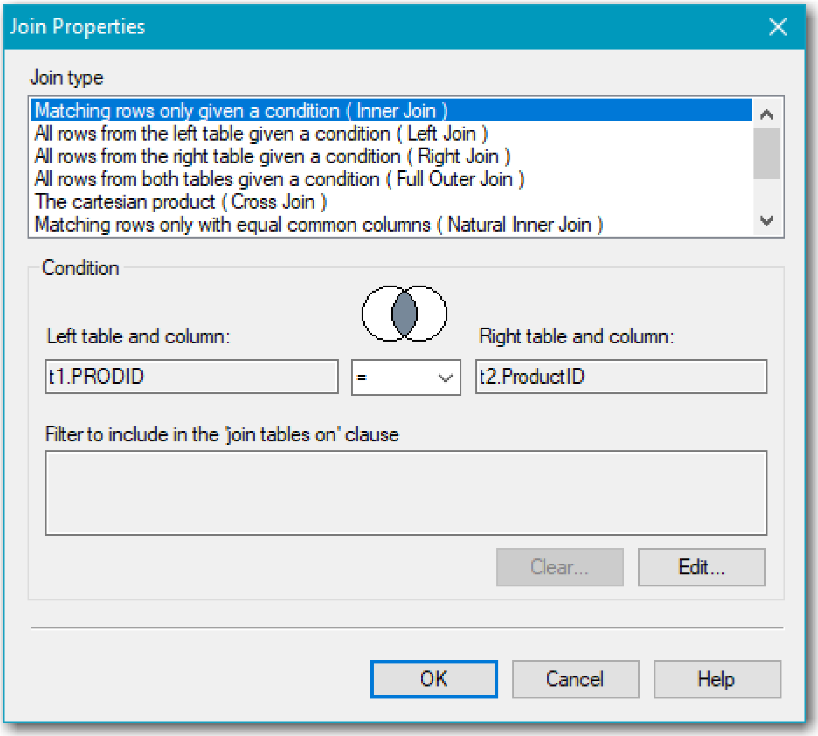
Modifying an Existing Join

You can modify an existing join by selecting a different type of join or a different comparison operator. You can change the order of the tables that are used in the join or remove a join. You can also choose to apply a filter to the data before the join is processed. This can improve the performance of the join.

Note: Changing the order of the tables in the join can have a dramatic effect on performance.

1 In the Query Builder window, click **Join Tables** to open the Tables and Joins window.

2 Select the join indicator on the join line and click **Properties**. The Join Properties window opens.



Join Properties

Join type

Matching rows only given a condition (Inner Join)
All rows from the left table given a condition (Left Join)
All rows from the right table given a condition (Right Join)
All rows from both tables given a condition (Full Outer Join)
The cartesian product (Cross Join)
Matching rows only with equal common columns (Natural Inner Join)

Condition

Left table and column: t1.PRODID

=

Right table and column: t2.ProductID

Filter to include in the 'join tables on' clause

3 If you want to change the type of join, select a new join type. For more information, see "Understanding the Types of Joins" on page 214.

4 If you want to change the comparison operator that is used to compare the columns that are being joined, select a new comparison operator from the drop-down list.

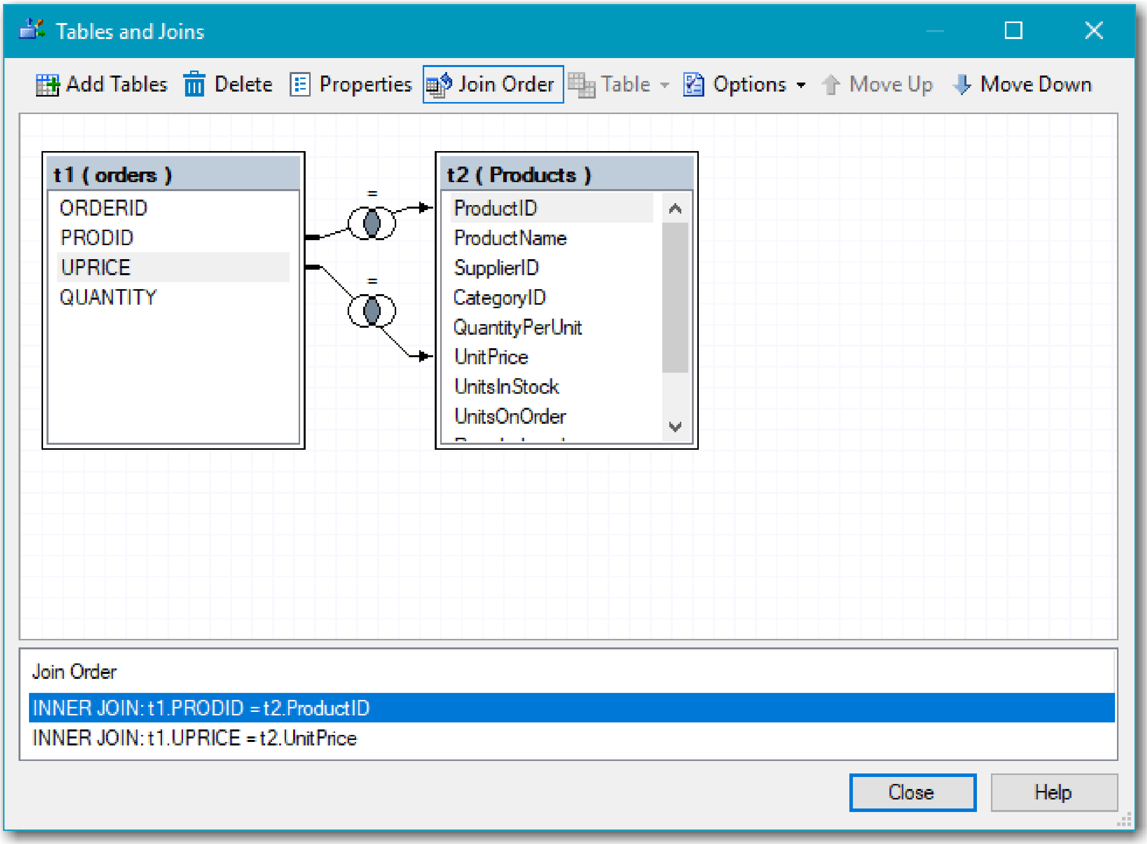
5 If you have selected a conditional join type and you want to add a new filter or edit an existing filter, click **Edit**. The Advanced Filter Builder window opens so that you can create or edit the filter. The filter expression that you create is added to the ON clause in the SQL query and is processed before the join is processed.

6 Click **OK** to save your changes.

To change the order of the tables in the join

1 In the Query Builder window, click **Join Tables** to open the Tables and Joins window.

2 Click **Join Order**. The Join Order pane opens at the bottom of the Tables and Joins window.



Join Order

INNER JOIN: t1.PRODID = t2.ProductID

INNER JOIN: t1.UPRICE = t2.UnitPrice

Note: You can also open the Join Order pane by right-clicking on the white space in the Tables and Joins window and selecting **Join Order**.

3 In the Join Order pane, select the join that you want to move.

4 Click **Move Up** or **Move Down** on the toolbar to move the join up or down the list.

To delete a join:

1 In the Query Builder window, click **Join Tables** to open the Tables and Joins window.

2 Select the join indicator on the join line and click **Delete**.

Note: You can also delete a join by right-clicking the join indicator on the join line and selecting **Delete Join**.

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.

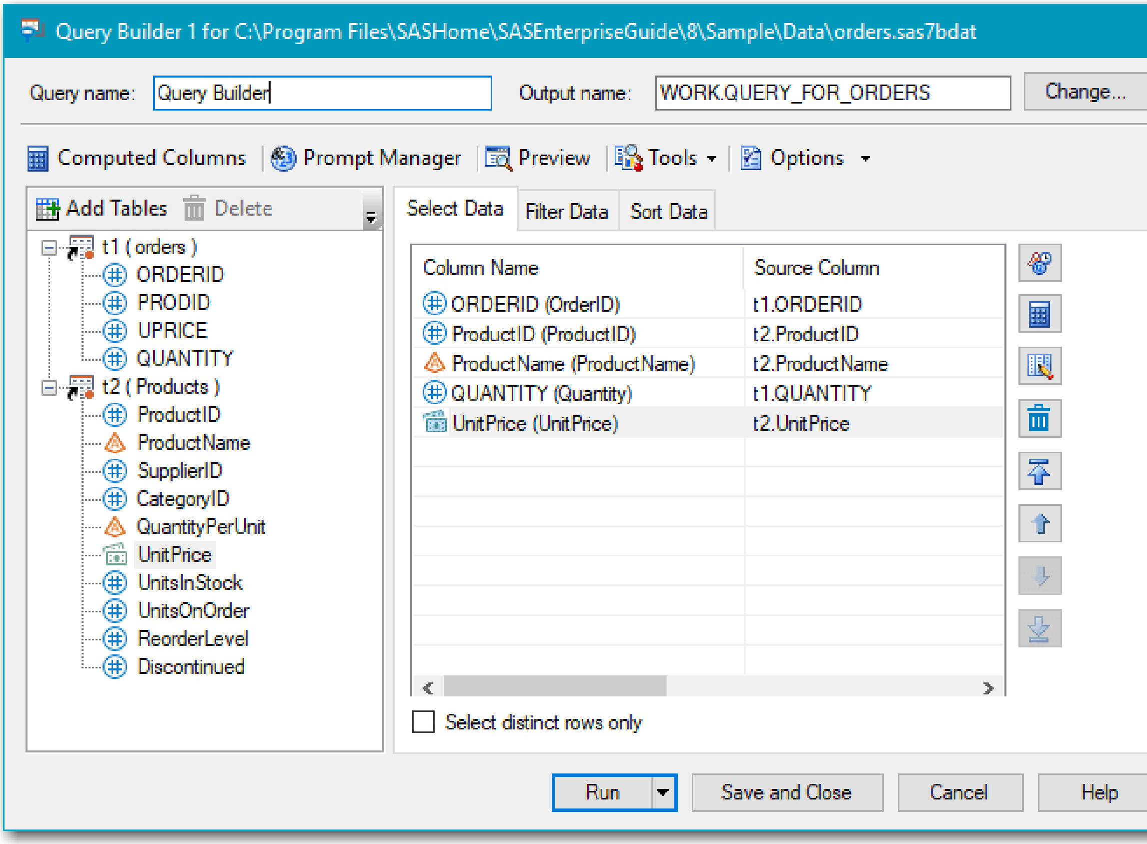
The order in which the columns are listed on the Select Data tab is the order in which they appear in the output table.

To select columns for the output table:

1 In the Query Builder, click the **Select Data** tab.

2 To add a column to the output table, double-click one or more columns that you want to add, and or right-click the columns and select **Select Column**.

To remove a column from the output table, select the column and click 🗑, or drag it back to the list of columns on the left.



Query Builder 1 for C:\Program Files\SASHome\SASEnterpriseGuide\8\Sample\Data\orders.sas7bdat

Query name: Query Builder

Output name: WORK.QUERY_FOR_ORDERS

t1 (orders): ORDERID, PRODID, UPRICE, QUANTITY

t2 (Products): ProductID, ProductName, SupplierID, CategoryID, QuantityPerUnit, UnitPrice, UnitsInStock, UnitsOnOrder, ReorderLevel, Discontinued

Column Name	Source Column
ORDERID (OrderID)	t1.ORDERID
ProductID (ProductID)	t2.ProductID
ProductName (ProductName)	t2.ProductName
QUANTITY (Quantity)	t1.QUANTITY
UnitPrice (UnitPrice)	t2.UnitPrice

☐ Select distinct rows only

Note: Even if a column is not selected for the output table, you can still use it to sort or group the output table.

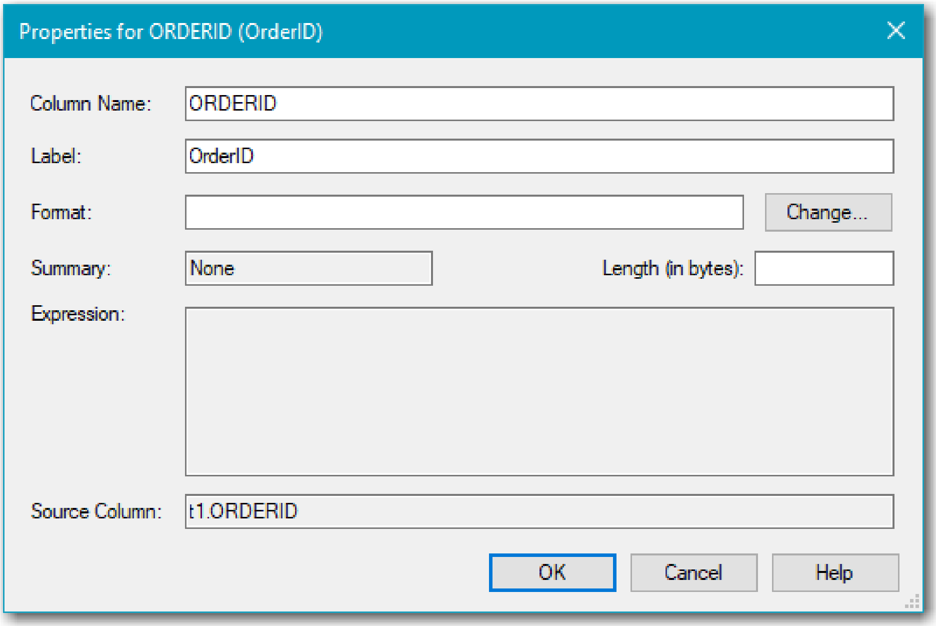
3 To change the order in which columns appear in the output table, select a column and click ↑ or ↓ to move it up or down. To move a column to the top of the list, click ⤒. To move a column to the bottom of the list, click ⤓.

Note: You can also drag a column up or down the list.

To specify an alias for a column:

1 In the Query Builder, click the **Select Data** tab.

2 Select the item for which you want to specify an alias and click ▦. The Properties for *column-name* window opens.



Properties for ORDERID (OrderID)

Column Name: ORDERID

Label: OrderID

Format:

Summary: None

Length (in bytes):

Expression:

Source Column: t1.ORDERID

3 In the **Column Name** box, type the desired alias name. The name will be displayed as the column heading in the output.

4 Click **OK** to save the alias.

Note: You can also specify an alias by single-clicking a column name twice in the **Column Name** column and entering the alias.

Creating a Computed Column

About Creating a Computed Column

You can insert a new column into your query that is computed from other columns or values. You can use a computed column to summarize values, replace values based on a condition, and perform string and numeric calculations. After you create a computed column, you can use it to sort or group your output.

Creating a Summarized Column

You can create a computed column by performing a summary function on any of the columns in your query. When you perform a summary function on a column, a new alias name is assigned to the column. You can use the Computed Column wizard to change the alias and format of the new computed column.

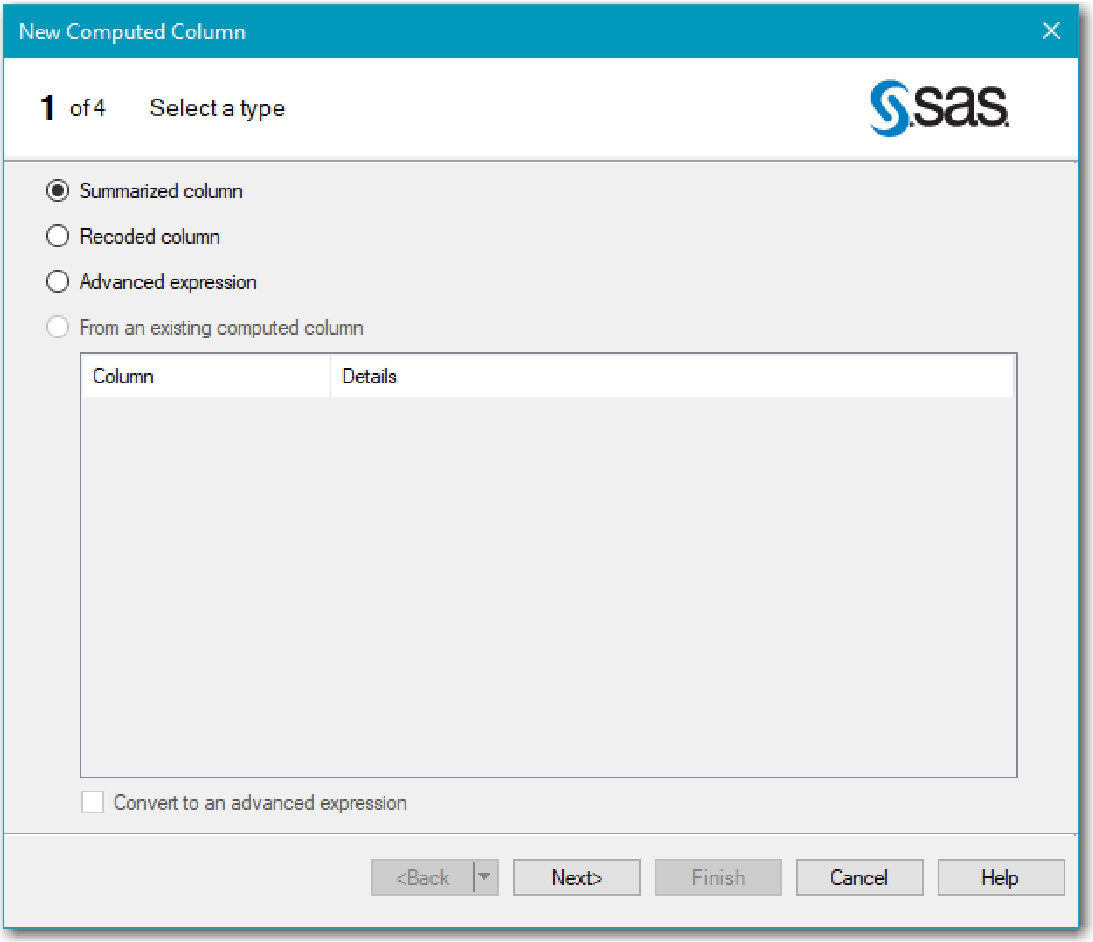
Note: You can also create a summarized column by selecting a summary function for the column on the Select Data tab. If you need to change the alias or format of the summarized column, then you need to edit the column properties separately. For more information, see "Using Summary Functions" on page 252.

To create a summarized column:

1 In the Query Builder, click **Computed Columns**. The Computed Columns window opens.

  Note: By default, the new column is included in the output table. If you do not want to include the new column, clear the **Add new computed columns to the query selection** check box.

2 Click **New**. The New Computed Column wizard opens.



New Computed Column

1 of 4 Select a type

● Summarized column

○ Recoded column

○ Advanced expression

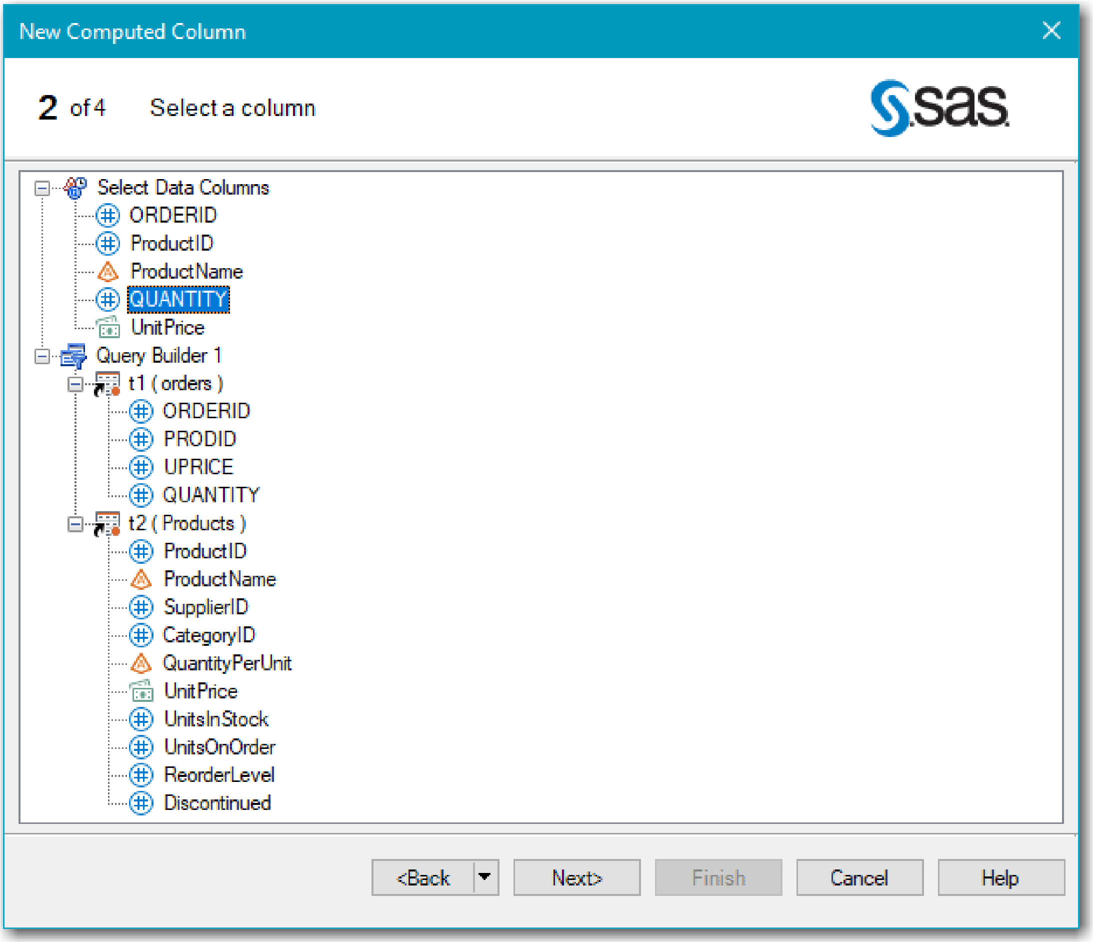
○ From an existing computed column

Column	Details

□ Convert to an advanced expression

Note: You can also open the New Computed Column wizard by clicking ▦ on the Select Data tab.

3 For the type of column, select **Summarized column**. Click **Next**.

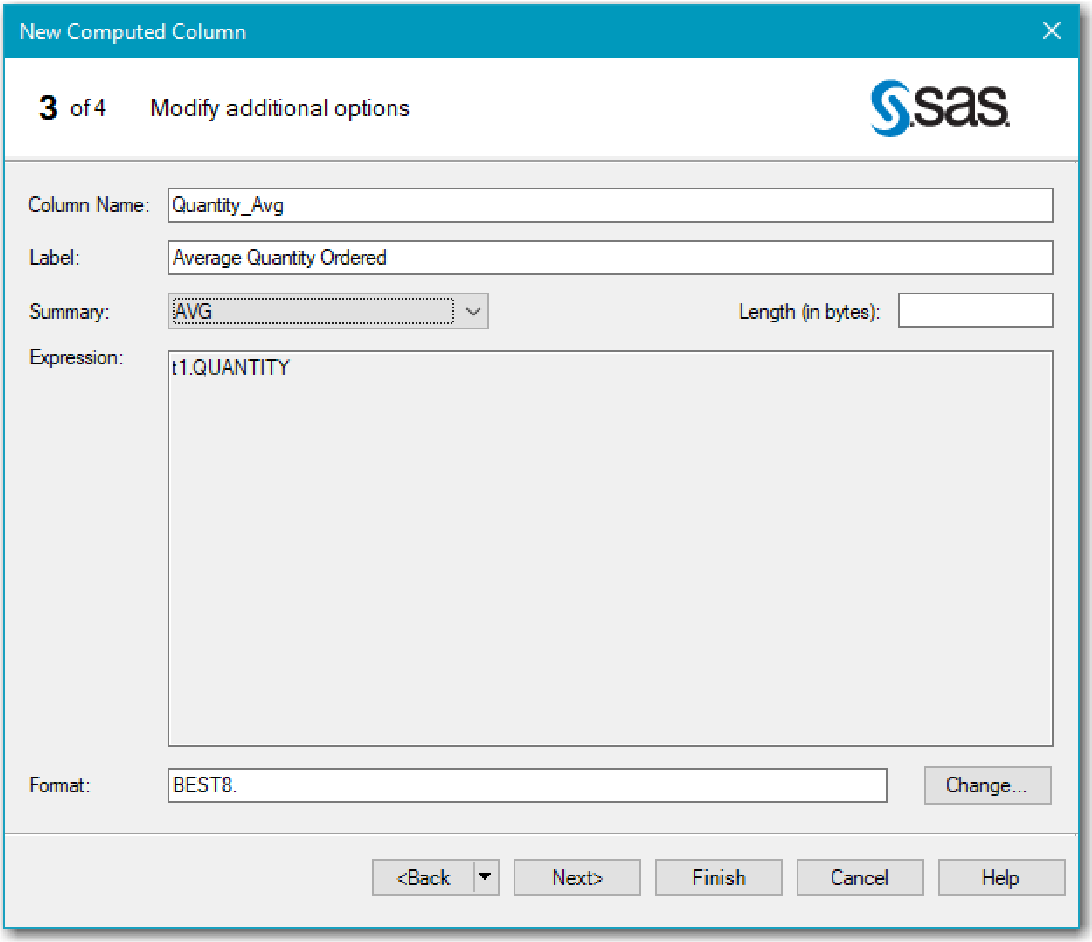


New Computed Column

2 of 4 Select a column

- Select Data Columns
  - ORDERID
  - ProductID
  - ProductName
  - QUANTITY
  - UnitPrice
- Query Builder 1
  - t1 (orders)
    - ORDERID
    - PRODID
    - UPRICE
    - QUANTITY
  - t2 (Products)
    - ProductID
    - ProductName
    - SupplierID
    - CategoryID
    - QuantityPerUnit
    - UnitPrice
    - UnitsInStock
    - UnitsOnOrder
    - ReorderLevel
    - Discontinued

4 From the list of tables in the project, expand the appropriate table and select the column on which you want to perform a summary function. Then click **Next**. By default, when you perform a summary function on a column, a new identifier is assigned to the column.



Column Name: Quantity_Avg

Label: Average Quantity Ordered

Summary: AVG

Length (in bytes):

Expression: t1.QUANTITY

Format: BEST8.

5 To change the default column name for the new column, enter the new column name in the **Column Name** box. The column name is displayed as the column heading in your output table.

6 To specify a label for the column, type the label in the **Label** box. Valid column labels cannot exceed 256 characters. The label is displayed as the column heading in your report output.

Note: You can specify that all spaces and special characters in column names and labels be converted to underscores. For more information, see "Setting Query Options" on page 500.

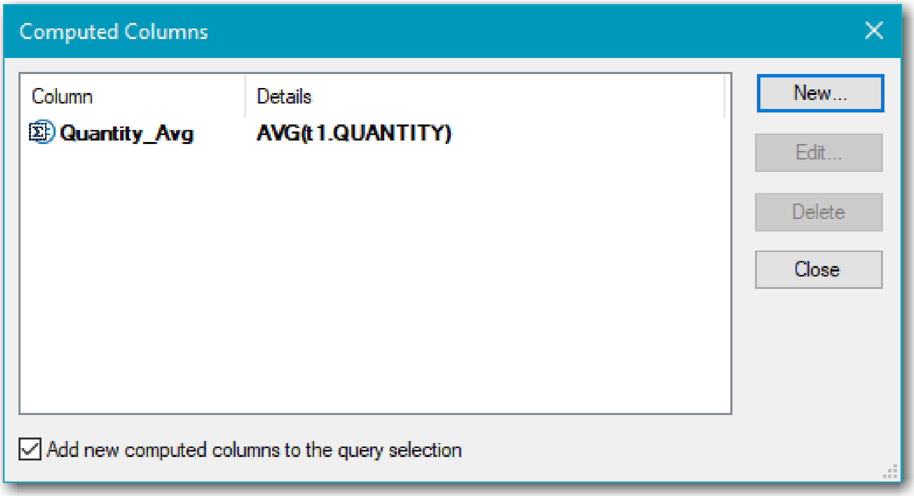
7 From the **Summary** drop-down list, select the function that you want to use.

8 To change the length of the new column, type a new length in bytes in the **Length** box. For numeric columns, the length must be a value between 3 and 8, inclusive. For character columns, the length must be a value between 1 and 32,767, inclusive.

9 To apply a format to the summarized column, click **Change** to open the Formats window and select the format and attributes that you want to use. Click **OK** to save the format changes. Then click **Next**.

10 Verify that the properties of the new column are correct. If you want to make any changes, click **Back** until you reach the page on which you want to make a correction.

11 Click **Finish** to create the new summarized column.



Column	Details
Quantity_Avg	AVG(t1.QUANTITY)

☑ Add new computed columns to the query selection

Creating a Recoded Column

You can create a recoded computed column by replacing individual values or ranges of values in a column with values that you specify. You can also choose to replace values that meet a condition that you specify.

- You can specify single values to be recoded as other values that you specify. For example, if your data contains a column named Gender with possible values of "M" and "F," then you can substitute "Male" for values of "M" and "Female" for values of "F" in your query results.
- You can substitute values for ranges of column values. For example, you could replace test scores from 90 to 100 with a letter grade of "A," scores from 80 to 89 with a grade of "B," and so on.
- You can create a conditional expression that enables you to use a comparison operator other than equality to replace column values in your results. For example, if your data contains a column named Age with possible values ranging from 1 to 99, then you can substitute values of "Under 35," "35 to 50," and "Over 50" for the numeric age values.

Note: You can also create a conditional expression or an expression to substitute values by creating an advanced expression computed column that uses the CASE function. For more information, see "Creating a Conditional Expression" on page 247.

To recode values:

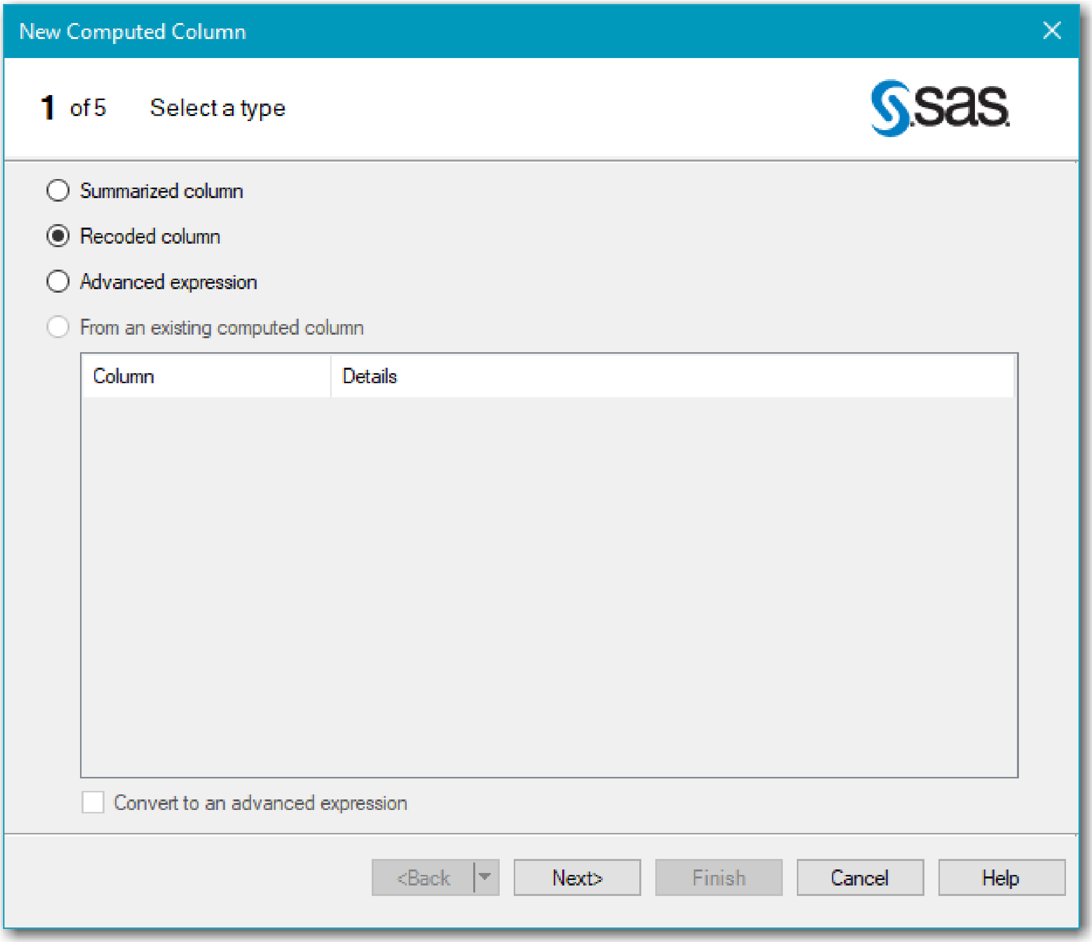
1 In the Query Builder, click **Computed Columns**. The Computed Columns window opens.

Note: By default, the new column is included in the output table. If you do not want to include the new column, clear the **Add new computed columns to the query selection** check box.

2 Click **New**. The **New Computed Column** wizard opens.

Note: You can also open the New Computed Column wizard by clicking ▦ on the **Select Data** tab.

3 For the type of column, select **Recoded column**. Click **Next**.



New Computed Column

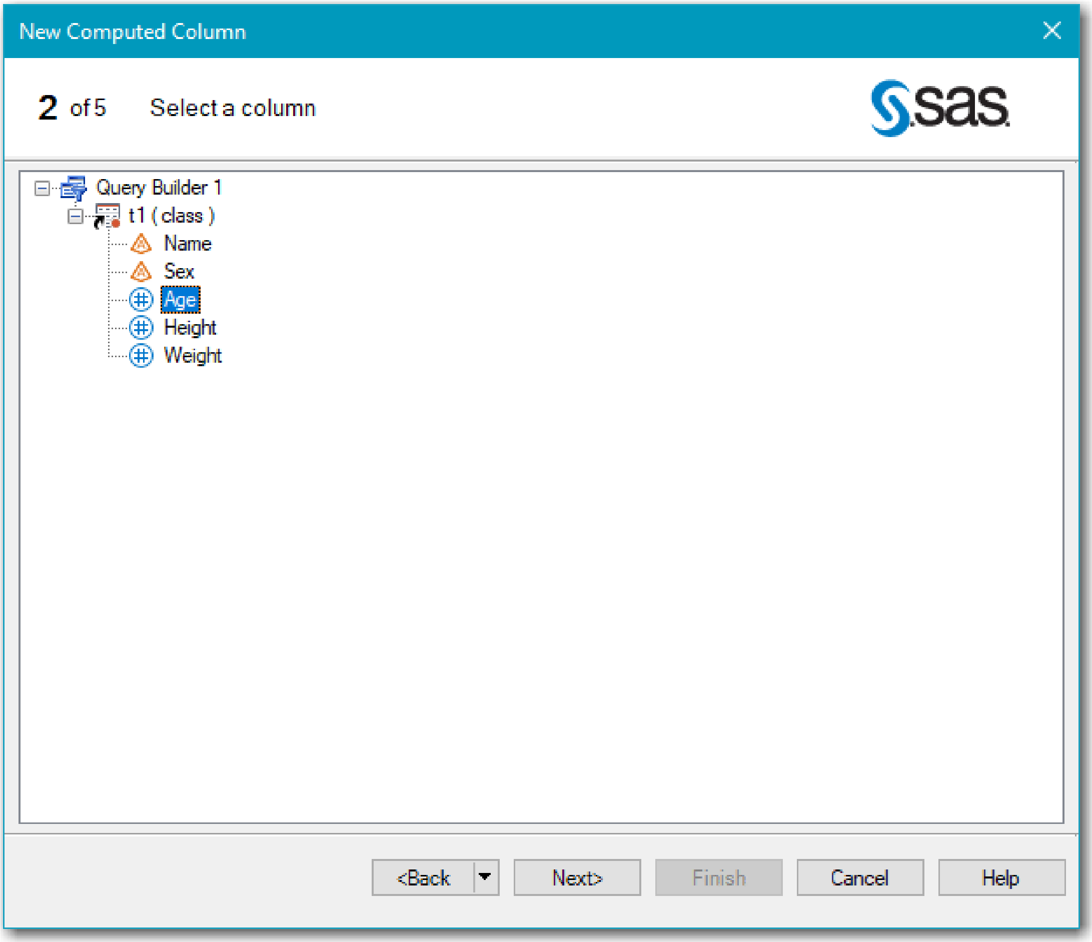
1 of 5 Select a type

○ Summarized column
● Recoded column
○ Advanced expression
○ From an existing computed column

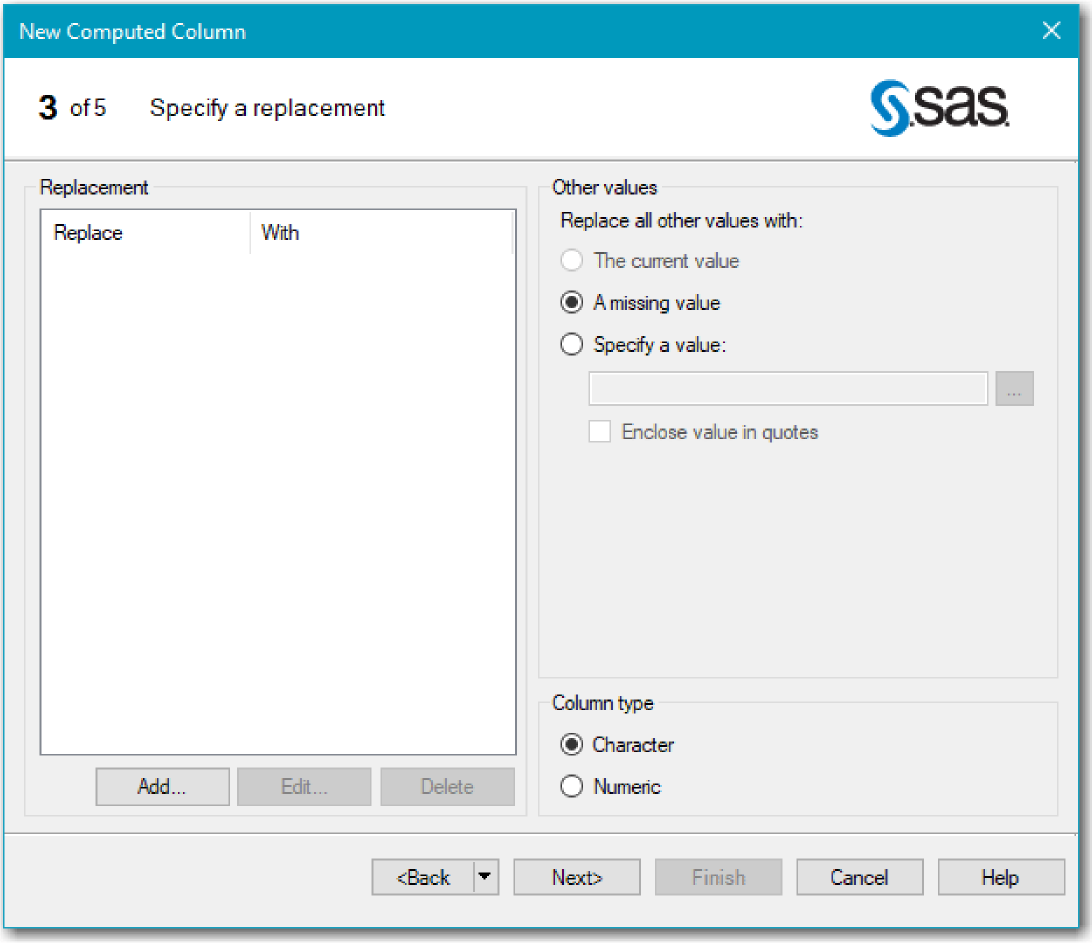
Column	Details

□ Convert to an advanced expression

4 From the list of tables that are in the query, expand the appropriate table and select the column whose values you want to replace. Click **Next**.



5 To specify replacement values for a column, click **Add**.



Other values

Replace all other values with:

○ The current value

● A missing value

○ Specify a value:

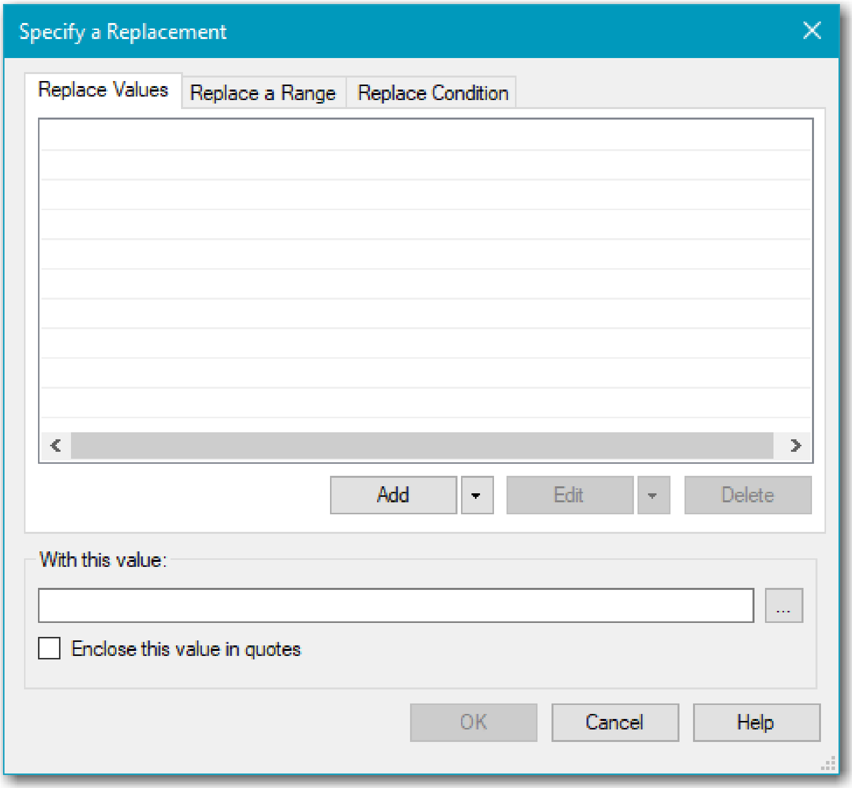
□ Enclose value in quotes

Column type

● Character

○ Numeric

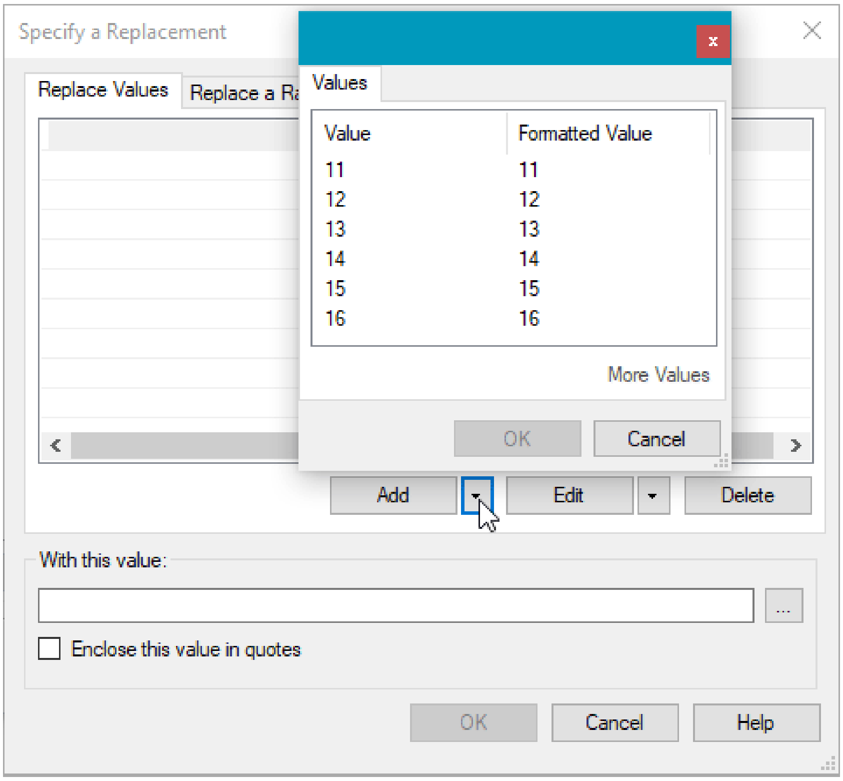
The Specify a Replacement window opens. You can choose whether you want to replace a single value, a range of values, or values that meet a specific condition.



To replace a single value:

a Click the **Replace Values** tab.

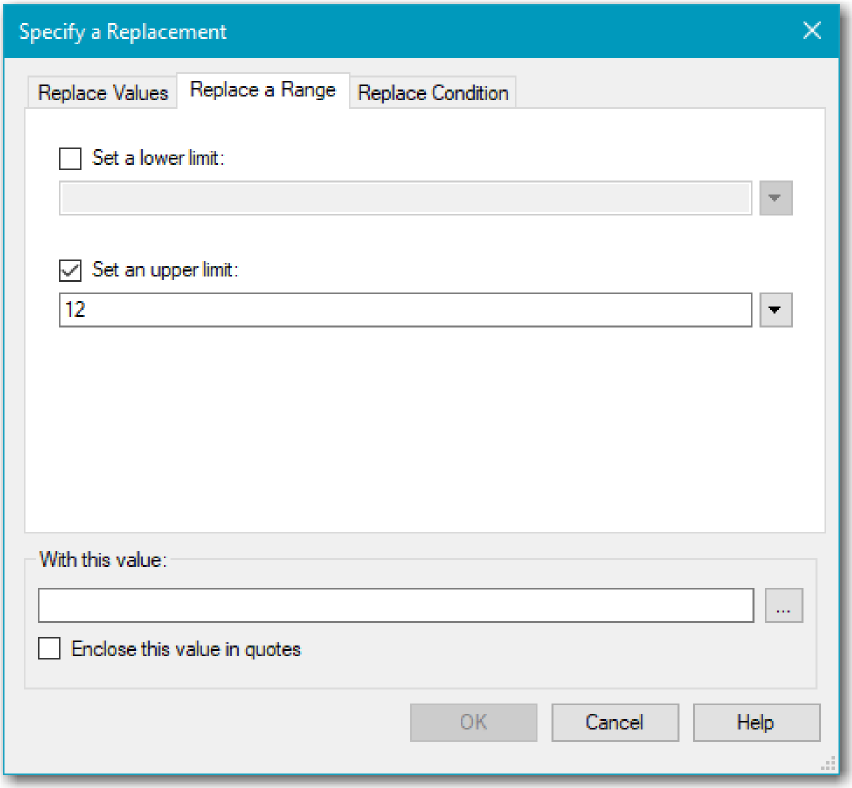
b Click **Add** and enter the value. To select a value from the list of values for the column, click ▾ next to the **Add** button. Select the value that you want to replace and click **OK**.



Note: If all of the values for that column are not displayed and you need to see more values, click **More Values**.

To replace a range of values:

a Click the **Replace a Range** tab.



☐ Set a lower limit:

☑ Set an upper limit:
12

With this value:

☐ Enclose this value in quotes

b If you want to set a lower limit, select the **Set a lower limit** check box. Then type the lower limit value or select a value from the drop-down list.

c If you want to set an upper limit, select the **Set an upper limit** check box. Then enter the upper limit value or select a value from the drop-down list.

Note: You must define a limit. You can define a lower limit, an upper limit, or a lower and an upper limit.

To specify a condition:

a Click the **Replace Condition** tab.

b Select a comparison operator from the **Operator** drop-down list. The default value is **Equal to**.

c If you are using a prompt value in the condition and you want to use the %_eg_WhereParam custom macro in the SQL code for the condition, then select **Generate filter for a prompt value**. If you are using the query in a stored process, then you must select this option. For more information, see "About Prompts" on page 372.

d In the **Value** box, you can enter the value that you want to compare to the column values. To select from a list of values, columns, prompts, or subqueries, click ▾ .

- On the **Values** tab to display a list of column values, click **Get Values**. The column values are displayed. If there are additional values that can be displayed, you can click **More Values**.

Note: For large tables, generating the list of available values can take several minutes. However, if you enter your own value instead of choosing one from the list, then the filter results might not be what you expect. You can use the Number of distinct column values to get for each request option to control how many values are displayed at one time. For more information, see "Setting Query Options" on page 500.

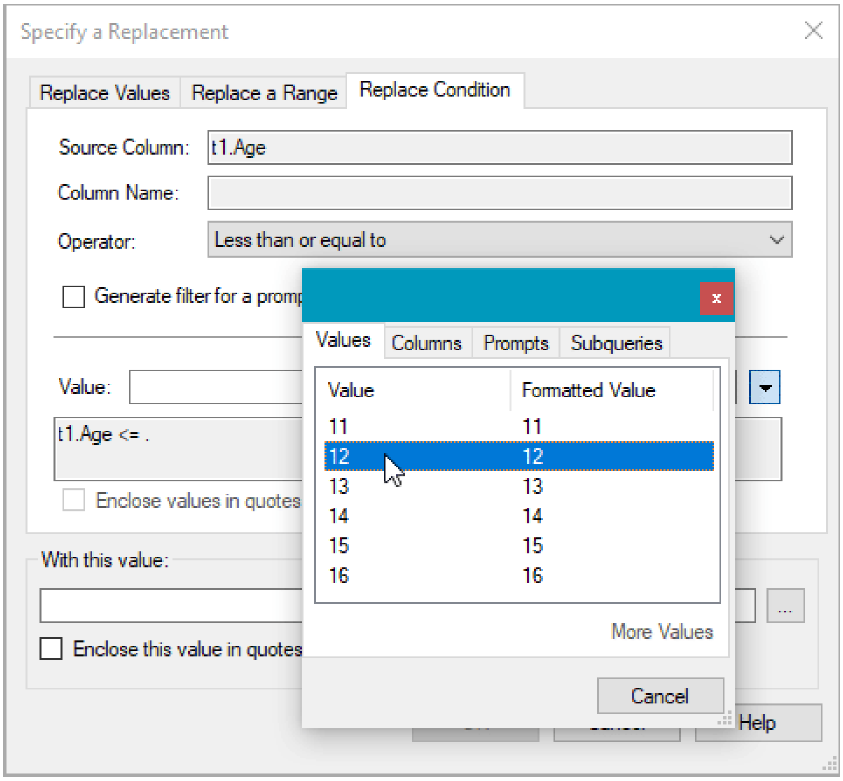
By default, the values are displayed with a common format. To display the values with a SAS format, select **SAS Format** from the **Show** drop-down list.

Note: If the column is defined as a currency, time, date, or datetime format, these common formats are used:

- Currency - $10.00
- Time – 2:29:09 PM
- Date – Monday, April 17, 2019
- DateTime – Monday, April 17, 2019 2:29:09 PM

- Click the **Columns** tab to select the column that contains the values that you want to use.
- Click the **Prompts** tab to select the prompt that you want to use.
- Click the **Subqueries** tab to select a template for a subquery. For more information, see "About Task Templates for Queries" on page 267. Verify that the data references that are listed are still valid. If you need to change a data reference, click **Change** and select another data file.

Note: If SAS Enterprise Guide detects a problem with the data references in the query, the ⚠ is displayed by the data. You must verify that the data exists before the OK button is enabled.



Source Column: t1.Age

Column Name:

Operator: Less than or equal to

□ Generate filter for a promp

Value:

t1.Age <= .

□ Enclose values in quotes

With this value:

□ Enclose this value in quotes

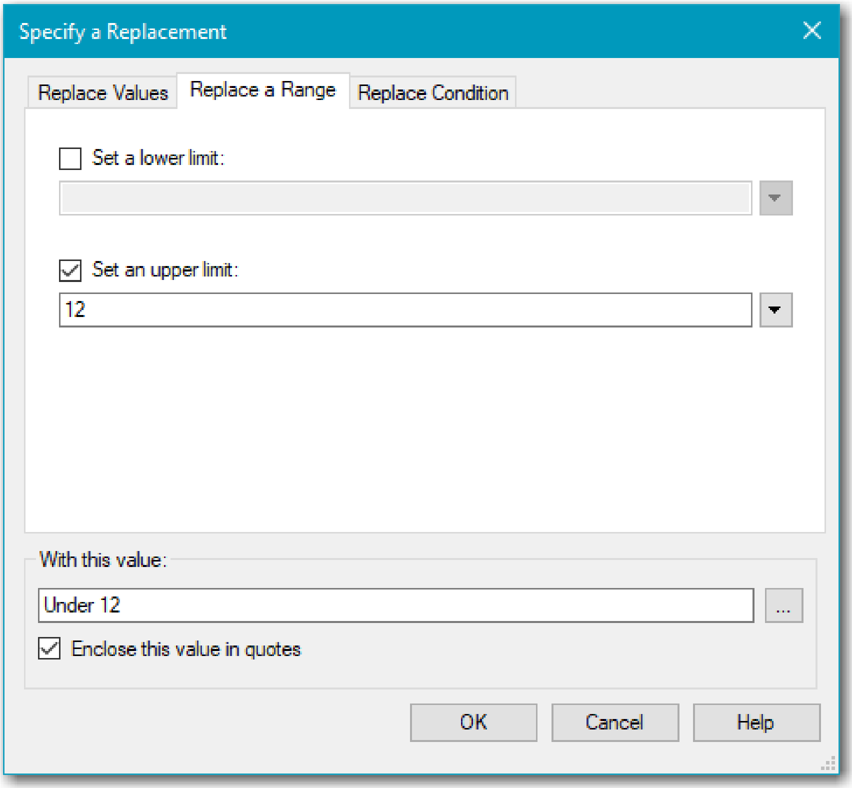
Value	Formatted Value
11	11
12	12
13	13
14	14
15	15
16	16

e If you want the value enclosed in quotation marks, make sure that the **Enclose this value in quotes** check box is selected.

f If you want to use unformatted dates in the comparison expression, clear the **Use formatted dates** check box. If you select this option, the formatted values are used in the expression. It is more efficient to use unformatted date values than formatted date values in the comparison expression.

Note: This option is available only if the column that you have selected uses a date, time, or datetime format.

6 In the **With this value** box, type the value that you want to assign to the column and click **OK** to add the replacement expression to the list.



Specify a Replacement

☐ Set a lower limit:

☑ Set an upper limit:

12

With this value:

Under 12

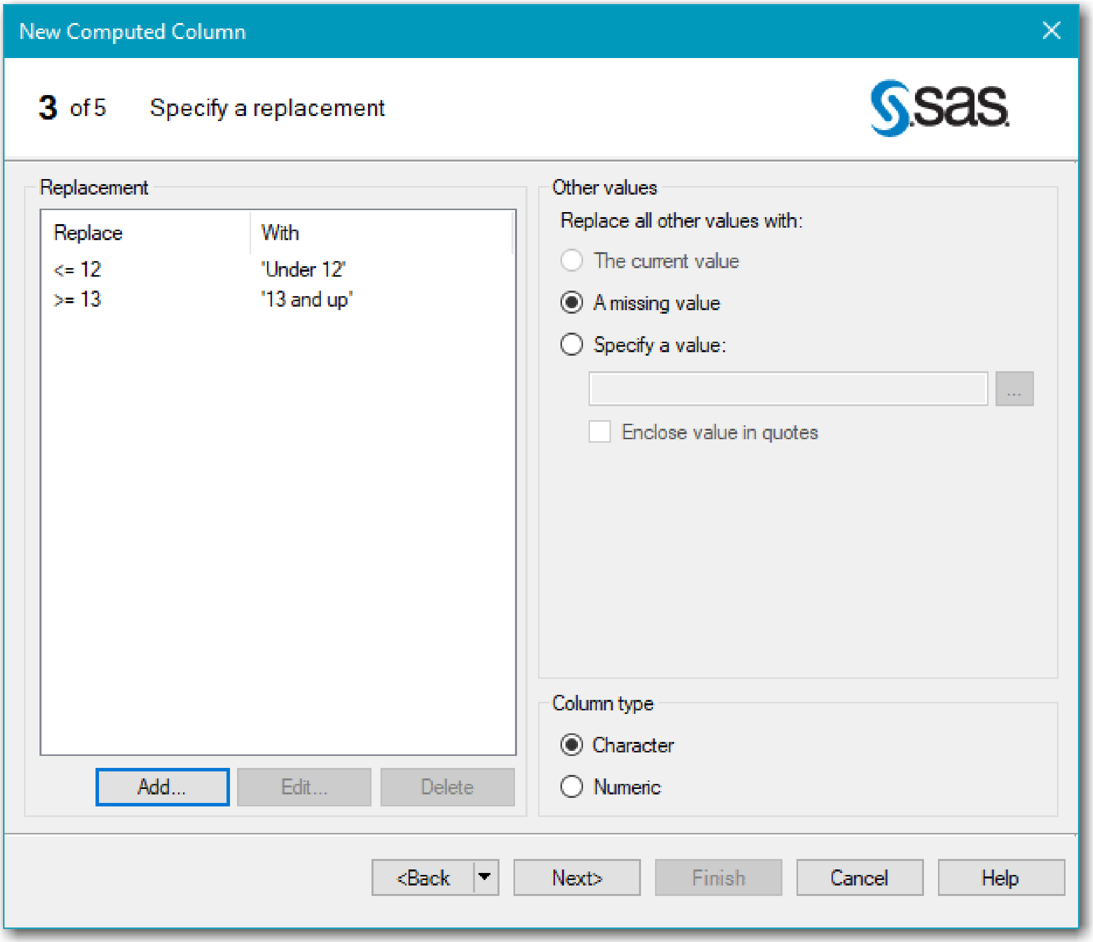
☑ Enclose this value in quotes

Note: If you want to delete a replacement expression after you add it, click the expression that you want to delete and click **Delete**.

7 If you want the value to be enclosed in quotation marks, select the **Enclose this value in quotes** check box.

8 You can repeat steps 5–7 to add more replacement expressions.

9 In the Other values area, select the type of value with which you want to replace all other values that are not included in the list of replacements.



Replacement

Replace	With
<= 12	'Under 12'
>= 13	'13 and up'

Other values

Replace all other values with:

○ The current value

● A missing value

○ Specify a value:

□ Enclose value in quotes

Column type

● Character

○ Numeric

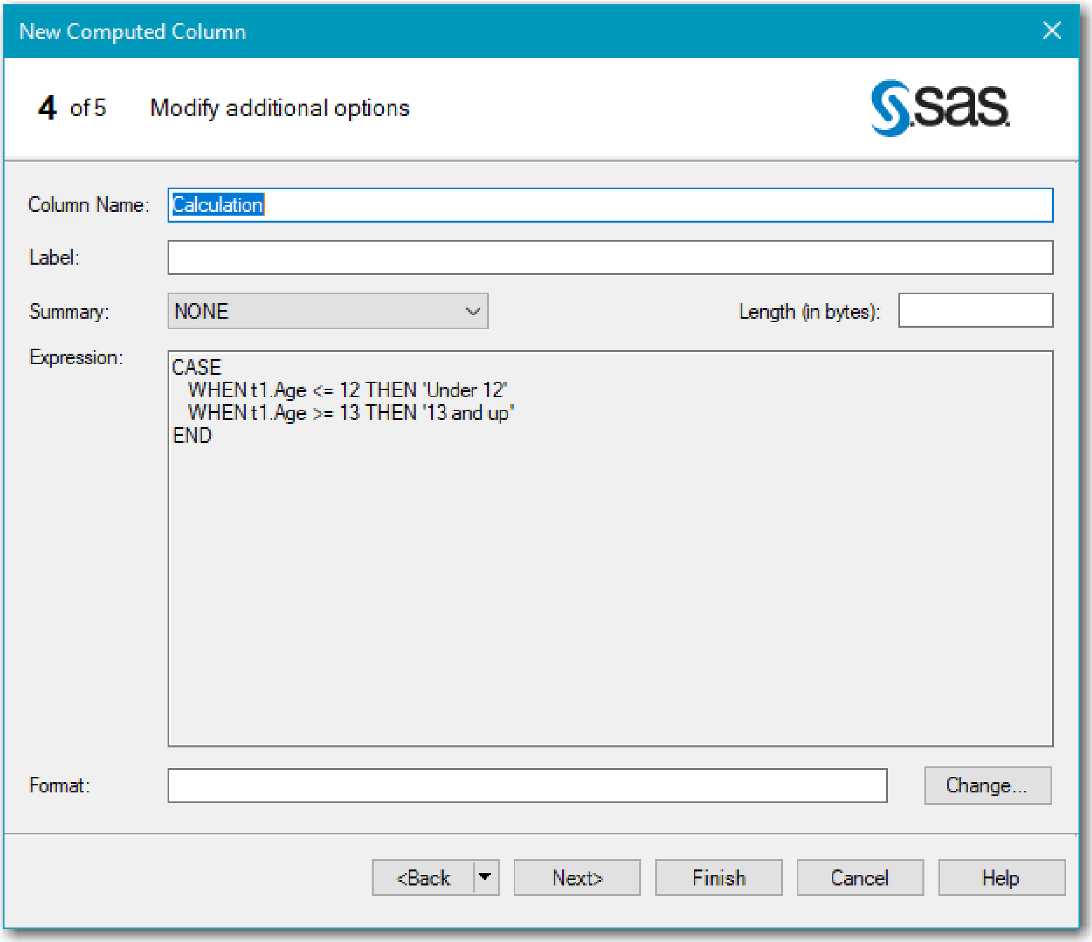
- **The current value** – uses the current value of each column.

  Note: If the column type of the new computed column is not the same as the column type of the original column, then the **The current value** option is disabled.

- **A missing value** – replaces the values with a missing value.
- **Specify a value** – replaces the values with a new value that you enter in the value box.

10 To change the data type of the new column, select the data type from the **Column type** area.

11 After you create all of the replacement expressions that you want, click **Next**.



Column Name: Calculation

Label:

Summary: NONE

Length (in bytes):

Expression:

```
CASE
  WHEN t1.Age <= 12 THEN 'Under 12'
  WHEN t1.Age >= 13 THEN '13 and up'
END
```

Format:

12 To change the default column name for the new column, enter the new column name in the **Column Name** box. The column name is displayed as the column heading in your output table.

13 To specify a label for the column, type the label in the **Label** box. Valid column labels cannot exceed 256 characters. The label is displayed as the column heading in your report output.

Note: You can specify that all spaces and special characters in column names and labels be converted to underscores. For more information, see "Setting Query Options" on page 500.

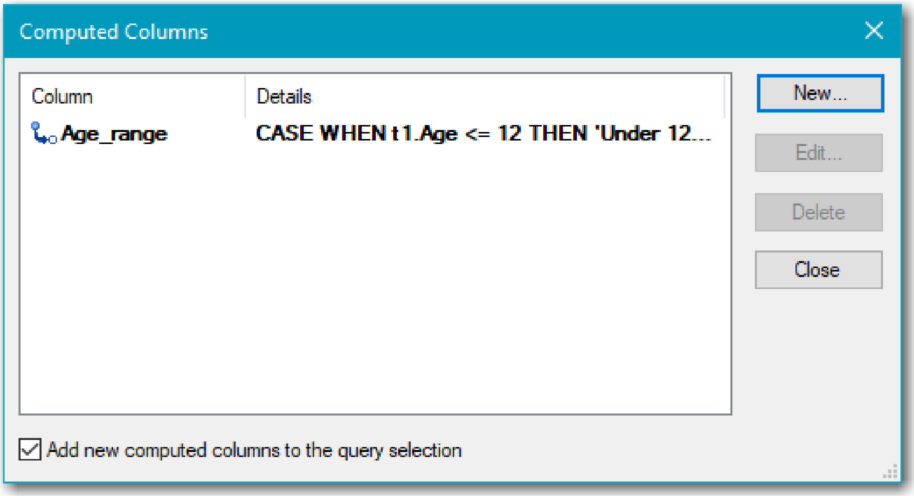
14 If you want to apply a summary function to the recoded column, select the function from the **Summary** drop-down list.

15 To change the length of the new column, type a new length in bytes in the **Length** box. For numeric columns, the length must be a value between 3 and 8, inclusive. For character columns, the length must be a value between 1 and 32,767, inclusive.

16 To apply a format to the recoded column, click **Change** to open the Formats window, and then select the format and attributes that you want to use. Only formats that are appropriate for the column type that you selected are displayed. Click **OK** to save the format changes. Then click **Next**.

17 Verify that the properties of the new column are correct. If you want to make any changes, click **Back** until you reach the page on which you want to make a correction.

18 Click **Finish** to create the new recoded column. The new column is added to the list of computed columns.



Computed Columns

Column	Details
Age_range	CASE WHEN t1.Age <= 12 THEN 'Under 12...

☑ Add new computed columns to the query selection

Creating a Computed Column with an Advanced Expression

You can use the advanced expression editor to create a computed column from other columns or values.

To create a computed column with an advanced expression:

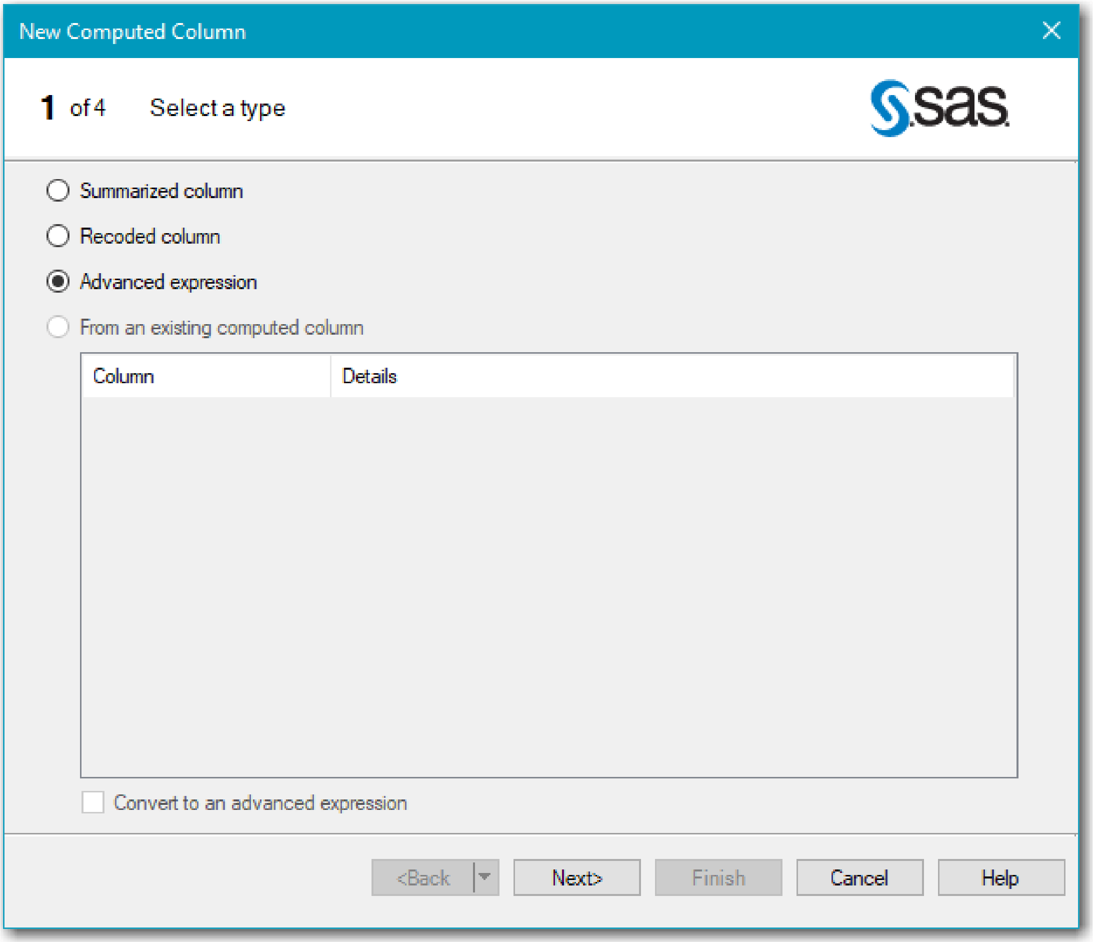
1 In the Query Builder, click **Computed Columns**. The Computed Columns window opens.

> **Note:** By default, the new column is included in the output table. If you do not want to include the new column, clear the **Add new computed columns to the query selection** check box.

2 Click **New**. The New Computed Column wizard opens.

> **Note:** You can also open the New Computed Column wizard by clicking ▦ on the **Select Data** tab.

3 For the type of column, select **Advanced expression**. Click **Next**.



New Computed Column

1 of 4 Select a type

○ Summarized column

○ Recoded column

● Advanced expression

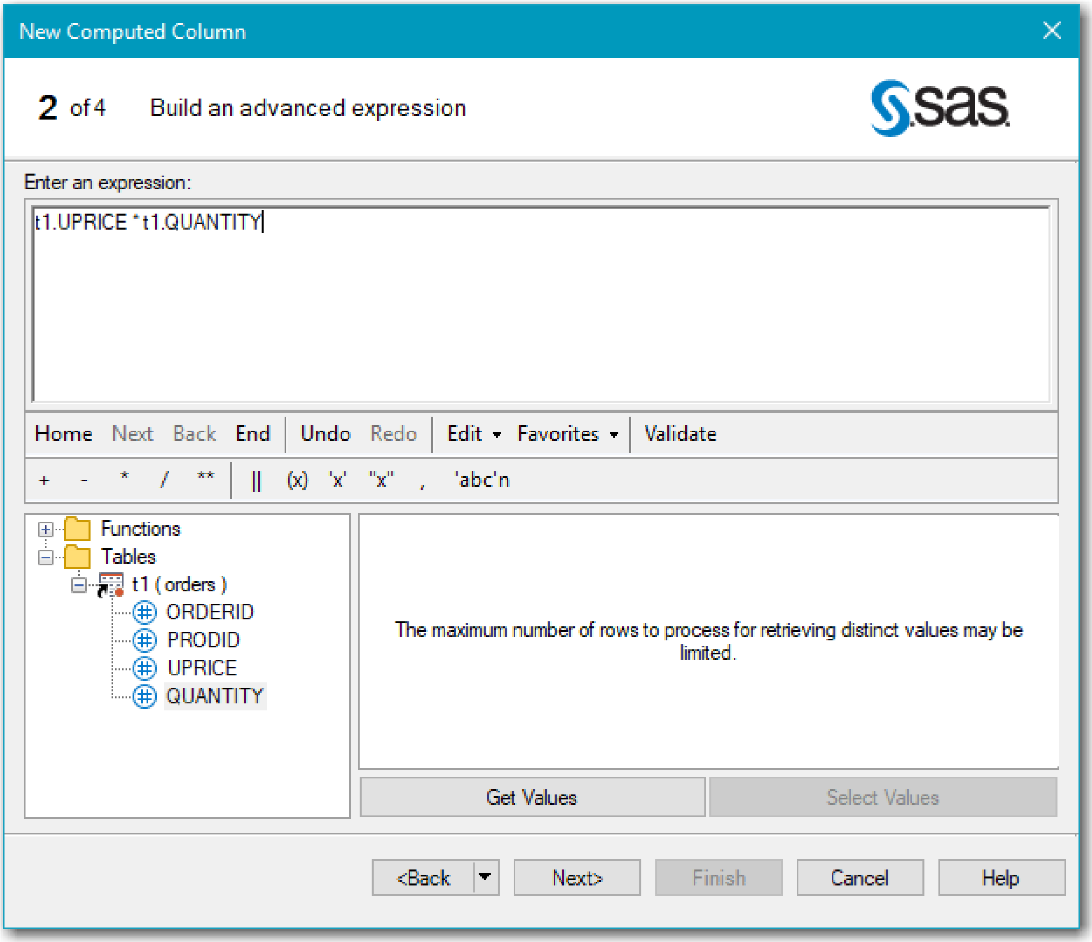
○ From an existing computed column

Column	Details

□ Convert to an advanced expression

4 You can build your expression by using mathematical operators, functions, columns, and constant values. For a list of all SAS functions, see *SAS Functions and CALL Routines: Reference*.

The constant value can either be a number (such as 10 or 3.14) or a string (such as 'NC' or 'Sales Numbers'). You can type these elements into the expression box at the top of the window, or you can use the expression editor to add these elements. For more information, see "Creating an Advanced Expression" on page 283.



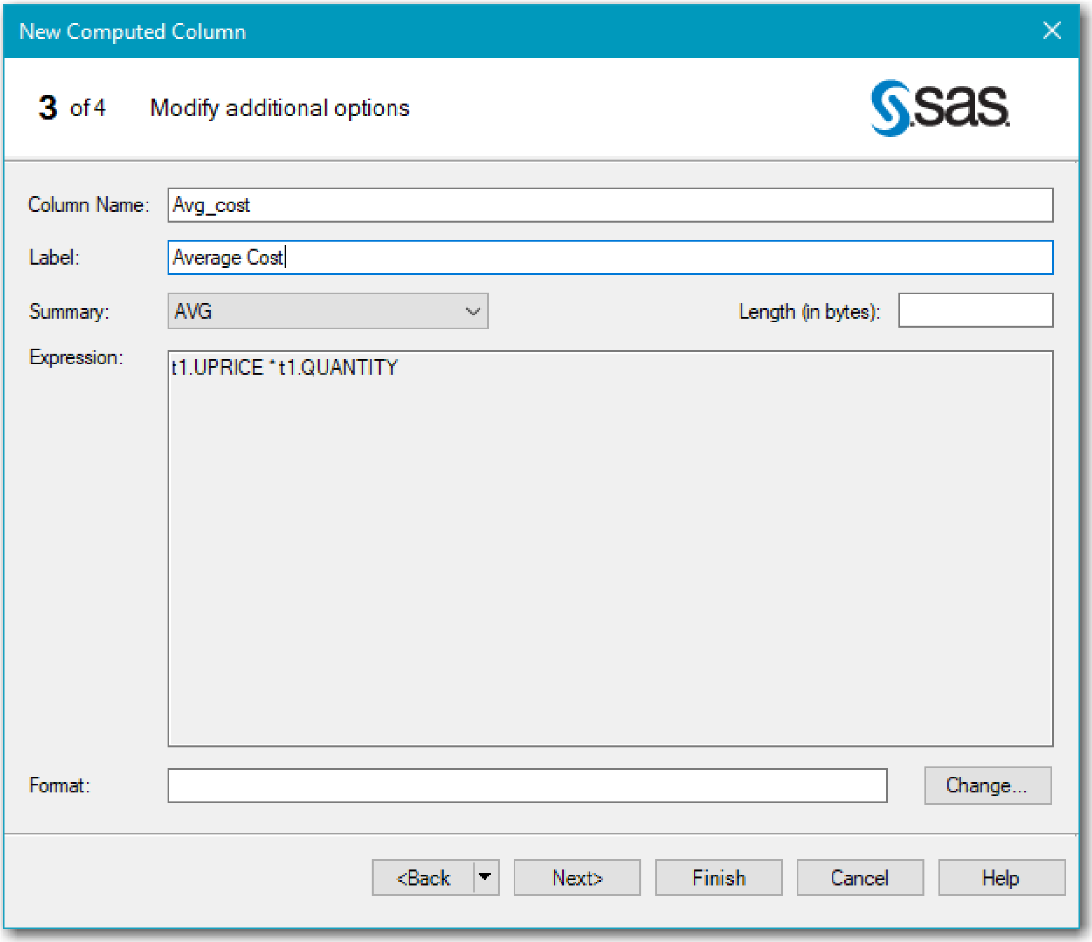
Enter an expression:

```
t1.UPRICE * t1.QUANTITY
```

The maximum number of rows to process for retrieving distinct values may be limited.

Note: When you select the **Use labels for column names** option to display the columns by label, an additional expression box opens to display the expression using the labels. However, you can type only in the expression box that uses column names. For more information, see "Setting General Data Options" on page 497.

5 After you have finished building the expression, click **Next**.



Column Name: Avg_cost

Label: Average Cost

Summary: AVG

Length (in bytes):

Expression: t1.UPRICE * t1.QUANTITY

Format:

6 To change the default column name for the computed column, enter the new name in the **Column Name** box. The column name is displayed as the column heading in your output table.

7 To specify a label for the column, type the label in the **Label** box. Valid column labels cannot exceed 256 characters. The label is displayed as the column heading in your report output.

Note: You can specify that all spaces and special characters in column names and labels be converted to underscores. For more information, see "Setting Query Options" on page 500.

8 If you want to apply a summary function to the computed column, select the function from the **Summary** drop-down list.

9 To change the length of the new column, type a new length in bytes in the **Length** box. For numeric columns, the length must be a value between 3 and 8, inclusive. For character columns, the length must be a value between 1 and 32,767, inclusive.

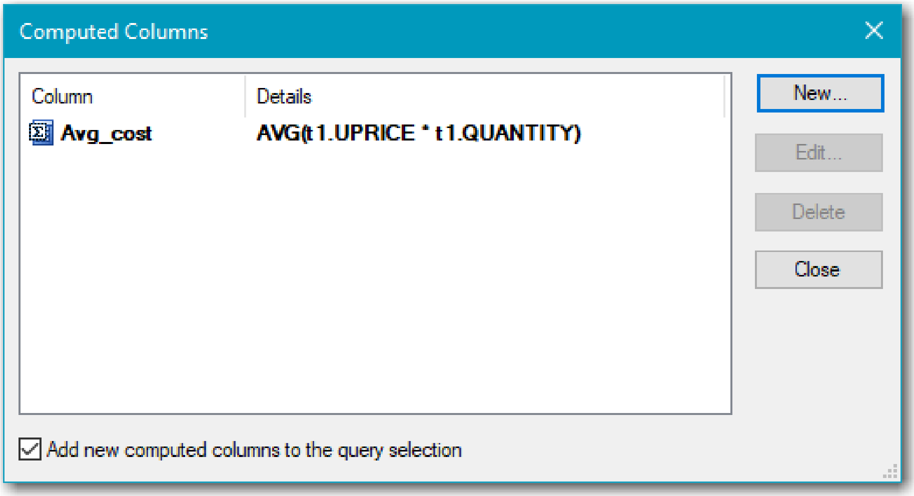
10 To apply a format to the column, click **Change** to open the Formats window, and then select the format and attributes that you want to use. Click **OK** to save the format changes. Then click **Next**.

For a list of all the SAS formats and informats, see *SAS Formats and Informats: Reference*.

Note: The type of format that you select determines the data type of the column. If no format is selected or the format value is **None**, then the data type of the column is numeric, by default.

11 Verify that the properties of the new column are correct. If you want to make any changes, click **Back** until you reach the page on which you want to make a correction.

12 Click **Finish** to create the new computed column. The new column is added to the list of computed columns.



Column	Details
Avg_cost	AVG(t1.UPRICE * t1.QUANTITY)

☑ Add new computed columns to the query selection

Creating a Computed Column from an Existing Computed Column

You can use an existing computed column as the basis for creating a computed column. You can choose to create the column by using the interface that was used to create the existing column, or you can convert the existing column to an advanced expression and edit it by using the expression builder.

For example, when you create a recoded column, the Query Builder automatically generates SQL code with the CASE function. If you want to create a computed column that is based on an existing recoded column, you can choose to create the new column by using the recoded column interface or by editing the CASE statement in the expression builder.

To create a column from an existing computed column:

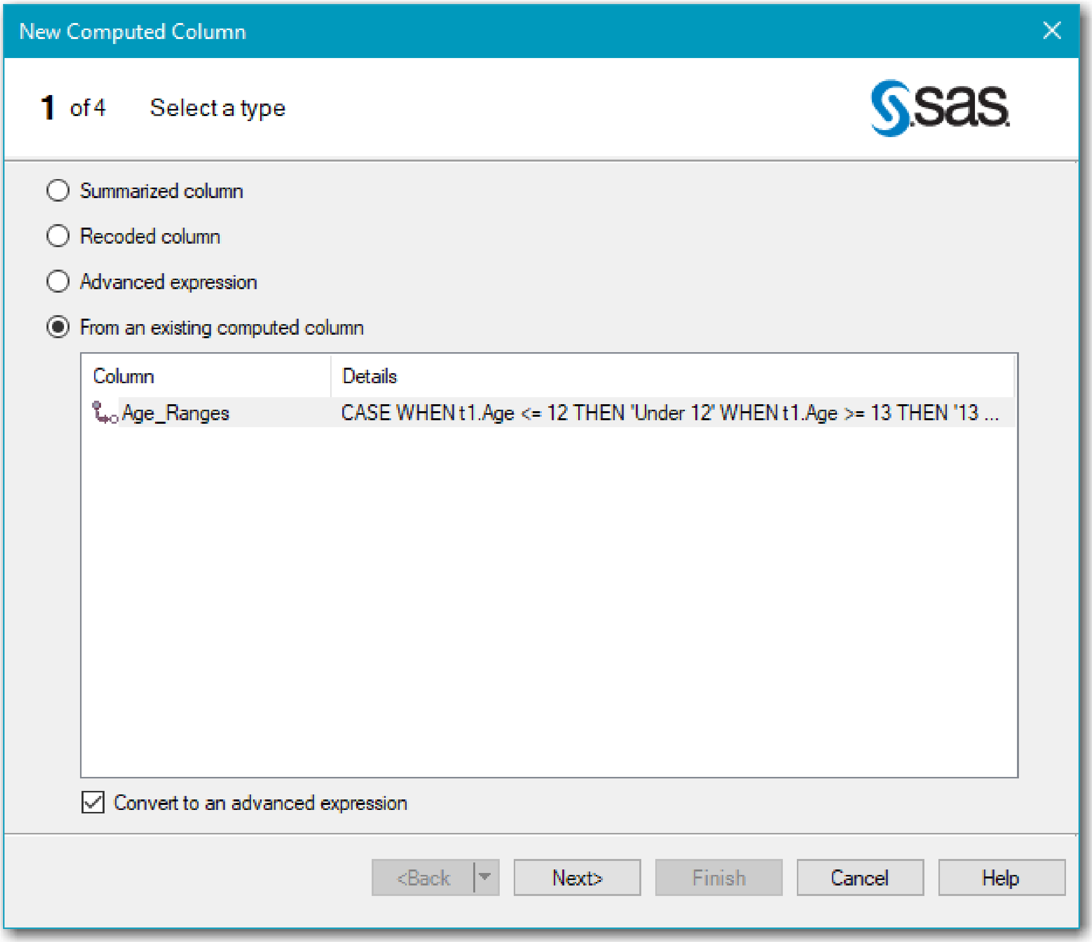
1 In the Query Builder, click **Computed Columns**. The Computed Columns window opens.

Note: By default, the new column is included in the output table. If you do not want to include the new column, clear the **Add new computed columns to the query selection** check box.

2 Click **New**. The New Computed Column wizard opens.

Note: You can also open the New Computed Column wizard by clicking ▦ on the **Select Data** tab.

3 For the type of column, select **From an existing computed column**.



New Computed Column

1 of 4 Select a type

○ Summarized column

○ Recoded column

○ Advanced expression

● From an existing computed column

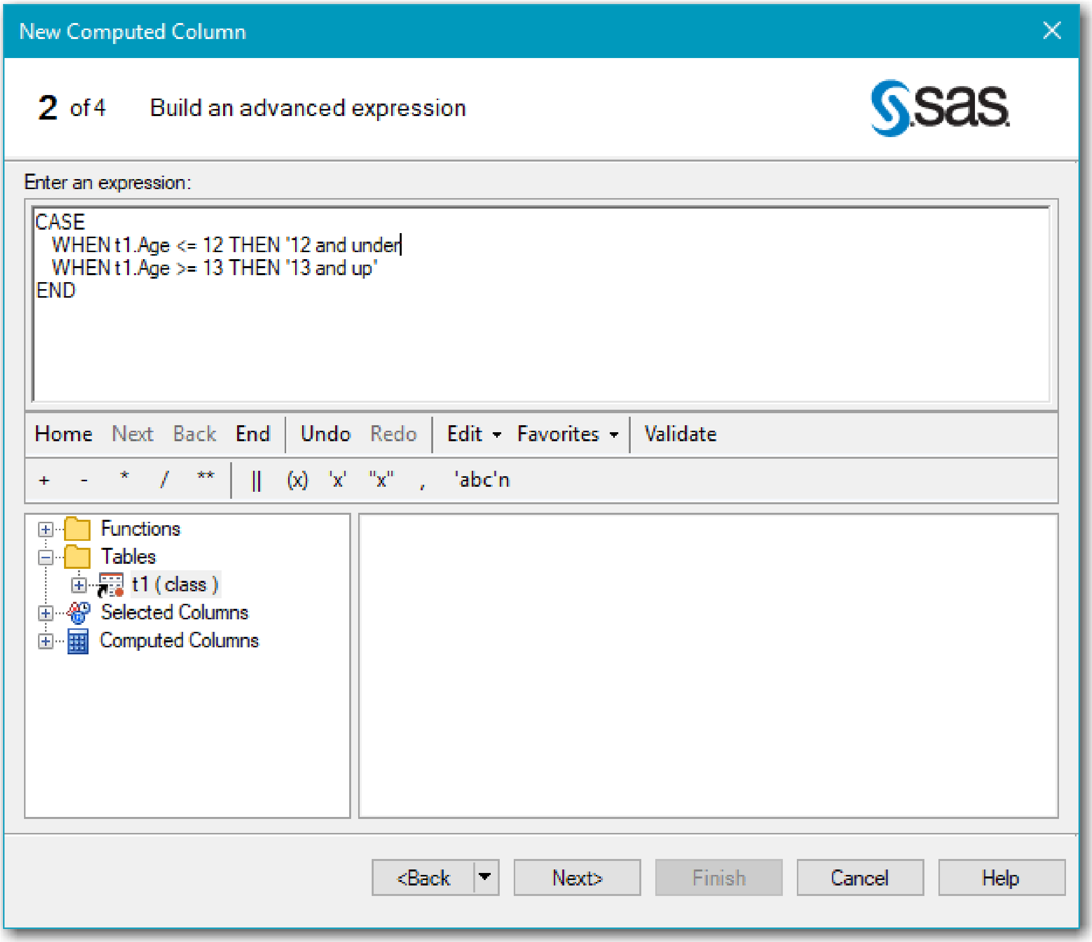
Column	Details
Age_Ranges	CASE WHEN t1.Age <= 12 THEN 'Under 12' WHEN t1.Age >= 13 THEN '13 ...

☑ Convert to an advanced expression

Note: If you do not have an existing computed column, then this option is disabled.

4 From the list of computed columns, select the column on which you want to base the new column.

5 If you want to convert the existing column to an advanced expression and edit it by using the expression editor, then select the **Convert to an advanced expression** check box. For more information, see "Creating an Advanced Expression" on page 283.


New Computed Column
2 of 4 Build an advanced expression
sas
Home Next Back End Undo Redo Edit Favorites Validate
+ - * / ** || (x) 'x' "x" , 'abc'n
Functions
Tables
t1 (class)
Selected Columns
Computed Columns
<Back
Next>
Finish
Cancel
Help

Enter an expression:

```
CASE
   WHEN t1.Age <= 12 THEN '12 and under
   WHEN t1.Age >= 13 THEN '13 and up'
END
```

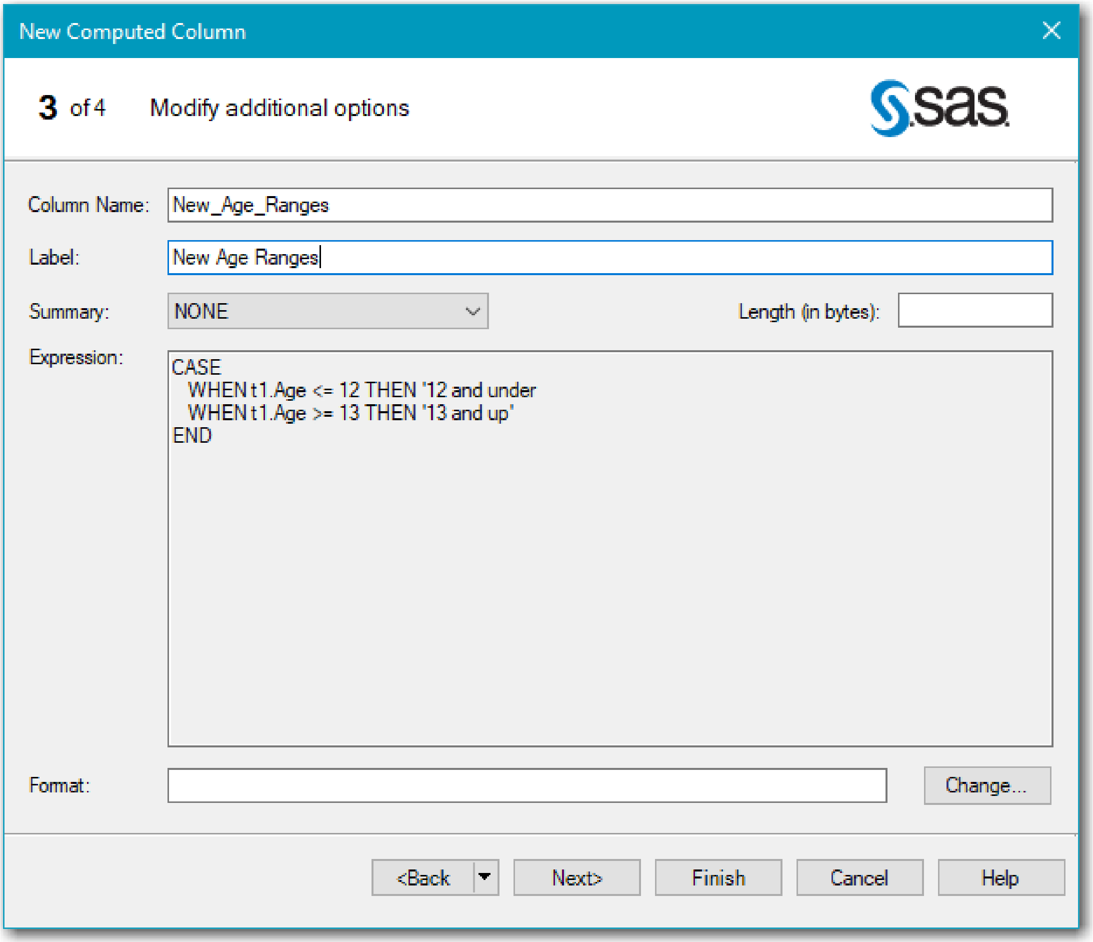
Note: Summary functions that were added to the computed column by using the **Summary** drop-down list on the additional options page are not included in the converted advanced expression. You can add or remove summary functions on the additional options page.

To create the column by using the interface that was used to create the existing column, clear the **Convert to an advanced expression** check box.

6 Click **Next** and edit the column as desired.

Note: For more information, see "Creating a Recoded Column" on page 227.

7 Click **Next** to specify additional options.



Column Name: New_Age_Ranges

Label: New Age Ranges

Summary: NONE

Length (in bytes):

Expression:

```
CASE
   WHEN t1.Age <= 12 THEN '12 and under
   WHEN t1.Age >= 13 THEN '13 and up'
END
```

Format:

Note: If you are creating the new column from an existing summarized column and you have not converted the column to an advanced expression, then you need to click **Next** only once to advance to the additional options.

---

8 To change the default column name for the new column, enter the new column name in the **Column Name** box. The column name is displayed as the column heading in your output table.

9 To specify a label for the column, type the label in the **Label** box. Valid column labels cannot exceed 256 characters. The label is displayed as the column heading in your report output.

Note: You can specify that all spaces and special characters in column names and labels be converted to underscores. For more information, see “Setting Query Options” on page 500.

---

10 If you want to apply a summary function to the new column, select the function from the **Summary** drop-down list.

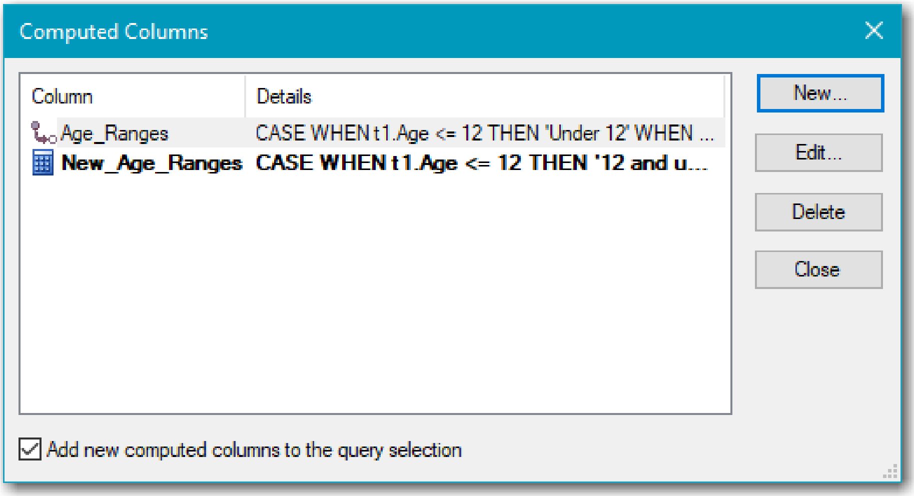
11 To change the length of the new column, type a new length in bytes in the **Length** box. For numeric columns, the length must be a value between 3 and 8, inclusive. For character columns, the length must be a value between 1 and 32,767, inclusive.

12 To apply a format to the new column, click **Change** to open the Formats window, and then select the format and attributes that you want to use. Click **OK** to save the format changes. Then click **Next**.

For a list of all the SAS formats and informats, see *SAS Formats and Informats: Reference*.

13 Verify that the properties of the new column are correct. If you want to make any changes, click **Back** until you reach the page on which you want to make a correction.

14 Click **Finish** to create the new column. The new column is added to the list of computed columns.



Computed Columns

Column	Details
Age_Ranges	CASE WHEN t1.Age <= 12 THEN 'Under 12' WHEN ...
New_Age_Ranges	**CASE WHEN t1.Age <= 12 THEN '12 and u...**

☑ Add new computed columns to the query selection

Working with Conditional Expressions

Creating a Conditional Expression

You can create a conditional expression in the Query Builder by creating a computed column that uses the CASE expression. A conditional expression enables you to use logic similar to "IF-THEN/ELSE" to return values that can be used to create a column.

Note: In some cases, it might be easier to recode the column values if you are replacing single values or a range of values with new values that you specify or replacing values that meet a condition that you specify. However, if you want to use an expression to substitute values, then you must create a conditional expression in the advanced expression editor. For more information, see "Creating a Recoded Column" on page 227.

For example, you could create an expression to extract the first letter of a last name, and then replace last names starting with "A" through "F" with the value "A - F," replace last names starting with "G" through "L" with the value "G - L," and so on.

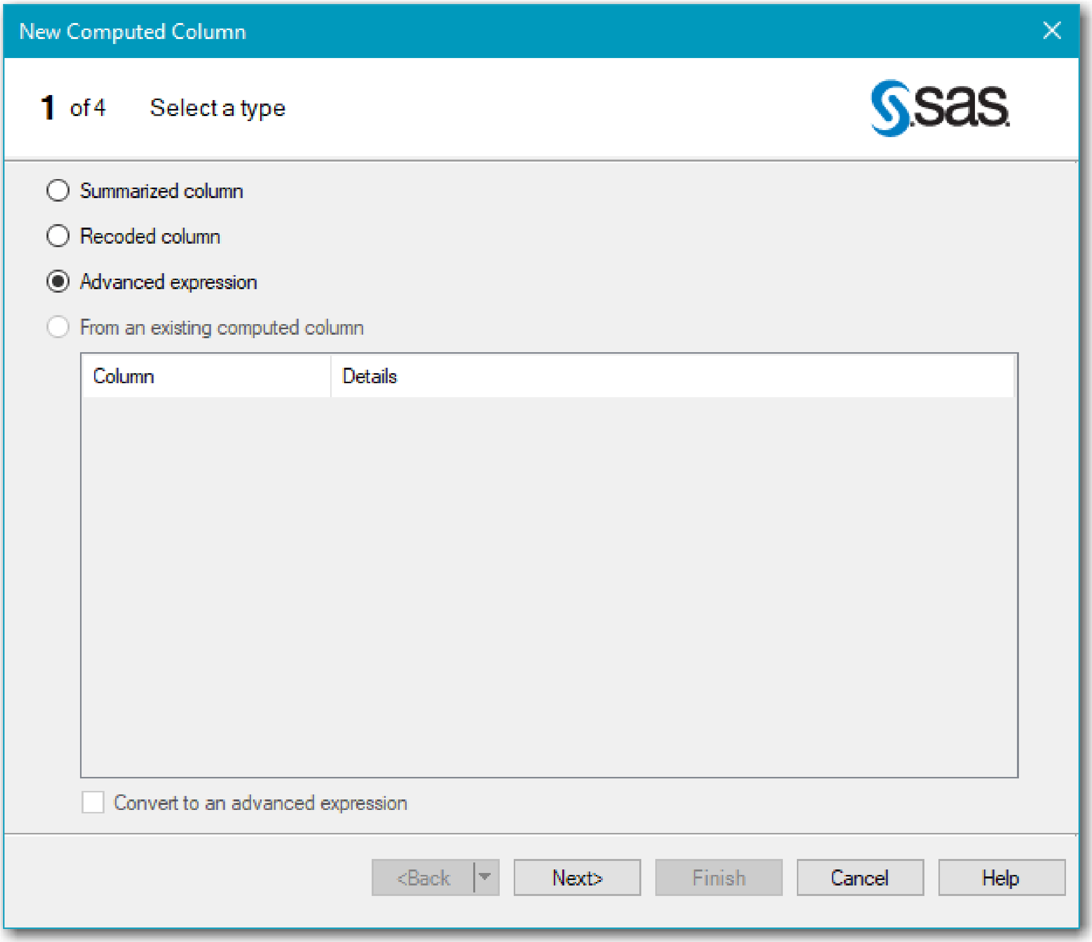
To create a conditional expression:

1 In the Query Builder, click **Computed Columns**. The Computed Columns window opens.

> **Note:** By default, the new column is included in the output table. If you do not want to include the new column, clear the **Add new computed columns to the query selection** check box.

2 Click **New**. The New Computed Column wizard opens.

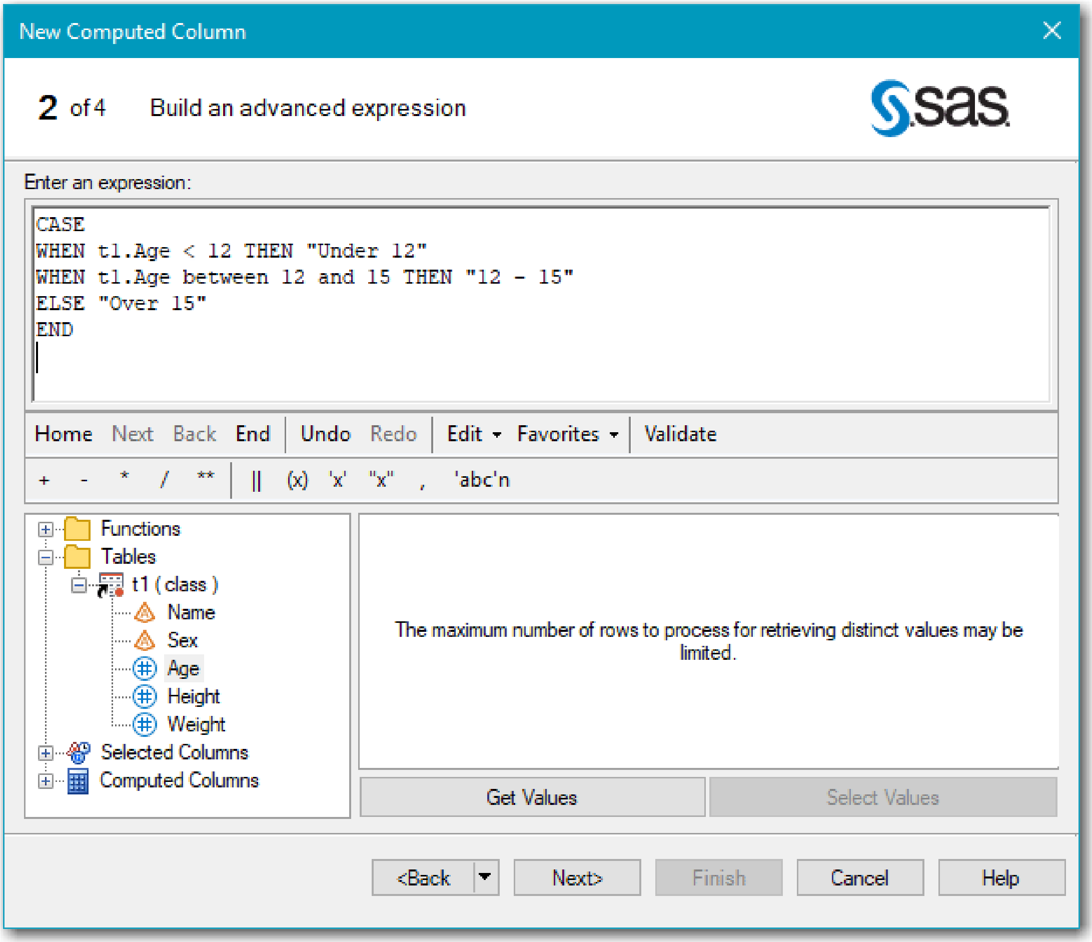
3 For the type of column, select **Advanced expression**. Click **Next**.



4 Use the following template for the CASE expression syntax to type your own CASE expression in the expression box:

```
CASE <caseOperand>
          WHEN <whenCondition> THEN <resultExpression>
          WHEN <whenCondition> THEN <resultExpression>
          ELSE <resultExpression>
       END
```

- <*caseOperand*> – the name of the column that you are using in the expression
- <*whenCondition*> – the existing value of the column
- <*resultExpression*> – the value that you want to assign to the column



```
CASE
WHEN t1.Age < 12 THEN "Under 12"
WHEN t1.Age between 12 and 15 THEN "12 - 15"
ELSE "Over 15"
END
```

The maximum number of rows to process for retrieving distinct values may be limited.

5 Enter the column name and values that you want to use. Click **Next** to specify additional options for the new computed column.

6 To change the default name of the computed column, enter the new name in the **Column Name** box.

7 To specify a label for the column, enter the label in the **Label** box. Valid column labels cannot exceed 256 characters. The label is displayed as the column heading in your report output.

Note: You can specify that all spaces and special characters in column names and labels be converted to underscores. For more information, see “Setting Query Options” on page 500.

8 If you want to apply a summary function to the computed column, select the function from the **Summary** drop-down list.

9 To change the length of the new column, type a new length in bytes in the **Length** box. For numeric columns, the length must be a value between 3 and 8,

inclusive. For character columns, the length must be a value between 1 and 32,767, inclusive.

10 To apply a format to the summarized column, click **Change** to open the Formats window, and then select the format and attributes that you want to use. Click **OK** to save the format changes. Then click **Next**.

11 Verify that the properties of the new column are correct. If you want to make any changes, click **Back** until you reach the page on which you want to make a correction.

12 Click **Finish** to create the new computed column.

Note: For more information, see "CASE" in *SAS SQL Procedure User's Guide*.

Sample Conditional Expressions

The following example uses the SUBSTR function to extract the first letter of the last name and then assign a replacement value based on that letter.

```
CASE WHEN (SUBSTR(tablename.LastName,1,1) between "A" and "F") THEN "A
- F"
         WHEN (SUBSTR(tablename.LastName,1,1) between "G" and "L")
THEN "G - L"
         WHEN (SUBSTR(tablename.LastName,1,1) between "M" and "R")
THEN "M - R"
         WHEN (SUBSTR(tablename.LastName,1,1) between "S" and "Z")
THEN "S - Z"
    END
```

The following example replaces age values with "Under 35," "35 - 50," and "Over 50."

```
CASE
       WHEN tablename.Age < 35 THEN "Under 35"
       WHEN tablename.Age between 35 and 50 THEN "35 - 50"
       ELSE "Over 50"
    END
```

Using Prompts in a Query

Prompts in a query require the user to select or enter a value when the query runs. In a query, prompts can be used as the comparison value in a conditional expression for a filter or a recoded column. The value that the user selects or enters for the prompt is compared with the value of the column that is used in the conditional expression. You can also use a prompt to supply a value for a title or footnote.

For example, suppose you have a query that lists sales by Product Name for a specific quarter. First, you create a numeric prompt called *Quarter_prompt*. You can populate the values for this prompt from the Quarter variable in your input data source so that when the prompt runs, the user can select only from the values in the input data source. Then, you open the Query Builder and create a basic filter as follows:

```
Column = Quarter
Operator = Equal to
Value = &Quarter_prompt
```

When the query is run, the user selects the quarter for which to display sales.

For a query, you can create a prompt in two ways:

- To create a prompt before you create the query, use the Prompt Manager in the Resources pane. For more information, see "About Creating Prompts" on page 385.
- To create a prompt after you create the query, use the Prompt Manager in the Query Builder.

If you use a prompt in a basic filter, then the Query Builder automatically associates the prompt with the query. If you want to use a prompt in an advanced filter, a recoded or advanced expression column, a title, or a footnote, then you must associate the prompt with the query by adding it to the list of project prompts for the query.

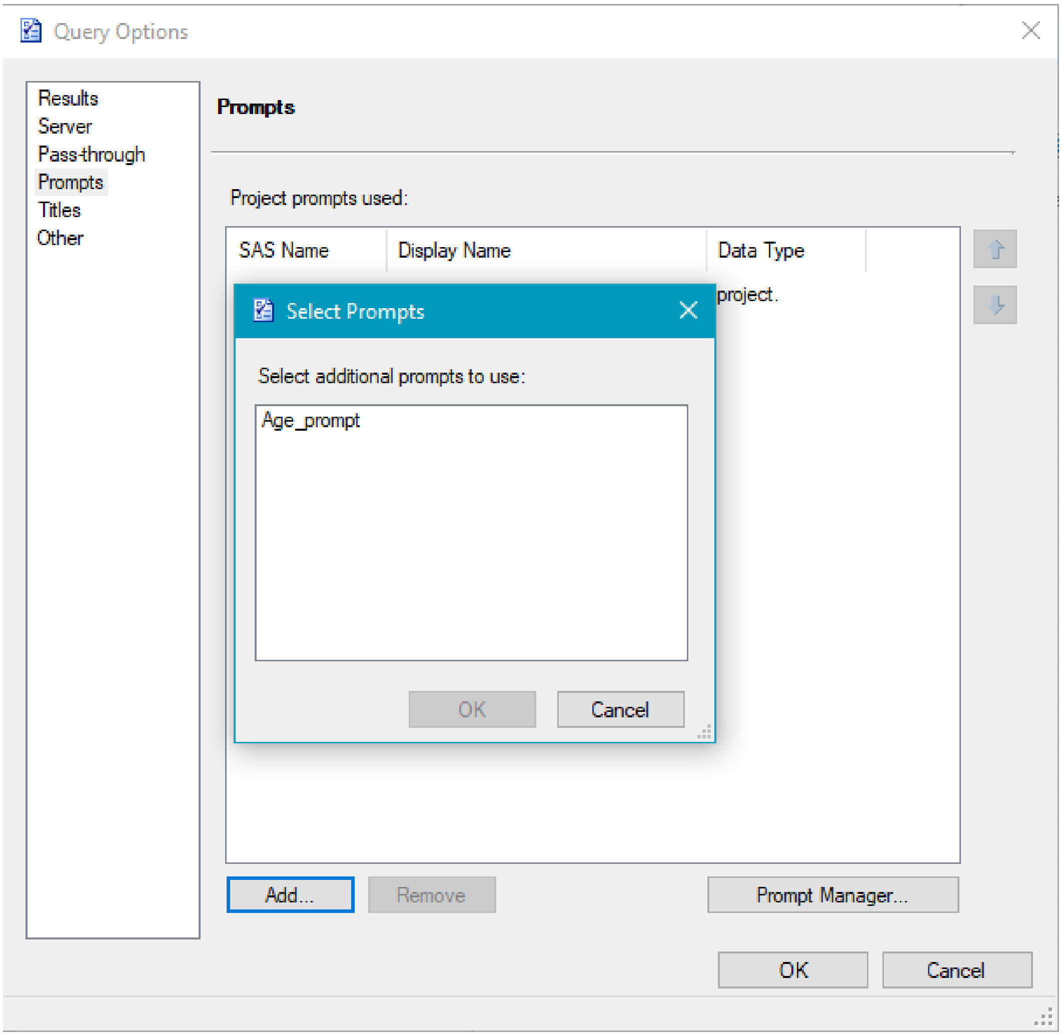
Note: If the prompt is not associated with the query item, then the user is not prompted for a prompt value when the query is run.

To create a prompt by using the Prompt Manager in the Query Builder:

1. In the Query Builder, click **Prompt Manager**. The Prompt Manager window opens.
2. To create a prompt, click **Add**. The Add New Prompt window opens. For more information, see "About Creating Prompts" on page 385.

To associate an existing prompt with a query:

1. In the Query Builder, select **Options** ⇨ **Options for This Query** . The Query Options window opens.
2. Click **Prompts** in the selection pane, and then click **Add**. The Select Prompts window opens.



3 Select the prompt that you want to associate with the query and click **OK**.

4 Click **OK** again to close the window. When you use the prompt in a query, the prompt should be added to the query as *&prompt-name*. You can now use the prompt as the comparison value in a conditional expression for a filter or a recoded column.

Note: To remove the association between a prompt and a query, select the prompt that you want to remove on the Prompts page of the Query Options window. Then click **Remove**. You might need to edit or delete any filters, computed columns, titles, or footnotes that use that prompt.

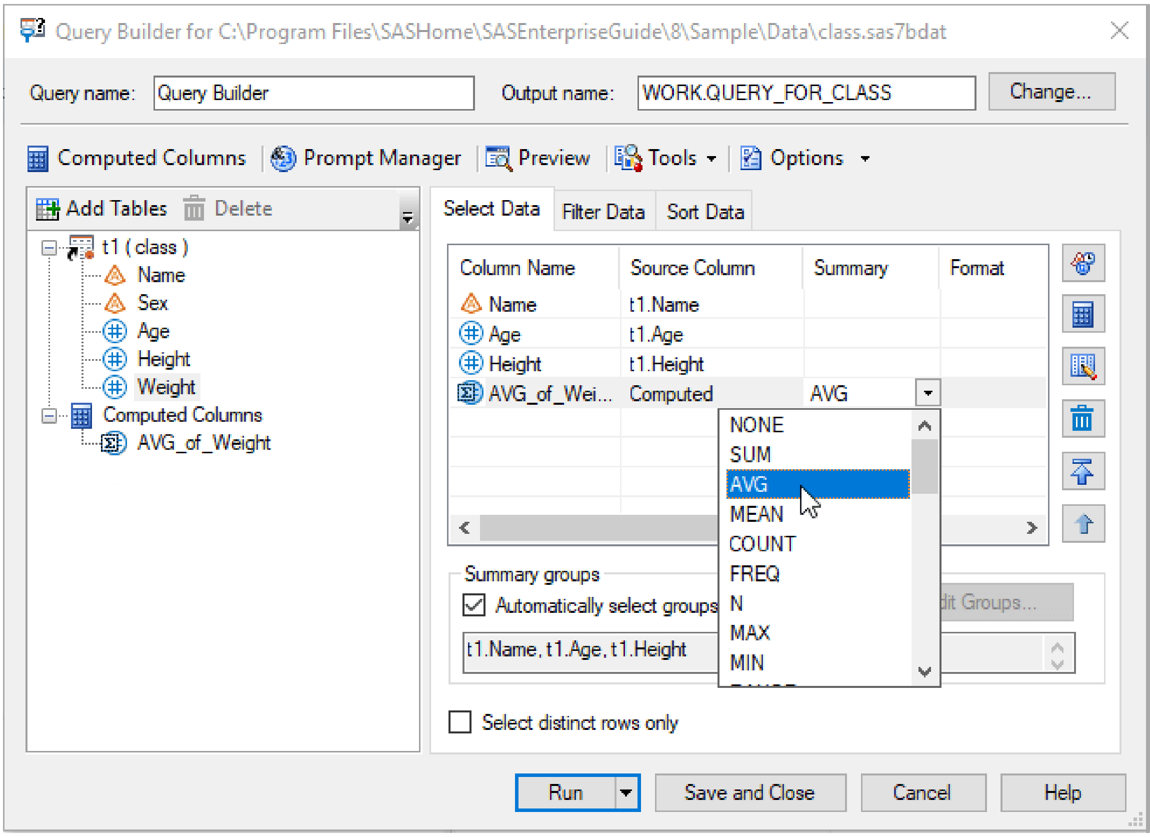
Using Summary Functions

You can perform summary functions on any of the columns in your query. When you perform a summary function on a column, a new alias name is assigned to the column. You can change the alias and format of the summarized column by editing the properties for the column.

Note: You can also create a summarized column by using the Computed Column wizard. In the Computed Column wizard, you can change the default alias and column format without having to edit the column properties separately. For more information, see "Creating a Summarized Column" on page 223.

1 In the Query Builder, click the **Select Data** tab.

2 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, when you perform a summary function on a column, a new alias is assigned to the column.



Query Builder for C:\Program Files\SASHome\SASEnterpriseGuide\8\Sample\Data\class.sas7bdat

Query name: Query Builder

Output name: WORK.QUERY_FOR_CLASS

Column Name	Source Column	Summary	Format
Name	t1.Name		
Age	t1.Age		
Height	t1.Height		
AVG_of_Wei...	Computed	AVG	

Summary groups

☑ Automatically select groups

t1.Name, t1.Age, t1.Height

☐ Select distinct rows only

3 To change the alias or format of the summarized column, click . The Properties for *column-name* window opens.

4 To change the name of the column, type the new name in the **Column Name** box. The name is displayed as the column heading in your output table.

Note: For information about changing the format of a column, see "Formatting the Data in a Query" on page 254.

5 Click **OK** to save the column properties.

Formatting the Data in a Query

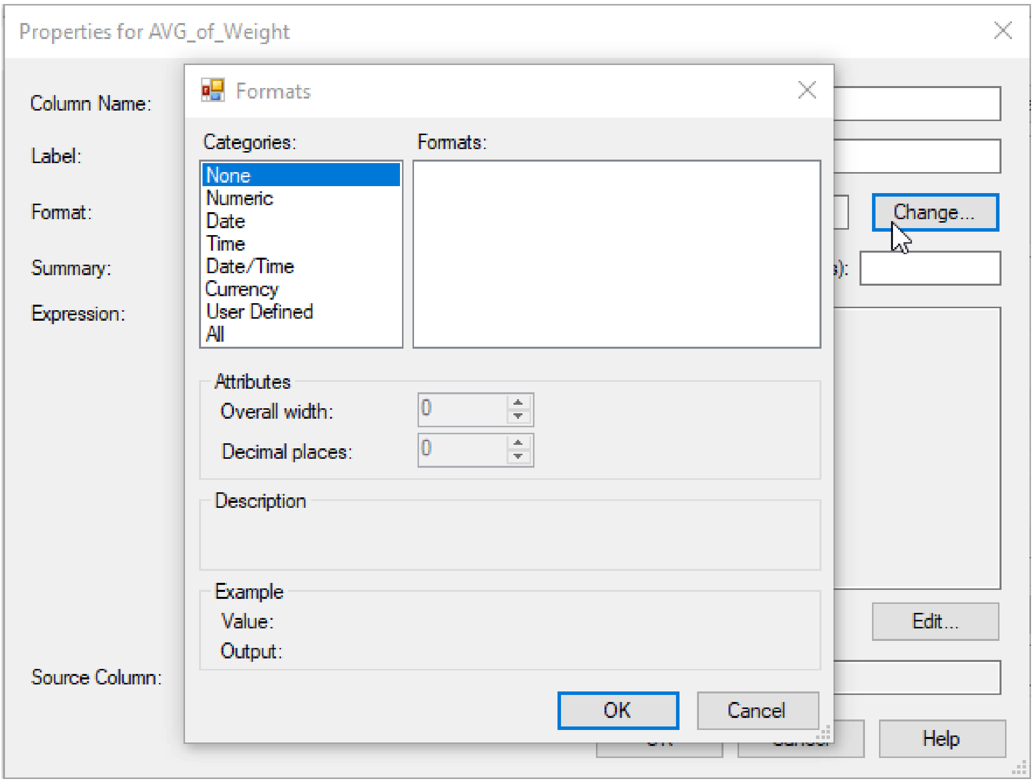
You can apply predefined and user-defined formats to columns in your output so that the results are easier to read and understand. For information about creating your own formats, see "About the Create Format Task" in *SAS Tasks in SAS Enterprise Guide and SAS Add-In for Microsoft Office*.

To format the data in a query:

1 In the Query Builder window, click the **Select Data** tab.

2 Select the column for which you want to specify data value formats, and then click . The Properties for *column-name* window opens.

3 Beside the **Format** box, click **Change**. The Formats window opens.



Overall width: 0

Decimal places: 0

4 Select the data type from the **Categories** box, and then select the desired format for that data type from the **Formats** box.

For a list of all the SAS formats and informats, see *SAS Formats and Informats: Reference*.

5 To change the overall width of the column, select a new value in the **Overall width** box. To change the number of decimal places that are displayed in the output, select a new value in the **Decimal places** box.

Note: A description of the format and a sample of how the formatted column will appear in your output are displayed at the bottom of the Formats window.

6 Click **OK** to return to the Properties for *column-name* window.

7 Click **OK** again to set the format.

Filtering Data

Creating a Filter

About 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 the Query Builder which rows to retrieve is called setting a filter and is done on the Filter Data tab. This corresponds to using a WHERE clause in an SQL query.

Note: You can also create a filter to subset grouped data. For more information, see “Subsetting Grouped Data” on page 273.

You can use just 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, and you can group elements together.

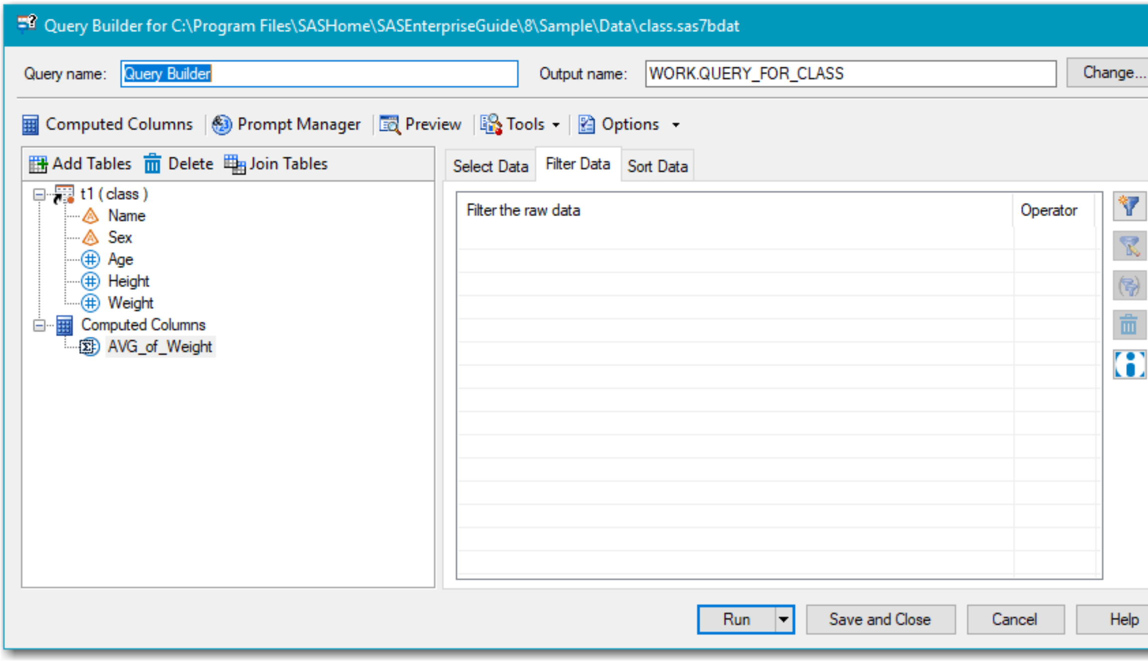
You can choose to create either a basic filter or an advanced filter by using the expression builder.

Creating a Basic Filter

You can create a basic filter by using the New Filter wizard in the Query Builder. After you select the column on which you want to filter the data, you can create the rest of the comparison expression by selecting an operator and a comparison value.

To create a basic filter:

1 In the Query Builder, click the **Filter Data** tab.



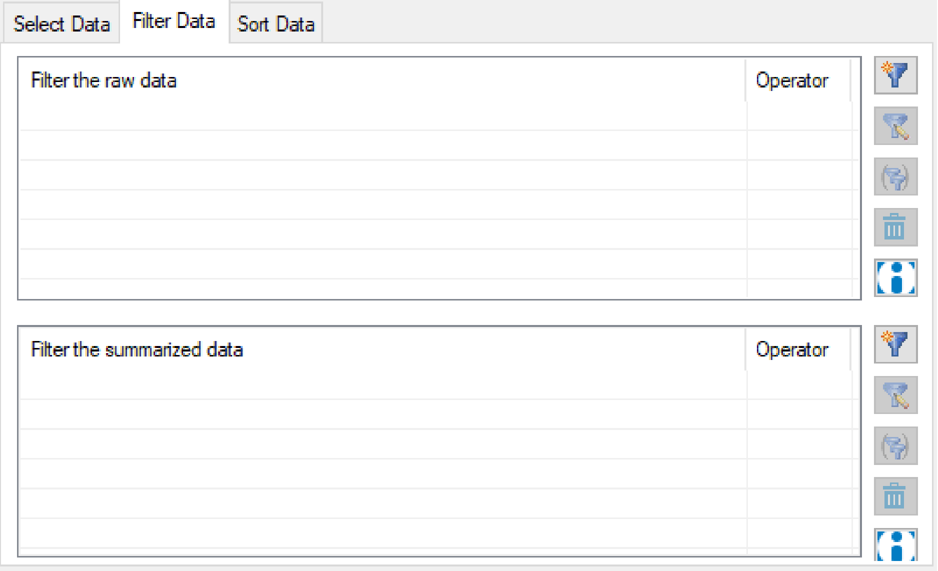
Query Builder for C:\Program Files\SASHome\SASEnterpriseGuide\8\Sample\Data\class.sas7bdat

Query name: Query Builder

Output name: WORK.QUERY_FOR_CLASS

2 Click to open the New Filter wizard.

Note: If your data is grouped by a summarized column, you can choose to filter either the raw data or the summarized data. If you want to filter the raw, unsummarized data, click in the Filter the raw data area. If you want to create a filter to subset grouped data, click in the Filter the summarized data area. The Filter the summarized data area appears only when there is grouped data in the query. For more information, see "Grouping Your Output" on page 271.

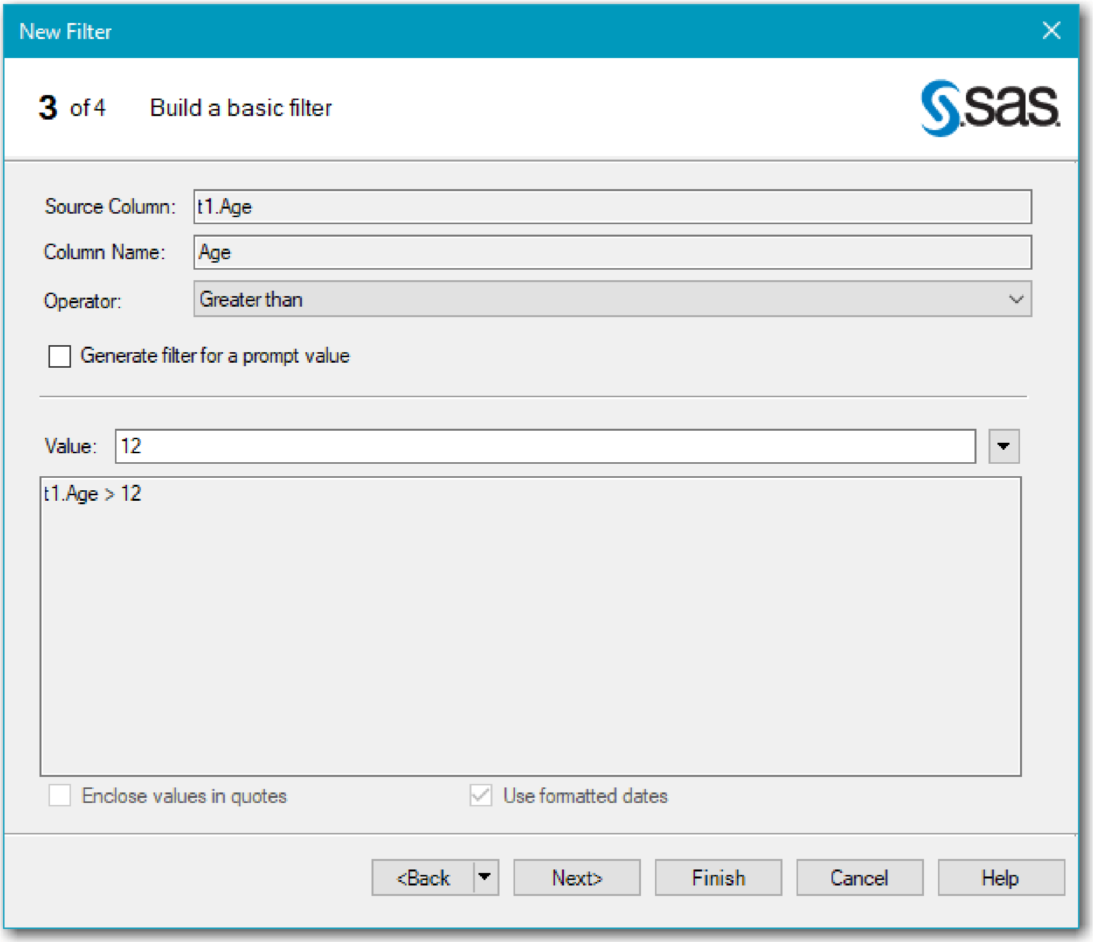


Note: You can also create a basic filter by dragging a column to the **Filter Data** tab.

3 Select **Basic Filter** and click **Next**.

4 Select the column on which you want to filter the data, and then click **Next**.

Note: By default, the columns in the list are displayed by name instead of label, and they are listed in the order in which they appear in the data table. You can change how the columns are displayed in the Query Builder. Changing how the columns are displayed does not affect how the SQL query is processed or how the columns appear in the output table. For more information, see "Customizing Data Display in the Query Builder" on page 200.



New Filter

3 of 4 Build a basic filter

Source Column: t1.Age

Column Name: Age

Operator: Greater than

☐ Generate filter for a prompt value

Value: 12

t1.Age > 12

☐ Enclose values in quotes ☑ Use formatted dates

5 Select a comparison operator from the **Operator** drop-down list. The default value is **Equal to**.

6 If you are using a prompt value in the filter and you want to use the %_eg_WhereParam custom macro in the SQL code for the filter, then select **Generate filter for a prompt value**. If you are using the query in a stored process, then you must select this option. For more information about prompts, see "About Prompts" on page 372.

7 If the operator that you have selected requires a value, you can enter the value in the **Value** box.

If you want to select from a list of values, columns, prompts, or subqueries, then click ▾.

- On the **Values** tab to display a list of column values, click **Get Values**. The column values are displayed. If there are additional values that can be displayed, you can click **More Values**.

  Note: For large tables, generating the list of available values can take several minutes. However, if you enter your own value instead of choosing one from the list, then the filter results might not be what you expect. You can use the **Number of distinct column values to get for each request** option to control how many values are displayed at one time. For more information, see "Setting Query Options" on page 500.

  By default, the values are displayed with a common format. To display the values with a SAS format, select **SAS Format** from the **Show** drop-down list.

  Note: If the column is defined as a currency, time, date, or datetime format, these common formats are used:

  - Currency – $10.00
  - Time – 2:29:09 PM
  - Date – Monday, April 17, 2019
  - DateTime – Monday, April 17, 2019 2:29:09 PM

- Click the **Columns** tab to select the column that contains the values that you want to use.
- Click the **Prompts** tab to select the prompt that you want to use.
- Click the **Subqueries** tab to select a template for a subquery. For more information, see "About Task Templates for Queries" on page 267. Verify that the data references that are listed are still valid. If you need to change a data reference, click **Change** and select another data file.

  Note: If SAS Enterprise Guide detects a problem with the data references in the query, the ⚠ is displayed by the data. You must verify that the data exists before the **OK** button is enabled.

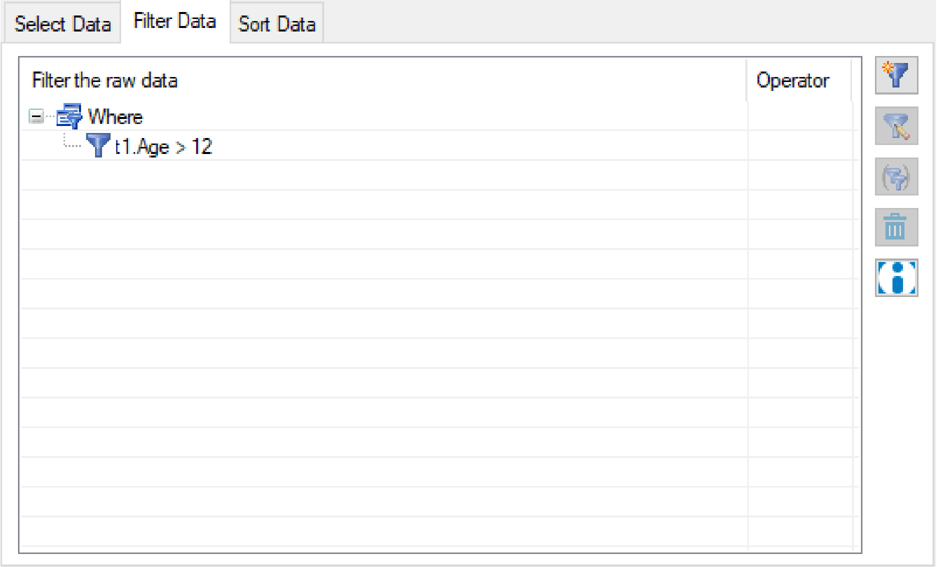
8 If you want the values to be enclosed in quotation marks, select the **Enclose values in quotes** check box.

9 To use unformatted dates in the comparison expression, clear the **Use formatted dates** check box. If you select this option, the formatted values are used in the expression. It is more efficient to use unformatted date values than formatted date values in the comparison expression.

Note: This option is available only if the column that you have selected uses a date, time, or datetime format.

10 Click **Next** to verify that the properties of the new filter are correct. If you want to make any changes, click **Back** until you reach the page on which you want to make a correction.

11 Click **Finish** to create the filter. The new filter is added to the list of filters on the Filter Data tab.



Note: To view the SQL code that was generated for the filter, click . The filter expression pane opens at the bottom of the Filter Data tab. To close the filter expression pane, click again.

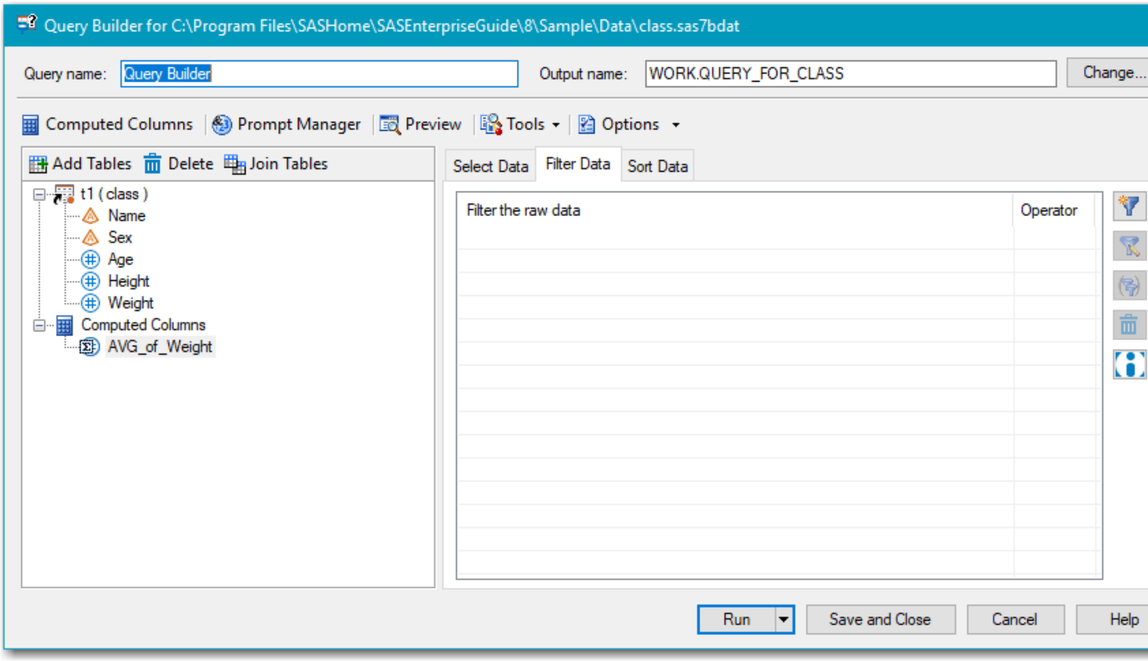
Creating an Advanced Filter

You can create an advanced filter by using the expression builder to build the comparison expression.

To create an advanced filter:

1 In the Query Builder, click the **Filter Data** tab.

2 Click to open the New Filter wizard.

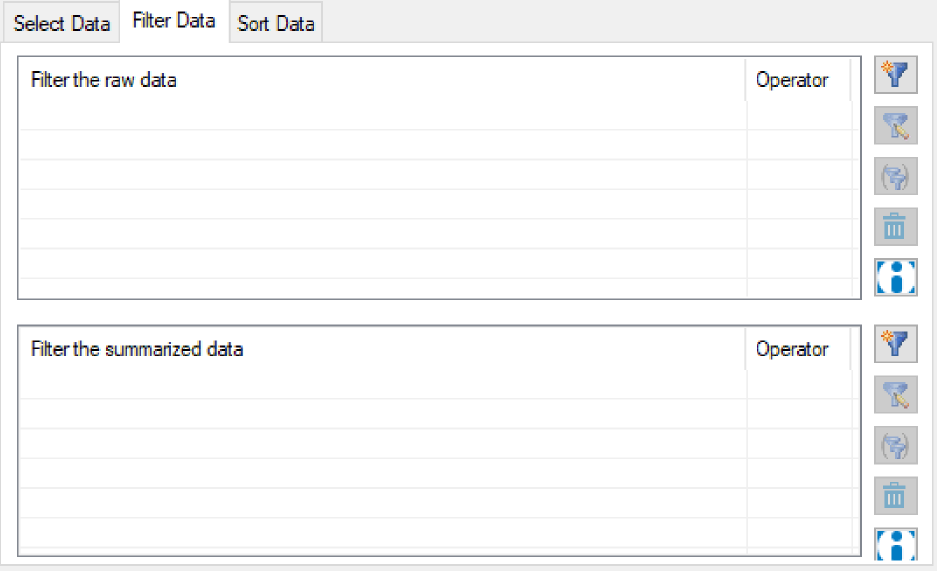


Query Builder for C:\Program Files\SASHome\SASEnterpriseGuide\8\Sample\Data\class.sas7bdat

Query name: Query Builder

Output name: WORK.QUERY_FOR_CLASS

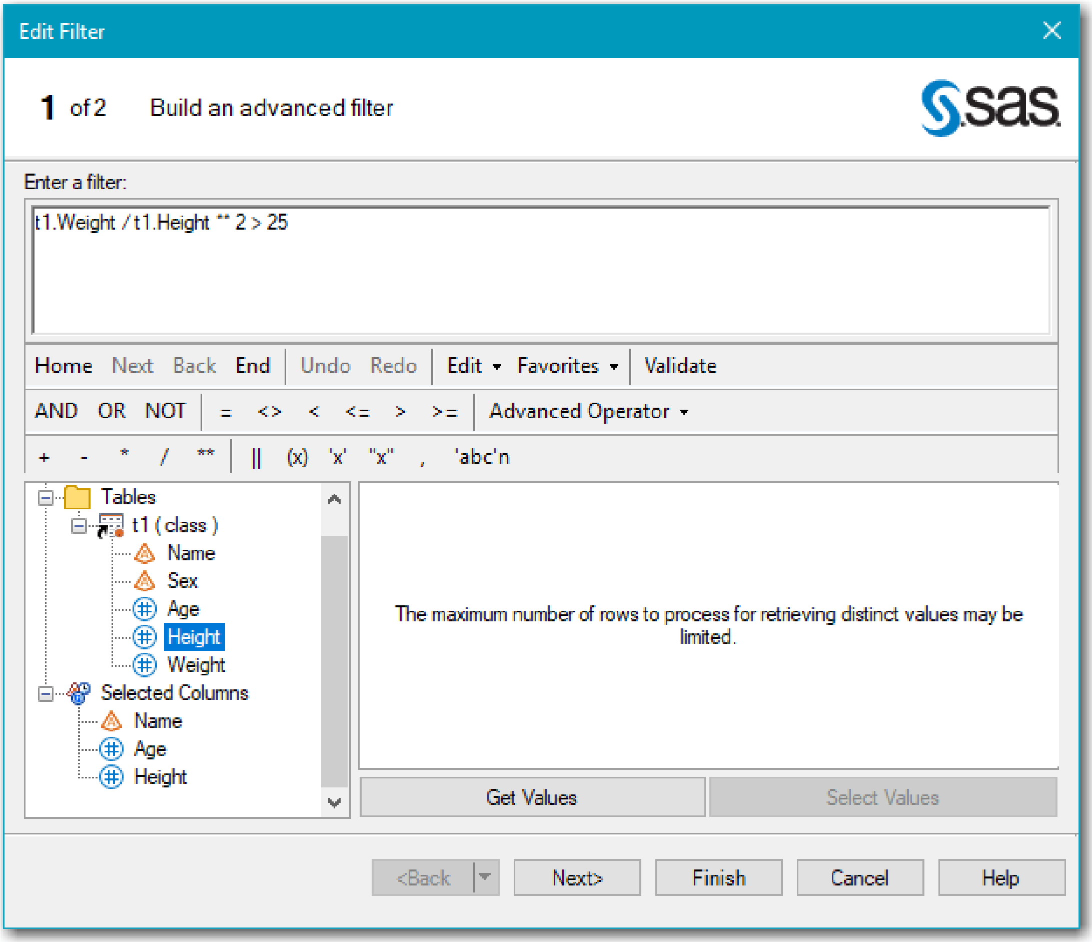
Note: If you want to filter the raw, unsummarized data, click in the Filter the raw data area. If you want to create a filter to subset grouped data, click in the Filter the summarized data area. The Filter the summarized data area appears only when there is grouped data in the query.



3 Select **Advanced Filter** and click **Next**.

4 You can build your expression by using mathematical operators, functions, columns, and constant values. For a list of all SAS functions, see *SAS Functions and CALL Routines: Reference*.

The constant value can either be a number (such as 10 or 3.14) or a string (such as 'NC' or 'Sales Numbers'). You can type these elements into the expression box at the top of the window, or you can use the expression editor to add these elements. For more information, see "Creating an Advanced Expression" on page 283.


Edit Filter
1 of 2 Build an advanced filter
Enter a filter:
Home Next Back End Undo Redo Edit Favorites Validate
AND OR NOT = <> < <= > >= Advanced Operator
+ - * / ** || (x) 'x' "x" , 'abc'n
Tables
t1 (class)
Name
Sex
Age
Height
Weight
Selected Columns
Name
Age
Height
Get Values
Select Values
<Back
Next>
Finish
Cancel
Help

t1.Weight / t1.Height ** 2 > 25

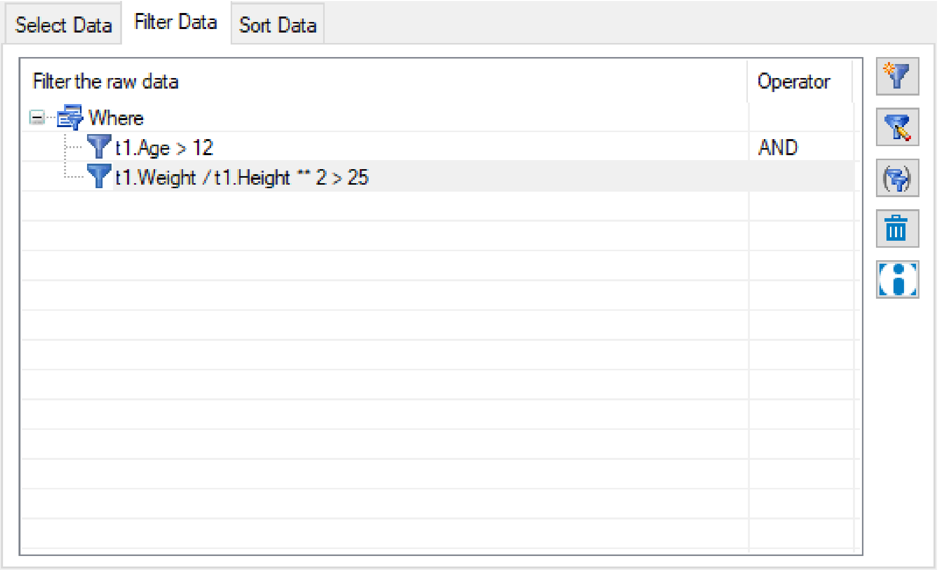
The maximum number of rows to process for retrieving distinct values may be limited.

Note: When you select the **Use labels for column names** option to display the columns by label, an additional expression box opens to display the expression using the labels. However, you can type only in the expression box that uses column names. For more information, see "Setting General Data Options" on page 497.

5 After you have finished building the filter expression, click **Next**.

6 Click **Next** to verify that the properties of the new filter are correct. If you want to make any changes, click **Back** until you reach the page on which you want to make a correction.

7 Click **Finish** to create the filter. The new filter is added to the list of filters on the **Filter Data** tab.



Where
t1.Age > 12 AND
t1.Weight / t1.Height ** 2 > 25

Note: To view the SQL code that was generated for the filter, click ⓘ. The filter expression pane opens at the bottom of the **Filter Data** tab. To close the filter expression pane, click ⓘ again.

Editing an Existing Filter

You can edit and delete filters that you have created.

To edit a filter:

1 In the Query Builder, click the **Filter Data** tab.

2 Select the filter that you want to edit and click . The Edit Filter wizard opens so that you can make changes to the filter.

Note: You can also edit a filter from the Filter Combination window by selecting the filter that you want to edit and clicking **Edit Filter**. For more information, see "Grouping Filter Elements" on page 264.

Note: To view the SQL code that was generated for the filter, click ⓘ. The filter expression pane opens at the bottom of the Filter Data tab. To close the filter expression pane, click ⓘ again.

To delete a filter:

1 In the Query Builder, click the **Filter Data** tab.

2 Select the filter that you want to delete and click 🗑.

Note: You can also delete a filter from the Filter Combination window by selecting the filter that you want to delete and clicking **Delete Filter**.

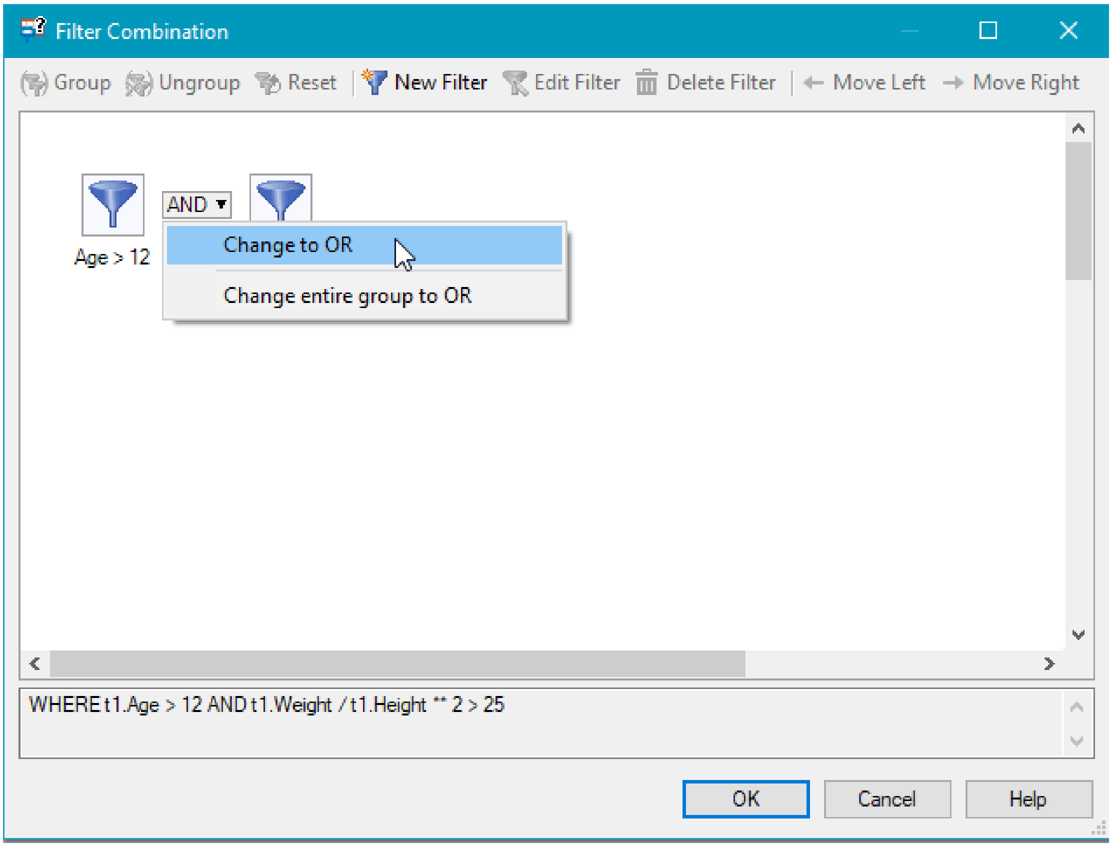
Changing the Relationship between Filters

You can use just 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, and you can group elements together. For more information, see "Grouping Filter Elements" on page 264.

To change the relationship between filters:

1 In the Query Builder, click the **Filter Data** tab.

2 Click . The Filter Combination window opens.



WHERE t1.Age > 12 AND t1.Weight / t1.Height ** 2 > 25

Note: If you want to combine filters on raw, unsummarized data, click in the Filter the raw data area. If you want to combine filters of grouped data, click in the Filter the summarized data area. The Filter the summarized data area appears only when there is grouped data in the query. For more information, see "Grouping Your Output" on page 271.

3 You can reorganize your filter elements in the following ways:

- To change the AND or the OR relationship between the filters, click on the relationship between the filters (the AND or the OR label) and select **Change to *relationship-value***.

  Note: You can also change the relationship between a pair of filters by using the Filter Data tab. In the Filter the raw data area of the Filter Data tab, click the current AND or OR relationship and select the new relationship from the drop-down list.

- To change the AND or the OR relationships between all of the filters, click on a relationship between two of the filters and select **Change entire group to *relationship-value***.
- To reset the filter so that no elements are grouped and the filter relationships are all AND, click **Reset**.
- To change the order in which the filter elements are processed, click the filter element that you want to move and then click **Move Left** or **Move Right**.

4 Click **OK** to save your changes.

Grouping Filter Elements

When you have two or more filter elements in a filter, you can group consecutive elements together. When you group filter elements, you are putting parentheses around the selected elements. The elements within the parentheses are evaluated as a group when the filter is applied.

For example, suppose that you have a filter that consists of these elements:

```
element1 AND element2 OR element3 AND element4
```

You can group together element2 and element3. The filter is then evaluated as follows:

```
element1 AND (element2 OR element3) AND element4
```

You can also group non-consecutive elements, such as element 1 and element 4.

```
(element1 AND element4) AND element2 OR element3
```

To group filter elements in the filter:

1 In the Query Builder, click the **Filter Data** tab.

2 Click button to open the **Filter Combination** window.

Note: If you want to group filters on raw, unsummarized data, click button in the Filter the raw data area. If you want to group filters of grouped data, click button in the Filter the summarized data area. The Filter the summarized data area appears only when there is grouped data in the query. For more information, see "Grouping Your Output" on page 271.

3 Select the filter elements that you want to group and click **Group**.

To ungroup grouped filter elements, select the parent parentheses node for the group and then click **Ungroup**.

Note:

- To reset the filter so that no elements are grouped and the filter relationships are all AND, click **Reset**.
- To change the order in which the filter elements are processed, click the filter element that you want to move and then click **Move Left** or **Move Right**. To move an entire group, select the parent parentheses node for the group and then click **Move Left** or **Move Right**.

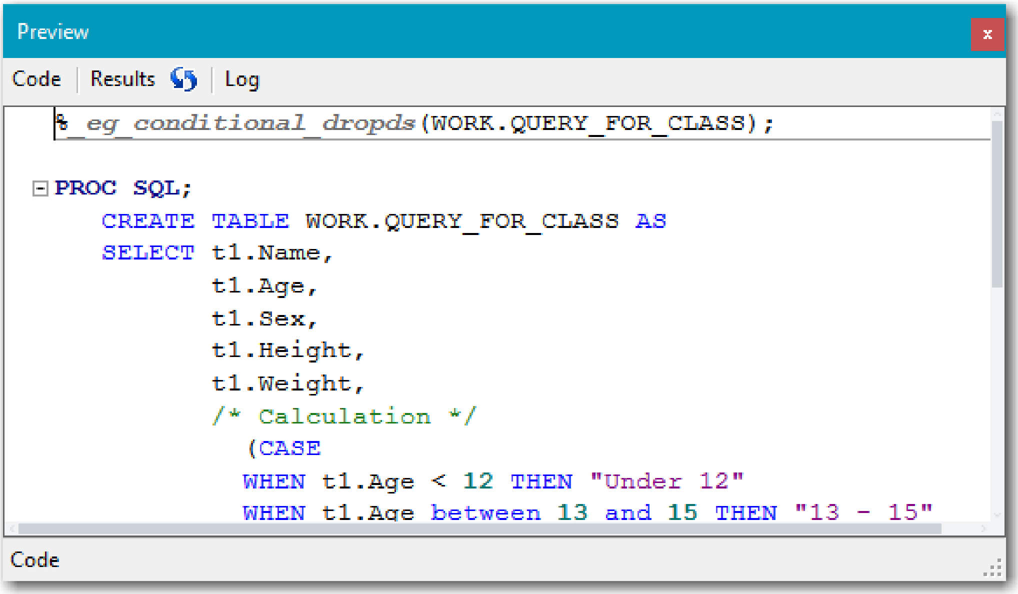
Validating the Query and Previewing the Results

Previewing the Query

You can preview the results of your query. And, although you do not need to know anything about the Structured Query Language (SQL) to use the Query Builder, you can also view the SQL code and the log that are generated by the Query Builder.

To preview your query:

- Click **Preview**. The Preview window opens. You can view the following information from the Preview window:



```
%_eg_conditional_dropds(WORK.QUERY_FOR_CLASS);

PROC SQL;
   CREATE TABLE WORK.QUERY_FOR_CLASS AS
   SELECT t1.Name,
          t1.Age,
          t1.Sex,
          t1.Height,
          t1.Weight,
          /* Calculation */
            (CASE
            WHEN t1.Age < 12 THEN "Under 12"
            WHEN t1.Age between 13 and 15 THEN "13 - 15"
```

- **Code** – displays the SQL code that is automatically generated by the Query Builder. The code changes dynamically as you make changes to tables, joins, columns, sorting, or filters.

  Note: You can view the SQL code only in the Preview window; you cannot edit it. If you want to make and save changes to the code that is generated by the Query Builder, you need to create a code template. For more information, see “Extracting Code from a Query” on page 204.

- **Results** – displays the query results. You can refresh the results by clicking ↻ to reflect any changes made to the tables, joins, columns, sorting, or filters.

  Note: The number of rows that are displayed in the Preview window is determined by the **Number of rows to process in preview results window** option. For more information, see “Setting Query Options” on page 500.

- **Log** – displays the log. The log is updated only after you refresh the results.

Validating Your Query Syntax

You can validate the query syntax that is automatically generated by the Query Builder. You can choose to validate the syntax automatically every time you make changes to the query. You can also manually validate the query syntax. The results of the validation are displayed in a separate window.

The validation log displays any error information that might be generated by the query as well as any implicit pass-through tracing information, if the query uses SQL pass-through. Examples of query syntax that might generate errors include joining tables on columns of different data types or creating a filter on an invalid condition. If the query was passed to a database that cannot process a function that was used in the query, then the validation process might return a syntax error. However, if SAS is able to process the function, then the query can still run without an error. After the query has been validated, you can decide whether you want to run the query.

To automatically validate the query syntax:

- In the Query Builder, create the query that you want to run.
- To validate the query syntax every time you make a change to the query, select **Tools** and make sure that the **Automatically Validate Changes** option is selected.
- If you make changes to the query that result in invalid syntax, a warning window opens. To view the log, click **Yes**. The Preview window opens with the log selected. You can check the log for errors. For more information, see “Previewing the Query” on page 265.

To manually validate the query syntax:

- In the Query Builder, create the query that you want to run.
- Select **Tools** ⇨ **Validate Changes Now**. The Validate window opens and displays the query syntax along with any syntax errors.

Creating Templates and Subqueries

About Task Templates for Queries

When you create a query, you might want to run the same query on data that is updated regularly. For example, you could create a query on monthly sales data that includes computed columns for profit and percent of target. A new sales data file is created each month, but the format of the data remains the same. Instead of re-creating the query each month for the latest sales data, task templates enable you to save the settings for your specific query to a template. You can then open the template, select the latest sales data as the input data, and run the query.

Note: Templates for queries that return only a single column can also be used to create subqueries. Subqueries enable you to use the results from one query as input for another query. For more information, see “About Subqueries” on page 268.

Query templates save all references to the data that was used to create them. Before you can open a query template, you must verify that the data references in the query refer to valid input data. When working with query templates, remember the following information:

- Query templates include references to specific data columns, so the data that you select must include all columns that are used in the query. A warning message is displayed if the data that you select does not include all of the columns that are used in the query.
- If the query template was originally created with temporary data that no longer exists, such as data in the Work library, you must select new input data.
- Prompts are saved as part of a query template, so the project in which you are using the template must include all prompts that are used in the query.

Note: For more information, see "About Task Templates" on page 524.

Creating Subqueries

About Subqueries

Subqueries enable you to use the results from one query as the input for another query. You can use subqueries to create filters and create a recoded column with a conditional value. In the Query Builder, you can create subqueries by using a task template for a query. In order to be used in a subquery, a template must return only a single column. For more information, see "About Task Templates for Queries" on page 267.

Note: If a template returns more than one row, then you can use only the **In a list** or **Not in a list** operators.

Using Subqueries in a Filter

Using subqueries to create a filter is an alternate way to subset your data based on values in other tables without joining the tables. You might use a subquery in a filter when you know how to search for one or more values that you want to use in the filter, but you do not know the exact values. For more information, see "About Creating a Filter" on page 255.

For example, suppose that you have a sales history table that includes product numbers, the order date, and a customer number for each order that was placed. A separate customer table includes data about each customer, including customer number and region. You need to analyze the sales data in the Central region.

You can use a subquery to create a query that returns only sales data for orders to customers in the Central region. First, you create a template from the customer table that returns only the values in the customer number column and includes a basic filter as follows:

```
Column = Region
Operator = Equal to
Value = "Central"
```

Note: For more information, see "About Task Templates for Queries" on page 267.

Then, you create a query on the sales history table with a basic filter that uses the values returned by the template as its filter values:

```
Column = Customer Number
Operator = In a list
Value = Central Region template
```

Note: If a template returns more than one row, then you can use only the **In a list** or **Not in a list** operators.

The filter in the query returns only sales data for customers who are in the Central region.

Note: It is also possible to create this query by joining the sales history table and the customer table. However, if the tables are very large, joining them might affect performance.

Using Subqueries in a Recoded Column

Using a subquery in a recoded column enables you to replace values in one column based on a conditional expression that includes a subset of values from another column. You might use a subquery in a recoded column to create a list of one or more values for a conditional expression when you know how to search for the values, but you do not know the exact values. For more information, see "Creating a Recoded Column" on page 227.

For example, suppose that you have a student table that includes student name, grade in school, and test scores. You want to easily see how the younger students' test scores compare to the scores of their older classmates. You can use a subquery to create a query that replaces the names of the 6th-grade students with the value "6th Grade Student." First, you create a template from the student table that returns only the values in the name column and includes a basic filter as follows:

```
Column = Grade
Operator = Equal to
Value = 6
```

Note: For more information, see "About Task Templates for Queries" on page 267.

Then, you create a query on the student table with a recoded computed column to replace the names of each student returned by the grade template with "6th Grade Student." The names of students in 7th and 8th grades are left as is. The recoded column replacement condition is as follows:

```
Column = Name
Operator = In a list
Value = 6th Grade Template
```

Note: If a template returns more than one row, then you can use only the **In a list** or **Not in a list** operators.

The SQL code that is generated for the recoded column is as follows:

```
SELECT t1.Name,
          /* SubqueryRecodeCalc */
            (CASE
               WHEN t1.Name IN
            (
               SELECT t1.Name
                  FROM MYFOLDER.STUDENT t1
                  WHERE t1.Grade = 6
            ) THEN '6th Grade Student'
               ELSE t1.Name
            END) AS SubqueryRecodeCalc
          t1.testScore,
          t1.testDate
    FROM MYFOLDER.STUDENT t1;
```

The query displays the names of all students in 6th grade as "6th Grade Student."

Managing Output

Sorting 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 Builder, click the **Sort Data** tab.

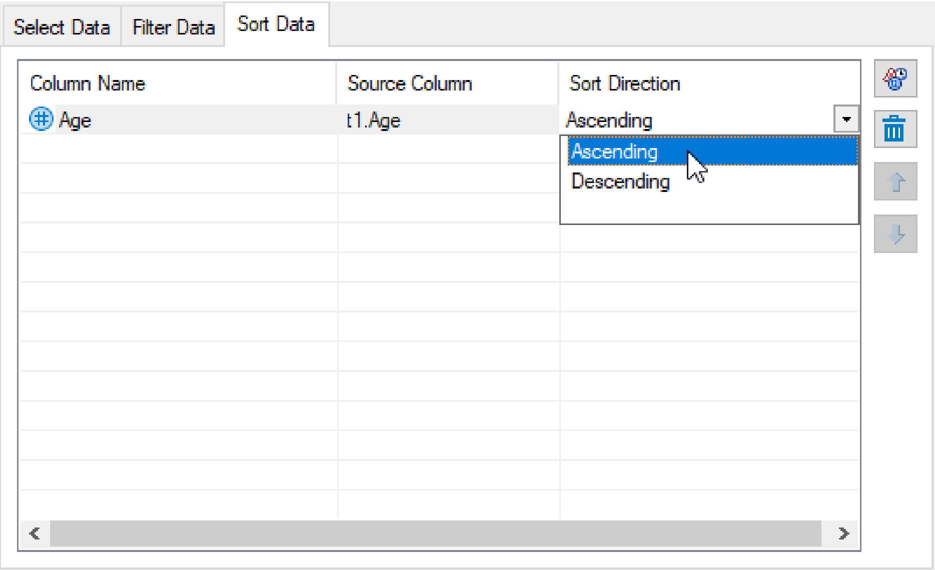
2 To add a column to the Sort Data tab, select one or more columns that you want to add, and then right-click the columns and select **Sort Query by Column**.

Note: You can also select a column for the Sort Data tab in the following ways:

- You can drag a column to the Sort Data tab or double-click it.
- On the Sort Data tab, click ⊕. The Add Column window opens. Select one or more columns that you want to add and click **OK**.

To remove a column from the Sort Data tab, select the column and click 🗑, or drag it back to the list of columns on the left.

3 Click the **Sort Direction** 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**.



Column Name	Source Column	Sort Direction
Age	t1.Age	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.

Note: You can also drag a column up or down the list.

Grouping Your Output

If you have created a summarized column or a computed column that contains a summary function, 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.

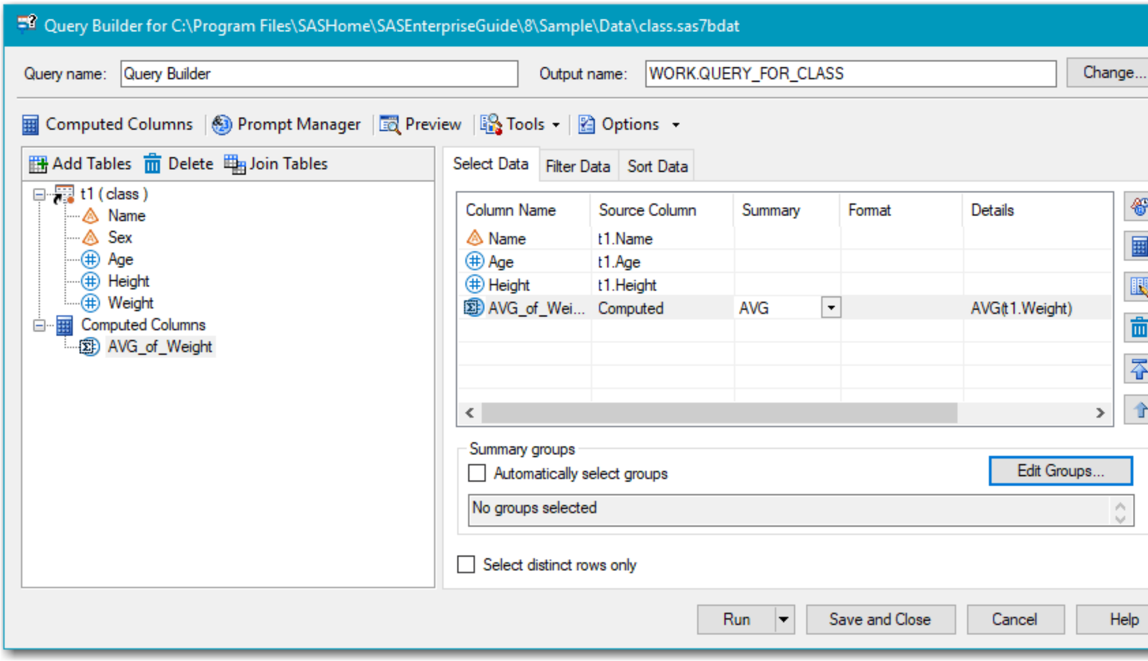
By default, when you perform a summary function on a column, your query is grouped by all columns without summaries. You can choose to edit the list of columns that the query is grouped by.

When you group your output by one or more columns, you can create a filter to subset the grouped data based on specified conditions. For more information, see “Subsetting Grouped Data” on page 273.

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

1 In the Query Builder, click the **Select Data** tab.

2 To edit the columns that the query is grouped by, make sure that the **Automatically select groups** check box is cleared. Then click **Edit Groups**. The Edit Groups window opens.



Query Builder for C:\Program Files\SASHome\SASEnterpriseGuide\8\Sample\Data\class.sas7bdat

Query name: Query Builder

Output name: WORK.QUERY_FOR_CLASS

Column Name	Source Column	Summary	Format	Details
Name	t1.Name			
Age	t1.Age			
Height	t1.Height			
AVG_of_Wei...	Computed	AVG		AVG(t1.Weight)

Summary groups

□ Automatically select groups

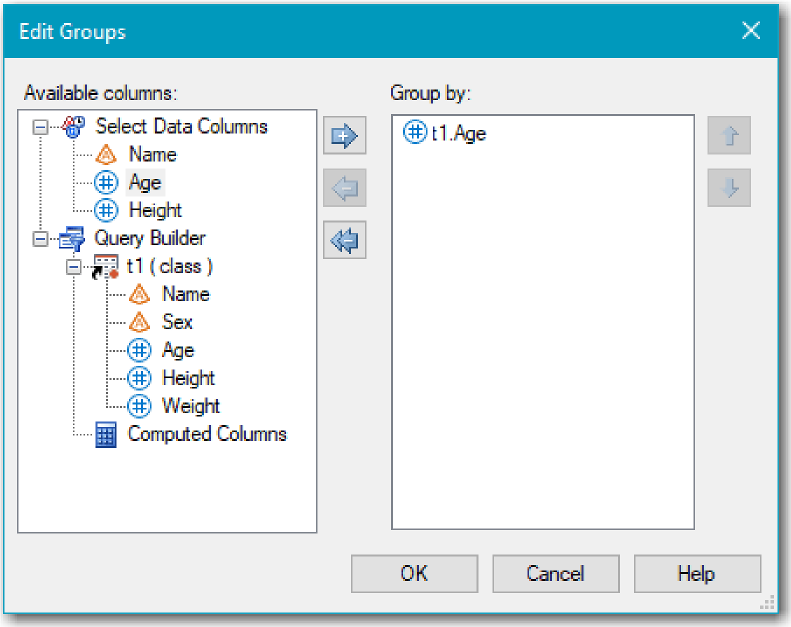
No groups selected

□ Select distinct rows only

Note: If you have not performed a summary function on a column or created a computed column that contains a summary function, then you cannot group your output.

3 To add columns to the **Group by** list, select one or more columns in the **Available columns** list that you want to add and click ⇨.

Note: To remove a column from the Group by list, select one or more columns in the Group by list that you want to remove and click ⇦ button. To remove all columns from the Group by list, click ⇚ button.



Group by: t1.Age

4 To change the order in which the columns are used to group the data, select the column that you want to move and click ⇧ or ⇩.

5 Click **OK** to save your changes and return to the Query Builder.

Subsetting Grouped Data

If you have created a summarized column or a computed column that contains a summary function, you can choose to classify your data into groups based on the values in a column. When you group your output by one or more columns, you can create a filter to subset the grouped data based on specified conditions. This is equivalent to using the HAVING expression in an SQL query.

Note: You can also choose to filter the raw, unsummarized data. For more information, see “About Creating a Filter” on page 255.

You can use just 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, and you can group elements together. For more information, see “Grouping Filter Elements” on page 264.

Note: You can choose to create either a basic filter or you can create an advanced filter by using the expression builder:

- "Creating a Basic Filter" on page 255
- "Creating an Advanced Filter" on page 259

Eliminating Duplicate Rows in Output

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

To eliminate duplicate rows:

- In the Query Builder, click the **Select Data** tab, and select the **Select distinct rows only** check box.

Specifying Limits for Input Processing

In the Query Builder, you can restrict the number of rows that your query processes from each source table. This option corresponds to the INOBS= option in PROC SQL and is useful when you want to reduce processing time in order to test a query.

When you set a limit *n* on the number of rows to process, the Query Builder uses the first *n* rows from each source table in the query. If you have added a filter to your query, then the Query Builder uses the first *n* rows from each table that match your filter criteria.

This option does not apply when the output format is a data view.

Note: If you have a computed column in your query data that performs a function on the whole table, such as sum or average, then limiting the number of input rows affects the value that is calculated for that column.

To limit the number of rows to process:

- In the Query Builder window, select **Options** ⇨ **Options for This Query** . The Query Options window opens.
- Select the **Limit number of matching rows to process** check box, and enter the number of rows that you want the query to process. You can select any number between 1 and 100,000,000, inclusive.
- Click **OK** to save your changes.

Note: When you limit the number of rows to process and you have joined tables together, the maximum number of rows that the query can return is the Cartesian product of the number of rows from each table. For example, if you set the limit to five rows and you join two tables together, the maximum number of rows that the query can return is 25 (5 * 5). If you join three tables, the maximum number of rows that the query can return is 125 (5 * 5 * 5).

Specifying Limits for Output Processing

In the Query Builder, you can restrict the number of rows that your query saves in the output. This option corresponds to the OUTOBS= option in PROC SQL and is useful when you want to restrict the size of your output for testing purposes.

This option does not apply when the output format is a data view.

Note: When you limit the size of your output, you are not reducing the processing time of the query. The Query Builder still processes all of the data in each source table. You are reducing only the size of the output.

To limit the number of output rows:

1 In the Query Builder window, select **Options** ⇨ **Options for This Query** . The Query Options window opens.

2 Select the **Limit number of rows to save in output** check box, and enter the number of rows that you want in your output. You can select any number between 0 and 100,000,000, inclusive.

3 Click **OK** to save your changes.

Choosing the Results Format

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

To choose the results format:

1 In the Query Builder window, select **Options** ⇨ **Options for This Query** . The Query Options window opens.

Note: You can also choose a format for your results from the Properties window.

- In the Project pane or the process flow, right-click the query and select **Properties**. Then click **Results** in the selection pane.

2 If you want to override the settings for query results that are specified on the Query page of the Options window, select **Override the corresponding default settings in Tools->Options**. Then select the format that you want to use in the Result format area.

- **Data table** – saves 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 Egtask library, if that library is defined. Otherwise, the data table is stored in the Sasuser library.

  Note: For information about creating a library, see "Administering SAS Enterprise Guide" in *SAS Enterprise Guide and SAS Add-In for Microsoft Office: Administrator's Guide*.

- **Data view** – saves 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 Egtask library, if that library is defined. Otherwise, the data view is stored in the Sasuser library.

  Note: For information about creating a library, see "Administering SAS Enterprise Guide" in *SAS Enterprise Guide and SAS Add-In for Microsoft Office: Administrator's Guide*.

  Note: This option is disabled if one or more of the tables in the query is a Microsoft Excel file, a slice of an OLAP cube, or a file that is stored in a temporary library.

- **Report** – saves the query results as one or more reports in the formats that you have specified on the Results General page of the Options window: HTML, SAS Report, Listing, Excel, PDF, PowerPoint, or RTF. Query results in this format are not updated until you rerun the query. You cannot run SAS Enterprise Guide tasks against query results in this format.

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 file name that you want to use. If you do not specify the library and file name, the results are saved to the Work library.

Note: For information about creating a library, see "Administering SAS Enterprise Guide" in *SAS Enterprise Guide and SAS Add-In for Microsoft Office: Administrator's Guide*.

To save your results to a specific location

1 In the Query Builder window, click **Change** beside the **Output name** box. The Save File window opens.

   Note: You can also specify a location for your results in the Properties window:

   - In the Project pane or the process flow, right-click the query and select **Properties**. The Properties for query name window opens. Click **Results** in the selection pane and then click **Change**.

2 To save the output data set in a different library, click **Servers** and then navigate to the library that you want to use. You can type a new file name in the **File name** box.

   Note: Only valid SAS data set names are accepted as file names. Valid data set names begin with a letter and contain alphanumeric characters or underscores only. The maximum length of a valid data set name is 32 characters.

3 Click **Save** to save your changes.

Note: To reset your selection back to the previously saved location, select **Options** ⇨ **Options for This Query** and then click **Reset**.

Running a Query

After you complete all the specifications for your query, you can generate your results.

Note: Before you generate the output table, you might want to preview the results in order to ensure that the output is what you want. For more information, see "Previewing the Query" on page 265.

To run a query

- After you specify all the criteria for your query, click **Run** in the Query Builder window. The work area is updated with a *Query-name* tab from which you can access the following items:
  - □ **Last submitted code** – displays the code that was automatically generated by the Query Builder. This code is read-only.

- □ **Log** – displays the log that was generated when the query was run. The log cannot be edited.
- □ **Output Data** or **Results** – displays the output data or report that was generated when the query was run. If the output format was a data table or a data view, you can use that data to run another task.

Note: You can also run a saved query from the Project pane or process flow by right-clicking the query and selecting **Run**.

Note: For information about saving your output, see "Saving Your Results" on page 276.

8

Filtering and Sorting Data

About Filtering and Sorting Data

You can use the Filter and Sort window to modify a single table in SAS Enterprise Guide. When you use the Filter and Sort window, you create an output table that you can use in another query or task. Any changes that you make to the data are not applied to the original data source.

Note: You must have an open project in order to use the Filter and Sort task. For more information, see Chapter 4, “Working with Projects,” on page 27.

Note: If you need to create a more complex query that joins multiple tables together, creates computed columns, or uses prompts, use the Query Builder. For more information, see “About the Query Builder” on page 194.

You can use the Filter and Sort window to modify data in these ways:

- select specific variables to include in the output table
- filter the data based on criteria that you specify
- sort the data
- customize the results

You can also choose to validate and preview the modifications that you are making.

Selecting Variables to Include in the Output Table

When you create an output table by using the Filter and Sort window, you must first select the variables that you want to include. By default, no variables are selected.

Note: To automatically select all of the variables by default, select the **Add columns from input tables to result set of query** option in the Options window. For more information, see "Setting Query Options" on page 500.

To select the variables to include in the output table:

1 Open the data that you want to use and select **Tasks** ⇨ **Browse** on the data grid toolbar. Select **Filter and Sort**.

   Note: You must have an open project in order to use the Filter and Sort task. For more information, see Chapter 4, "Working with Projects," on page 27.

2 Click the **Variables** tab if it is not already selected.

3 From the **Available** list, select one or more variables and click ⇨. To add all of the variables to the output table, click ⇛. The variables are added to the **Selected** list.

   Note: To view the variable labels instead of the variable names, select the **Display labels instead of variable names** check box.

4 Click **OK** to save your changes and create the output table. You can also create a filter, specify sorting criteria for the data, and customize your results.

   Note: The **OK** button is available only after you have selected variables to include in the output.

Filtering Data

About Creating Filters

When you create an output table by using the Filter and Sort window, you can create a filter in order to include only rows that meet certain criteria, based on values in the data. You can use just 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, you can specify whether the relationship between the filter elements is AND or OR.

You can choose to create either a simple filter by using the options in the Filter and Sort window, or you can create your own filter expression by using the Advanced Filter Builder.

Creating a Simple Filter

When you create a simple filter, you can use one variable in the filter, or you can use multiple variables to create several comparison expressions. If you create more than one comparison expression in your filter, then you can specify whether the relationship between the filter elements is AND or OR. Any filter elements that are joined by an AND relationship are evaluated first. The filter elements are not evaluated in the order in which you created them. For example, suppose that you created the following filter:

```
Sex = 'M' or Name = 'Carol' and Date = '1980'
```

In this example, SAS Enterprise Guide first filters the data to find all of the rows that have **Name** equal to **Carol** and **Date** equal to **1980**. Next, the OR relationship is evaluated, and SAS Enterprise Guide adds all rows that have **Sex** equal to **M** to the output table. If you want the filter elements to be evaluated in a different order, use the Advanced Filter Builder to create your filter.

To create a filter by using the Filter and Sort window:

1 After you have selected the variables that you want to include in the output table, click the **Filter** tab.

2 Select a variable from the first drop-down list. The drop-down list contains all of the variables in your data.

Note: To view the variable labels instead of the variable names, select the **Display labels instead of variable names** check box.

3 Select an operator from the second drop-down list.

4 Enter a value in the third box or click ... to display a list of values that you can select for the value of the variable.

Note: If you selected the **Between** or **Not between** operator, then you must specify a beginning and ending value. If you selected the **In a list** or **Not in a list** operator, then you must specify the list of values for the comparison. You can enter or paste values in the box. Click **Add Values** to select a value from the data source. Each value must be on a new line.

For all other operators, by default, the first 250 distinct values in the data source are displayed in the list. If you would like to view more values, click **More Values**. Each time you click **More Values**, the next set of distinct values for the variable are displayed in addition to the initial values. The number of values that are retrieved each time is specified by the **Number of distinct column values to get for each request** option on the Query page of the Options window.

5 Select **AND** or **OR** from the last drop-down list to add additional filter elements to the data. Repeat steps 2–5 to create additional filter elements.

6 If you want the filters to retrieve only rows that match the capitalization of the value that you specify, select **Match case**. This option is selected by default.

7 Click **OK** to save your changes and create the output table. You can also specify sorting criteria for the data and customize your results.

Note: The **OK** button is available only when you have selected variables to include in the output on the **Variables** tab.

Creating an Advanced Filter

You can create an advanced filter by using the Advanced Filter Builder to build the comparison expression, or you can create a simple filter by using the options in the Filter and Sort window.

To create a filter expression:

1 After you have selected the variables that you want to include in the output table, click the **Filter** tab.

2 Click **Advanced Edit** to open the Advanced Filter Builder.

3 You can build your expression by using mathematical operators, functions, columns, and constant values. The constant value can either be a number such as 10 or 3.14, or a string such as 'NC' or 'Sales Numbers'. You can enter these elements in the expression box at the top of the window, or you can use the expression editor to add these elements. For more information, see "Creating an Advanced Expression" on page 283.

Note: When you select the **Use labels for column names** option to display the columns by label, an additional expression box opens to display the expression using the labels. However, you can enter only in the expression box that uses column names. For more information, see “Setting General Data Options” on page 497.

4 Click **OK** to save the filter. The filters that you created are displayed on the **Filter** tab in the Filter and Sort window. You can also specify sorting criteria for the data and customize your results.

Creating an Advanced Expression

To create an advanced expression

- To add operators, click the operator buttons below the expression box. To use advanced operators, such as Between and Contains, expand the **Advanced Operator** drop-down list and click the operator that you want to use. The operators are added to the expression. For more information, see “Using Operators in an Advanced Expression” on page 284.

  Note: The advanced operators are not available if you are creating an expression for a computed column in the Query Builder.

- To add a function, expand the list of functions in the lower left section of the window. You can select a function from the alphabetical list of functions, or you can expand the list of categories and select the function from within a category. When you select a function, a short description of the function appears in the lower right section of the window. Double-click the function to add it to your expression.

  Note: The functions in the Data Quality category require SAS Data Quality Server to be licensed and installed on the server.

- To add a column, expand the list of tables in the lower left section of the window. Expand the table from which you want to add a column. Double-click the column that you want to add to your expression.

  Note: By default, the columns in the list are displayed by name instead of label, and they are listed in the order in which they appear in the data table.

- To change the sort order of the data, right-click the white space around the columns, and then select **Sort Columns** and the option that you want to use.
- To change the column display, right-click the white space around the columns, and then select **Show Columns by Label**.

  Changing how the columns are displayed does not affect how the columns appear in the output table. If you are building an expression from the Query

Builder, you can learn how to customize the data display in the Query Builder and how to specify columns in the output. For more information, see "Customizing Data Display in the Query Builder" on page 200.

- To add a specific value of a column, select the column and click **Get Values** to view all of the unique values of the column. If there are a lot of values to retrieve, you can stop the retrieval by clicking the **Stop** button that appears. To restart the retrieval, click **Get Values** again. Double-click the value that you want to add to your expression, or select one or more values and click **Select Values**.

  Note: For information about using dates in filter expressions, see "Using Dates in Filter Expressions" on page 286. For examples of valid filter expressions, see "Examples: Filtering Expressions" on page 287.

- To validate the expression syntax before you save the filter, click **Validate**. The Validate window opens with a message that indicates whether the expression syntax is valid. If the syntax is not valid, you can view the error log.

  Note: The **Validate** option is not available if you are creating an advanced filter in the join operation in the Query Builder.

Using Operators in an Advanced Expression

When you create a filter expression, you can use the following basic or advanced operators. In the filter syntax, the text for character variables is case sensitive. For example, "Tom" is not equal to "tom" or "TOM".

Table 8.1 *Basic Operators*

Operator	Description
+	Adds x and y.
-	Subtracts y from x.
*	Multiplies x and y.
/	Divides x by y.
**	Raises x to the y exponential power.
||	Concatenates x and y.
(x)	Inserts a value in parentheses.

Operator	Description
'x'	Inserts a value in single quotation marks.
"x"	Inserts a value in double quotation marks.
,	Inserts a comma into the expression.
'abc'n	Enables you to specify variable names that contain characters that are usually restricted by SAS. This option is especially useful if your variable name or value contains DBCS characters. For more information, see "Creating Variable Names in SAS Enterprise Guide" on page 73.
AND	Joins two or more conditions and returns a record when all the conditions are true.
OR	Joins two or more conditions and returns a record when any of the conditions are true.
NOT	Negates the condition in the expression.
=	Relational comparison of x being equal to y.
< >	Relational comparison of x not being equal to y.
<	Relational comparison of x being less than y.
<=	Relational comparison of x being less than or equal to y.
>	Relational comparison of x being greater than y.
>=	Relational comparison of x being greater than or equal to y.

To insert the advanced operators, click **Advanced Operator** and select the operator that you want to include.

Table 8.2 Advanced Operators

Operator	Description
In a list of values	Relational comparison of x to a list of specified values. A record is returned if x matches one of the values in the list.
Not in a list of values	Relational comparison of x to a list of specified values. A record is returned if x does not match one of the values in the list.
Between	Relational comparison of x being between y and z.

Operator	Description
Not between	Relational comparison of x not being between y and z.
Is missing	Relational comparison to determine whether the value of x is missing.
Is not missing	Relational comparison to determine whether the value of x is not missing.
Contains	Relational comparison of x containing y. **Note:** CONTAINS is a SAS keyword and can be used only on SAS data sources. This operator is available only when you open a SAS data source from a server. This operator is not available if the data source is opened from your computer's file system.
Does not contain	Relational comparison of x not containing y.
Matches pattern	Specifies that x matches the specified pattern.
Does not match pattern	Specifies that x does not match the specified pattern.
Sounds like	Uses the sounds-like operator to select values that have a spelling variation of the specified string. The algorithm is English-biased, and is less useful for languages other than English.
Does not sound like	Combines the sounds-like operator with the NOT logical operator to select values that do not contain a spelling variation of the string. The algorithm is English-biased, and is less useful for languages other than English.

Using Dates in Filter Expressions

You can use date values in your filter criteria by using the form "DDMMMYYYY"d. The letters are defined as follows:

- DD is the day of the month in numeric format.
- MMM is a three-letter abbreviation of the month in character format.
- YYYY is the four-digit year in numeric format.

The date value must be enclosed within double quotation marks and the unquoted letter "d" must immediately follow the value. For example, January 19, 2003 would be specified as "19JAN2003"d.

The following examples of formatting dates in filter expressions assume that you have two fields, Birthday and Anniversary, both of which display date values.

- The following filter expression displays all rows where the birthday is January 1, 1973 or later.

  ```
  Birthday >= "01JAN1973"d
  ```

- The following filter expression displays all rows where the anniversary is June 20, 1988 or earlier.

  ```
  Anniversary <= "20JUN1988"d
  ```

Examples: Filtering Expressions

The following examples of valid filtering expressions assume that there are several variables: character variables called NAME, KIND, and TITLE, and numeric variables called AGE, RETIREMENT, and OFFSET. The values of character variables are in text format, and the values of numeric variables are typically in number format, although date variables are an exception. For more information about formatting dates, see "Using Dates in Filter Expressions" on page 286.

Note: The text for character variables is case sensitive. For example, "Tom" is not equal to "tom" or "TOM".

This statement selects all rows where the value for the variable AGE is less than 30.

```
age < 30
```

This statement selects rows where the NAME contains the text "Tom" and the value of AGE is greater than 30.

```
name contains "Tom" and age > 30
```

This statement shows that you can include expressions. This divides the value of RETIREMENT by 2 and adds the value of OFFSET. Rows are selected where the value of AGE is less than this calculated value.

```
age < retirement / 2 + offset
```

This statement shows how parentheses are used to change the order of evaluation; the value of AGE is divided by the value of RETIREMENT, and the result is multiplied by 100. Rows are selected if this result is greater than 75. The rows that are returned show the employees who have completed 75% of their work life.

```
( age / retirement ) * 100 > 75
```

This statement compares the value of OFFSET to either the value of AGE or the value of RETIREMENT. Whether KIND contains the text "F" determines whether the value of OFFSET is compared to AGE or RETIREMENT. The ?: operator set enables you to create a subexpression to be conditional on the result of a test or value. A ?: subexpression takes the place of a single item; if the test value is itself a subexpression (as shown earlier) it must be enclosed in parentheses.

```
offset > ( kind contains "F" ) ? age : retirement
```

This expression bases its test on whether the value of KIND contains a C. If it does, then NAME is compared to TITLE. If the value of KIND does not contain a C, then the value of AGE is compared to the value of RETIREMENT. This actually causes the expression to compare the text or numeric values of the variables, depending on the state of a variable in the row.

```
(( kind contains "C" ) ? name : age ) > (( kind contains "C")? title : retirement )
```

Sorting Your Data

When you create an output table by using the Filter and Sort window, you can sort the data by one or more columns from the original data source.

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

To sort your data:

1 After you have selected the variables that you want to include in the output table, click the **Sort** tab.

2 From the **Sort by** drop-down list, select the variable that you want to use to sort the rows.

3 Select whether you want to sort the values for the variable in ascending or descending order.

4 To sort by additional variables, select another variable from the **Then by** drop-down list and select whether you want to sort the values for the variable in ascending or descending order. Repeat this step for all of the remaining variables that you want to use to sort the data.

5 Click **OK** to save your changes and create the output table. You can also create a filter for the data and customize your results.

Note: The **OK** button is available only when you have selected variables to include in the output on the **Variables** tab.

Customizing Your Results

When you create an output table by using the Filter and Sort window, you specify a unique name for the task, so it is easy to find in your SAS Enterprise Guide project. You can also specify where to save the output table that is generated by the task.

To customize your results:

1. After you have selected the variables that you want to include in the output table, click the **Results** tab.
2. In the **Task name** box, enter the task name that you want to appear in the Project pane and process flow of your project. By default, the task name is **Filter and Sort**.
3. In the **Output name** box, specify a name for the output table that is created by the task. The name for the output is in the *SASLibrary.Table-name* format. By default, the output table is saved in the Work library. To select a location for the output table, click **Change**.
4. Click **OK** to save your changes and create the output table.

Note: The **OK** button is available only when you have selected variables to include in the output on the **Variables** tab.

Validating the Query Syntax

You can validate the query syntax that is automatically generated by the Filter and Sort window. The results of the validation are displayed in a separate window.

The validation log displays any error information that might be generated by the query as well as any implicit pass-through tracing information, if the query uses SQL pass-through. Examples of query syntax that might generate errors include creating an output table with no variables or creating a filter on an invalid condition. If the query was passed to a database that cannot process a function that was used in the query, then the validation process might return a syntax error. However, if SAS can process the function, then the query can still run without an error. After the query has been validated, you can decide whether you want to run the query.

To validate the query syntax:

1 In the Filter and Sort window, select the variables that you want to include in your output table, specify any filters or sorting criteria that you want to use, and customize your results, if desired.

2 Click **Validate**. The Validate window displays the query syntax along with any syntax errors.

Previewing the Output Table

You can preview the output table that you are creating in the Filter and Sort window. You can show or hide the preview pane, and you can specify the maximum number of columns and rows to display in the preview pane. You can also update the preview pane when you make changes in the Filter and Sort window.

To preview the output table:

1 In the Filter and Sort window, select the variables that you want to include in your output table, and specify any filters or sorting criteria that you want to use.

2 Click **Show Preview**. The preview pane opens.

3 (Optional) To change the maximum number of columns and rows that are displayed in the preview pane, specify the number that you want to display in the **Max columns** and **Max rows** boxes.

4 (Optional) To update the results that are displayed in the preview pane, click **Update**. The results are updated to reflect any changes that you have made to options in the Filter and Sort window.

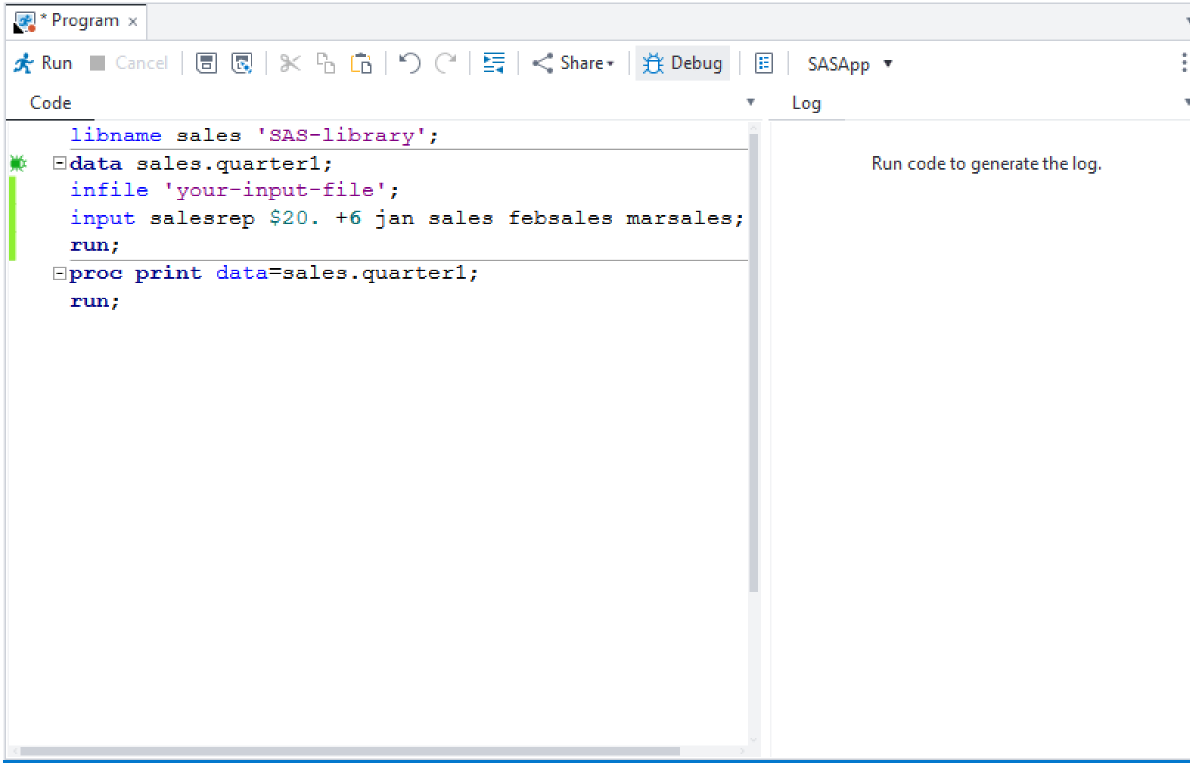
9

Working with Programs

About the Program Editor

SAS Enterprise Guide provides a color-coded, syntax-checking SAS language editor for editing new or existing SAS programs in your project. The editor includes a wide variety of features such as autocompletion, automatic formatting, pop-up help, and parenthesis matching. You can use the Program Editor to submit code for the SAS 9 programming environment or the SAS Viya programming environment.

This example shows syntax for the SAS 9 programming environment. In this program, you define a library, create a SAS data set, and create a report that lists the data.



```
libname sales 'SAS-library';
data sales.quarter1;
infile 'your-input-file';
input salesrep $20. +6 jan sales febsales marsales;
run;
proc print data=sales.quarter1;
run;
```

With the Program Editor, you can write, run, and save SAS programs. You can also save and modify the code that is automatically generated when you create a task or query.

If you are not familiar with SAS 9 programming, see *Step-by-Step Programming with Base SAS*.

For information about SAS Viya programming, see *SAS Cloud Analytic Services: User's Guide*.

Note: To work with the SAS Viya, you must create a connection. For more information, see "Connecting to the SAS Viya Environment" on page 7.

Opening and Creating Programs

Opening a Program

You can open a SAS program in SAS Enterprise Guide in several ways. You can drag the files from the Servers tab of the navigation pane or from Windows Explorer, or you can use the menus.

Note: When you insert an existing program as an item in your project, the item remains linked to your original program file. Any changes that you make to the program in your project are applied to the original program file. You can use the Properties for *program-name* window to embed the program in your project so that your changes are not applied to the original program file. For more information, see Table 9.2 on page 338.

To add a program by using the menu:

1. Select **Program** ⇨ **Open program**, and then browse to find the program.

   Note: You can also open a program by selecting **File** ⇨ **Open**.

2. Select the program that you want to open and click **Open**. The program opens in the SAS Enterprise Guide work area.

   Note: You can also open programs that you have recently viewed. To open one of these programs, select **File** ⇨ **Open Recent Items** ⇨ ***program-name***.

To add a program by dragging and dropping:

1. If you are dragging program files from Windows Explorer to SAS Enterprise Guide, position the Explorer window so that you can see both it and either the SAS Enterprise Guide Project pane or the process flow. Both of these must be visible before you can drag files between them. If you do not have a project open, you can drag the program to an open program tab.
2. Use the left mouse button to drag the file from the Explorer or the Servers tab into the Project pane, process flow, or program tab in SAS Enterprise Guide. The mouse pointer changes to indicate where you can drop the file.

Create a Program

You can create SAS programs from within SAS Enterprise Guide. The new program file is automatically added to the current project, if you have one open.

Select **Program** ⇨ **New program**. The new program opens in the work area.

Note: You can also create a program by selecting **File** ⇨ **New** ⇨ **Program**.

Understanding the Components of a SAS 9 Program

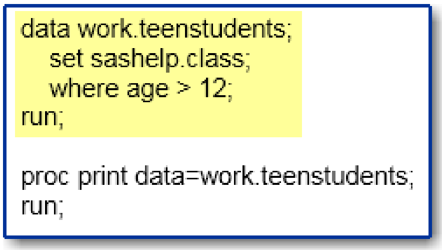
A SAS program is a sequence of steps that you submit to SAS for execution. Each step in the program performs a specific task. Only two types of steps make up SAS 9 programs: DATA steps and PROC steps.

A SAS program can contain a DATA step, a PROC step, or any combination of DATA steps and PROC steps. The number and type of steps depend on what tasks you need to perform.

A DATA step enables you to manage and manipulate your data. You typically use a DATA step to read data from an input source, process it, and create a SAS table. With DATA steps, you can perform these tasks:

- put your data into a SAS table
- compute the values for new variables
- check for and correct errors in your data
- produce new SAS data sets by subsetting, merging, and updating existing data sets

In the following sample program, the DATA step creates a SAS table TEENSTUDENTS in the temporary Work library by using the existing SAS table CLASS in the Sashelp library. The program includes only students who are older than 12 in the new SAS table.



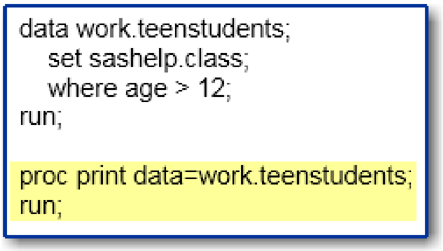
```
data work.teenstudents;
   set sashelp.class;
   where age > 12;
run;

proc print data=work.teenstudents;
run;
```

A PROC (procedure) step enables you to analyze and present your data. You can use a PROC step to generate reports and graphs. PROC steps include a library of prewritten routines (procedures) that perform tasks on SAS tables, such as listing, sorting, and summarizing data. With PROC steps, you can perform these tasks:

- print a report
- produce descriptive statistics
- create a tabular report
- produce plots and charts

The sample program uses the PRINT procedure, which lists the data in a SAS table. The DATA= option tells SAS what data to use for the procedure. Here is a sample SAS program with the PROC step highlighted.



```
data work.teenstudents;
   set sashelp.class;
   where age > 12;
run;

proc print data=work.teenstudents;
run;
```

For information about SAS Viya programming, see *An Introduction to SAS Viya Programming*.

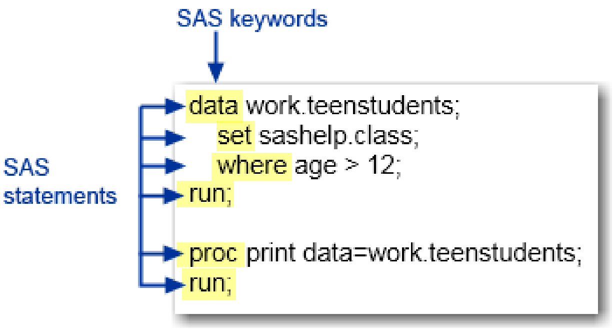
Writing a SAS 9 Program

SAS 9 programs consist of SAS statements. A SAS statement has two important characteristics:

- It usually begins with a SAS keyword.
- It always ends with a semicolon.

Remember that SAS programs include two types of steps: DATA steps and PROC steps. DATA steps begin with the keyword DATA, and PROC steps begin with the keyword PROC. Generally, a step ends with a RUN statement or when a new DATA or PROC step begins. A RUN statement tells SAS to process all of the preceding lines of the step.

Each step in a SAS program consists of a sequence of statements. SAS statements usually begin with an identifying keyword, and they always end with a semicolon.



```
data work.teenstudents;
   set sashelp.class;
   where age > 12;
run;

proc print data=work.teenstudents;
run;
```

SAS statements are free-format:

- They can begin and end anywhere on a line.
- One statement can continue over several lines.
- Several statements can be on one line.

To create a program in SAS Enterprise Guide, select **Program** ⇨ **New program**.

Extracting Code from a Task

You can extract the code that is generated automatically when you run a SAS Enterprise Guide task. After you extract the code, you can make changes to the program and run it.

Note: The extracted code contains a copy of the task program and is no longer associated with the original task. Editing this program does not affect the task, and modifying the original task does not affect the contents of the extracted code.

To extract the code from a task, right-click the task in the Project pane or in the process flow and select **Extract code**. A new program item named **Code For *task-name*** is added to the project. You can make and save changes to the new program.

Note: You can also edit the code from a task by opening the task after you have run it and clicking the **Last submitted code** tab. Enter any character in the program, and you are prompted to create a copy of the code that you can modify.

Understanding Program History

About Program History

The program history feature enables you to track the changes that you and other programmers make to programs in SAS Enterprise Guide. As you write a program, you can commit the changes that you are making and create interim versions of the program that are saved in the program history.

Note: For information about managing version control in SAS Enterprise Guide using Git repositories, see Chapter 15, “Understanding Git Integration in SAS Enterprise Guide,” on page 461.

You can use your program history to compare two different committed versions of a program, and you can compare a committed version of the program to the code that is currently in the Program Editor. You can also revert the code in the Program Editor to a previous version of code.

Note: To use the program history feature, you must complete these steps:

- Create or open a project. For more information, see Chapter 4, “Working with Projects,” on page 27.
- Select the **Enable program history** option. For more information, see “Setting Version Control Options” on page 513.

Getting Started with Program History

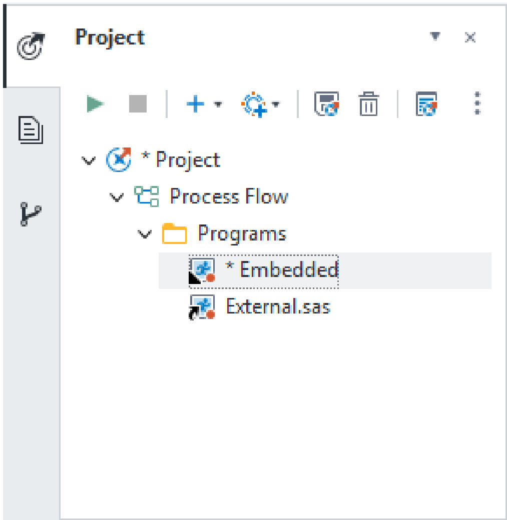
Understanding Embedded Programs

Your programs in SAS Enterprise Guide can be either embedded in a project or externally referenced. To use the SAS Enterprise Guide program history feature, your program must be embedded in a project.

Embedded programs are stored inside a SAS Enterprise Guide project. When the project is closed, an embedded program cannot be accessed or edited. When you create a program in SAS Enterprise Guide, it is embedded in your project until you save it to an external location. You can embed existing external programs in your project.

An externally referenced program is a program that is saved on your local computer or a remote server. When you open an existing program in SAS Enterprise Guide, the path name of the program is stored in the SAS Enterprise Guide project. When you make changes to the program and save them, SAS Enterprise Guide writes the changes to the program. An external program is accessible when SAS Enterprise Guide is closed.

You can determine whether a program is embedded or external by looking at the program icon in the Project pane. External program icons have an arrow in the bottom left corner as shown here:



- - External program icon
- - Embedded program icon

You can also view the program properties to determine whether a program is embedded or external and to change how a program is stored.

To view how a program is stored, right-click the program in the Project pane and select **Properties**. In the Properties for *program-name* window, the **File path** field displays the text (Embedded In Project) if the program is embedded or the file path name if the program is external.

To change how a program is stored:

1 Right-click the program in the Project pane and select **Properties**.

2 To save an embedded program as an external program, click **Save As** and specify the location where you want to save the file. Click **Save**. The program code is no longer stored in the project.

  To save an externally referenced program as an embedded program, click **Embed**. The program no longer references the external file.

3 Click **OK** to save your changes.

Committing Your Program Changes

You can save interim versions of a program by committing the program. When you commit your program, you create a version of the program in program history that you can refer to. It can be useful to commit your program changes at meaningful intervals in your program development so that you can revert to previously committed versions of your program if you need to. For more information, see “Reverting a Program to a Previous Version” on page 304.

Note: You can commit only programs that are embedded in your project. For more information, see “Understanding Embedded Programs” on page 298.

Committing a program is different from saving a program. When you commit a program, you are storing a snapshot of the current program, which you can use for comparison with other versions and to revert to, if necessary. When you save a program, you are overwriting the last saved version without any information about the previous changes.

You can commit a single program, and you can also commit multiple programs at once.

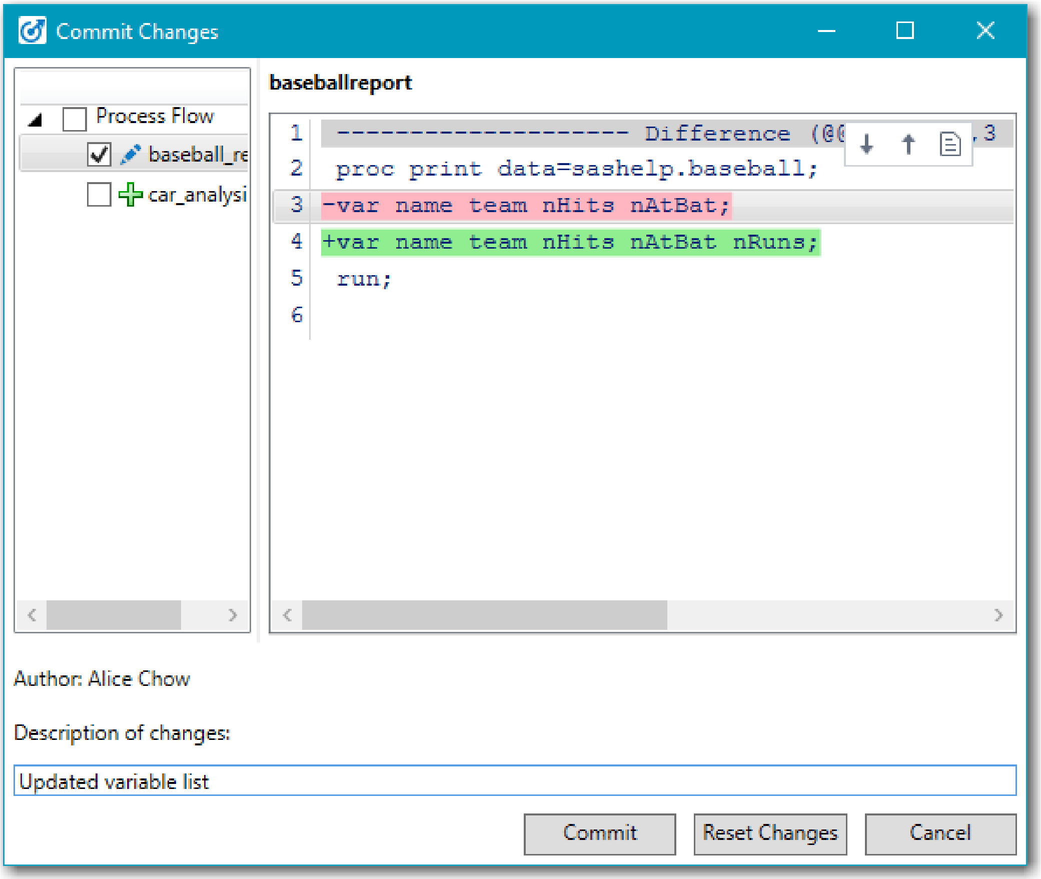
To commit a program:

1 On the Program Editor toolbar, select **Program History** ⇨ **Commit**. The **Commit Changes** displays a list of all new and updated embedded programs for each process flow in your project. When you view the list of programs, note the following information:

- New programs that have not been committed yet are displayed with the ✚ program icon.
- Updated programs are displayed with the ✎ icon.
- You can click a program in the list to view the changes for that program.
- If you have a new or updated embedded program open in the Program Editor, that program is selected by default.
- Deleted lines are highlighted in red, and new lines are highlighted in green.

Note: To use the program history feature, you must complete these steps:

- Create or open a project. For more information, see Chapter 4, "Working with Projects," on page 27.
- Select the **Enable program history** option. For more information, see "Setting Version Control Options" on page 513.



baseballreport

```
--------------------- Difference (@@ ,3
proc print data=sashelp.baseball;
-var name team nHits nAtBat;
+var name team nHits nAtBat nRuns;
run;
```

Author: Alice Chow

Description of changes:

Updated variable list

2 Select the programs that you want to commit.

3 To specify a description of the changes that you are committing, enter the description in the **Description of changes** box.

4 Click **Commit**.

Viewing Uncommitted Changes

Before you commit a program, you can view changes that you have made to the program since the last time it was committed. SAS Enterprise Guide includes a built-in comparison tool that enables you to view the changes in one program at a time.

Note: You can also specify an external file comparison tool to compare your programs by using the File Comparison options. For more information, see "Setting Diff and Merge Tools Options" on page 512.

To view uncommitted changes to a program, on the Program Editor toolbar, select **Program History** ⇨ **Changes**. The Uncommitted Changes in *program-name* window displays the changes in the program compared to the last time it was committed.

Note: If you have specified an external file comparison tool, then that tool opens instead of the built-in tool.

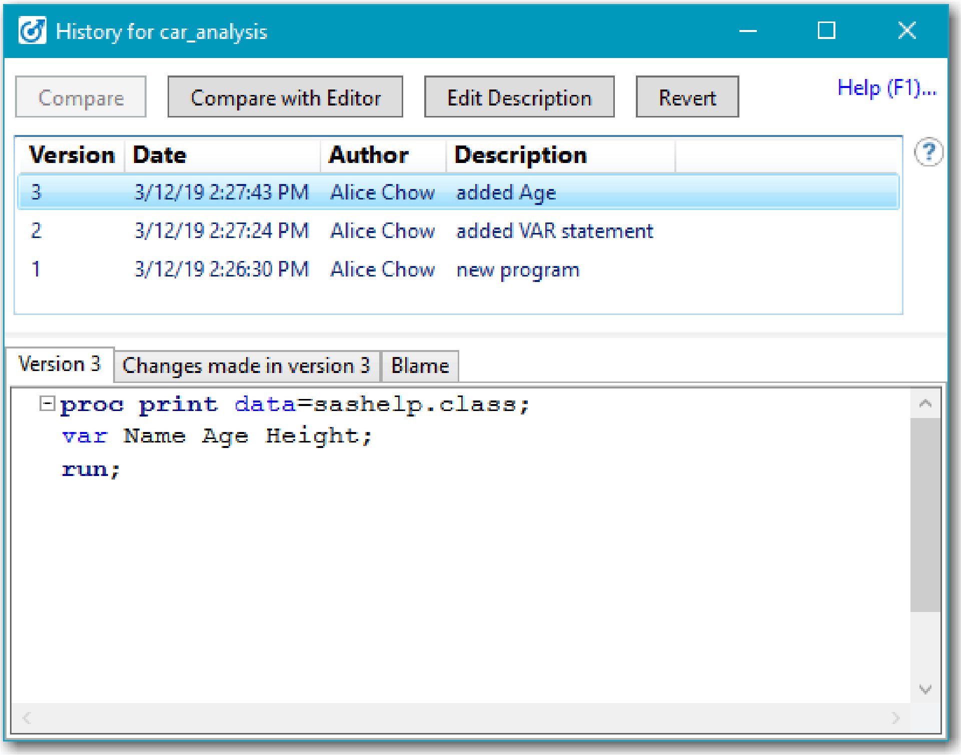
Viewing Individual Program History

Using the Program History Window

When you view the history of a single program, you can see a list of the different versions of the program that were committed along with the date, author, and a description of each version. You can use the built-in file comparison tool to compare different versions of a program. You can also specify an external comparison tool by using the File Comparison options. For more information, see "Setting Diff and Merge Tools Options" on page 512.

View Code for a Selected Version

To view the code for a specific version of a program, select **Program History** ⇨ **History** on the program toolbar to open the History for *program-name* window, and select the version that you want to view. The code is displayed on a tab for that version in the lower half of the window. By default, the latest version of the program is selected.



Version	Date	Author	Description
3	3/12/19 2:27:43 PM	Alice Chow	added Age
2	3/12/19 2:27:24 PM	Alice Chow	added VAR statement
1	3/12/19 2:26:30 PM	Alice Chow	new program

```
proc print data=sashelp.class;
 var Name Age Height;
 run;
```

Compare Two Versions of a Program

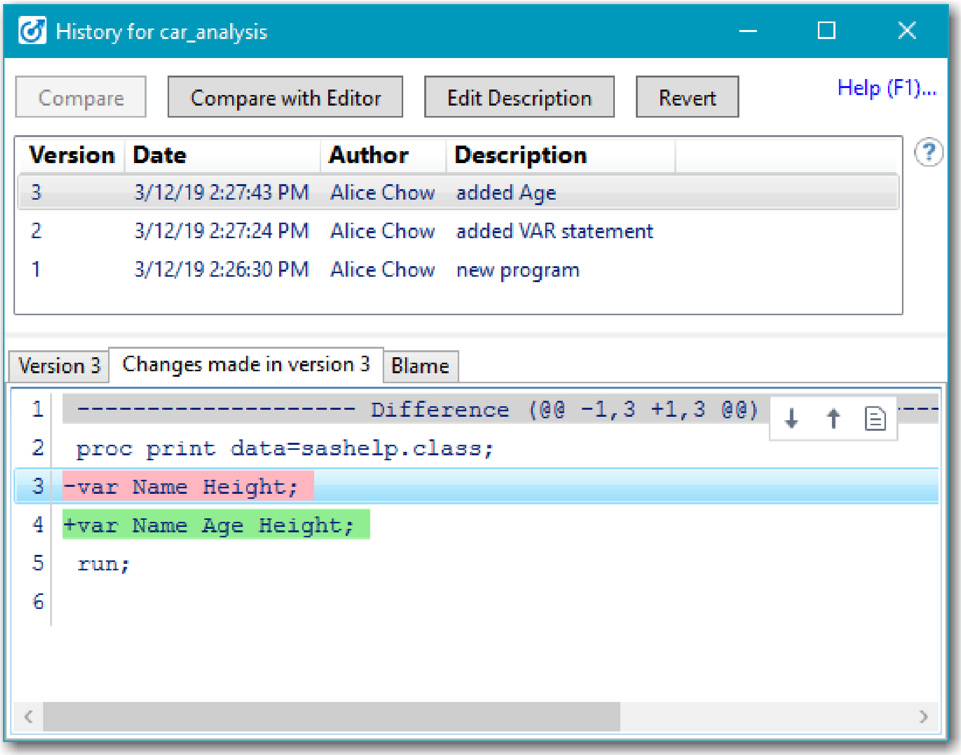
If you have committed more than one version of your program, you can use the program history to compare different versions of the same program. By default, if you have more than one version of the program committed, the latest version is selected and compared with the previous version by using the built-in comparison tool in SAS Enterprise Guide. You can also select the two versions that you want to compare.

Note: If you want to use another file comparison tool, you can specify an external file comparison tool in the Options window. For more information, see “Setting Diff and Merge Tools Options” on page 512.

To compare two versions of a program:

1 select **Program History** ⇨ **History** on the program toolbar to open the History for *program-name* window.

2 Select the two versions of the program that you want to compare. The comparison is displayed on the **Changes made in version x** tab in the lower half of the window. Deleted lines are highlighted in red, and inserted lines are highlighted in green.



Version	Date	Author	Description
3	3/12/19 2:27:43 PM	Alice Chow	added Age
2	3/12/19 2:27:24 PM	Alice Chow	added VAR statement
1	3/12/19 2:26:30 PM	Alice Chow	new program

```
 -------------------- Difference (@@ -1,3 +1,3 @@) ---
 proc print data=sashelp.class;
-var Name Height;
+var Name Age Height;
 run;
```

Note: If you have specified an external file comparison tool, then you can select the versions that you want to compare and click **Compare**. The comparison tool opens so that you can compare the versions. The **Compare** button is enabled only when you have specified an external comparison tool.

Compare a Version of a Program with the Program Editor

To compare a version of a program with the code currently in the Program Editor, you must first specify an external file comparison tool by using the Diff and Merge Tools options. For more information, see “Setting Diff and Merge Tools Options” on page 512.

To compare a version with the Program Editor:

1 Select **Program History** ⇨ **History** on the program toolbar to open the History for *program-name* window.

2 Select the version that you want to compare and click **Compare with Editor**. The comparison tool opens so that you can compare the versions of code.

Edit the Description of a Program Version

To edit the description:

1 Select **Program History** ⇨ **History** on the program toolbar to open the History for *program-name* window.

2 Select the version whose description you want to edit and click **Edit Description**. Enter the new description.

Reverting a Program to a Previous Version

You can replace the code in the Program Editor with the code from a previously committed version.

1 Select **Program History** ⇨ **History** on the program toolbar to open the History for *program-name* window.

2 Select the version that you want to use and click **Revert**. The code in the Program Editor is updated.

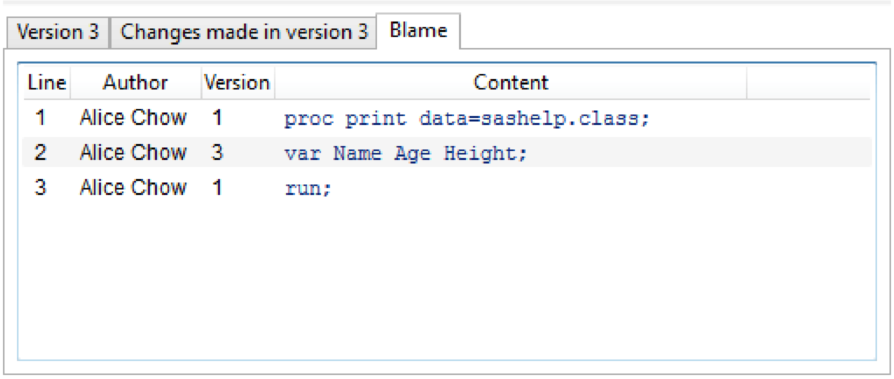
View Code Modifications

When you view the history of a program, you can use the **Blame** tab to view each line of code, the user who last modified that line, and in which version the code was last modified.

To view the code modifications:

1 Select **Program History** ⇨ **History** on the program toolbar to open the History for *program-name* window.

2 Click the **Blame** tab to display all of the modifications for the selected program.



Line	Author	Version	Content
1	Alice Chow	1	`proc print data=sashelp.class;`
2	Alice Chow	3	`var Name Age Height;`
3	Alice Chow	1	`run;`

Editing Program Files

About Editing Program Files

You can use the SAS Enterprise Guide Program Editor to create programs and to edit existing SAS programs in your project. The Program Editor offers several features to help you reduce your programming time, including autocompletion for hundreds of SAS statements and procedures and options. The Program Editor also provides parenthesis matching, built-in syntax and function help, and automatic code formatting.

By default, when you select or search for a word in the Program Editor, all other occurrences of that word are also highlighted. You can disable or customize the smart highlighting feature by using the Editor options.

You can also analyze the contents of a program to create a process flow or optimize the program steps to run in parallel in a grid computing environment.

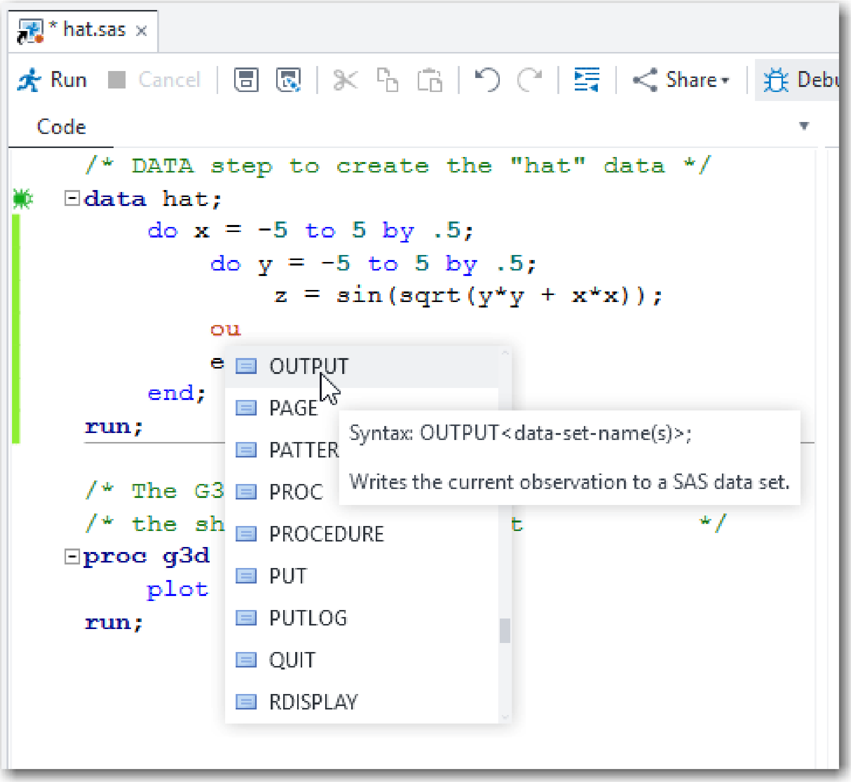
Using the Autocomplete Feature

The autocomplete, or code completion, feature in the Program Editor can predict the next word that you want to enter before you enter it completely. The autocomplete feature can complete keywords that are associated with SAS statements, procedures, macros, functions, CALL routines, formats, informats, librefs or libraries, SAS data sets, macro variables, editor snippets, SAS colors, style elements, style attributes, statistics keywords, and various SAS statement and procedure options, including enumerated option values.

The autocomplete feature can also complete the names of SAS data sets and column names. The autocomplete feature includes nonstandard column names, such as "Total Retail $", when the default VALIDVARNAME=ANY system option is used and nonstandard data set names, such as "March Sales Data", when the VALIDMEMNAME=EXTEND system option is used. The VALIDMEMNAME option is available only in SAS 9.3 or later.

You can also manually open the autocompletion window by using shortcut keys. For more information, see "SAS Program Editor Shortcuts" in *SAS Enterprise Guide: Accessibility Features*.

Note: To view a definition of a keyword, position the mouse pointer over the keyword in the pop-up window.



```
/* DATA step to create the "hat" data */
data hat;
    do x = -5 to 5 by .5;
        do y = -5 to 5 by .5;
            z = sin(sqrt(y*y + x*x));
        ou
        e
    end;
run;

/* The G3
/* the sh                    t              */
proc g3d
    plot
run;
```

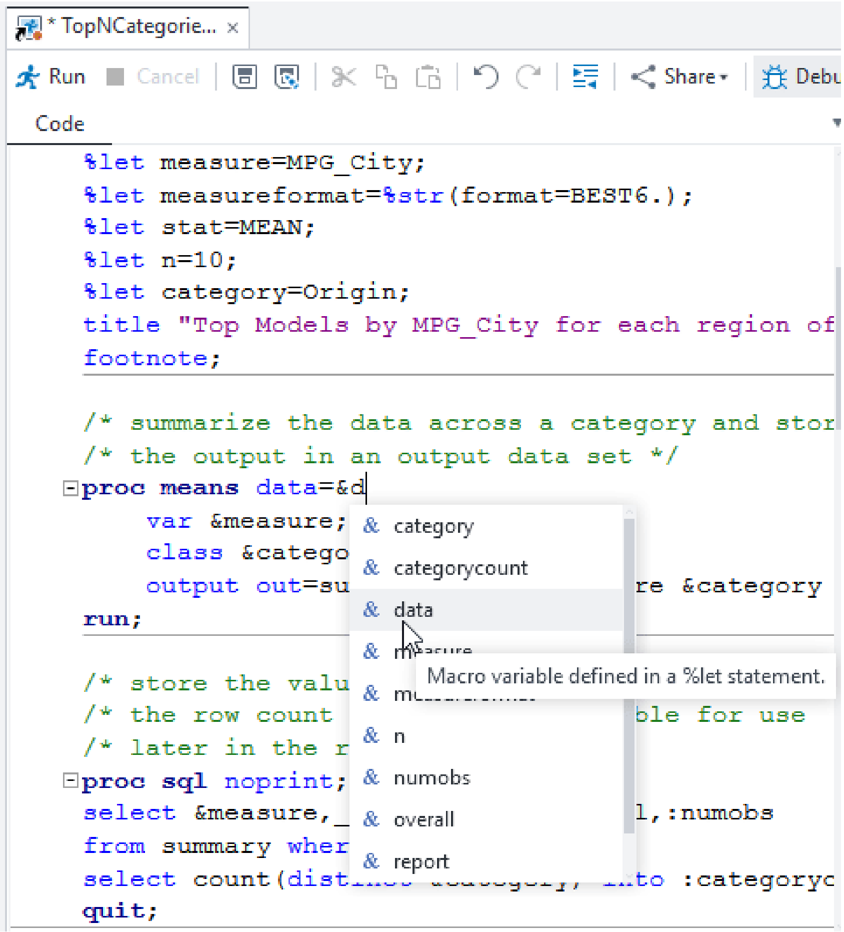
Syntax: OUTPUT<data-set-name(s)>;

Writes the current observation to a SAS data set.

The Program Editor can also read your current program and suggest syntax for these program elements:

- macro variables that are defined by using the %LET statement or SYMPUT CALL routine
- macro routines that are defined by using the %MACRO statement
- data set names that are defined by using the DATA step statement

Note: The Program Editor does not automatically list macro variables and routines that are defined outside the current document (for example, external macro programs, %include files, and autoexec files).



```
%let measure=MPG_City;
%let measureformat=%str(format=BEST6.);
%let stat=MEAN;
%let n=10;
%let category=Origin;
title "Top Models by MPG_City for each region of
footnote;

/* summarize the data across a category and stor
/* the output in an output data set */
proc means data=&d
     var &measure;
     class &catego
     output out=su                   re &category
run;

/* store the valu
/* the row count                     ble for use
/* later in the r
proc sql noprint;
select &measure,_                    1,:numobs
from summary wher
select count(dist                    into :categoryc
quit;
```

& category
& categorycount
& data
& measure
& me
& n
& numobs
& overall
& report

Macro variable defined in a %let statement.

Note: You can disable or customize the autocomplete feature by using the Autocomplete options. For more information, see “Setting Autocomplete Options” on page 510.

To use the autocomplete feature:

1 In the Program Editor, enter the first one or more letters of the word that you want to use. A window opens with a list of suggested keywords that begin with those letters.

2 You can navigate to the keyword that you want to use in these 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 or the Page Up and Page Down keys.
   - Scroll through the list by using your mouse scroll wheel.

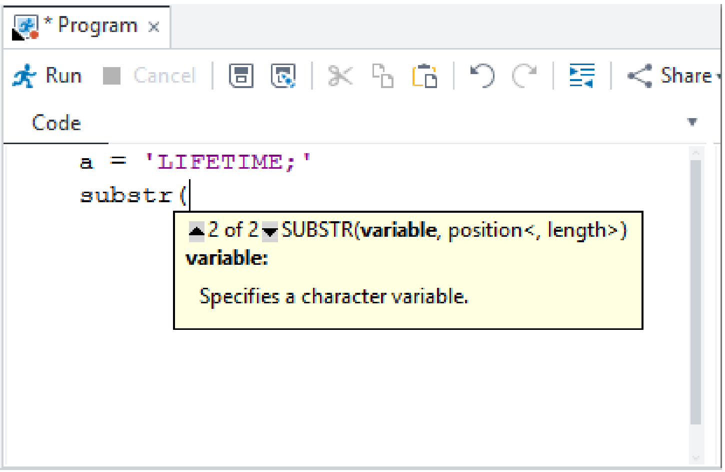
3 You can add the keyword to your program in several ways:

   - Double-click the selected keyword.
   - Press the Spacebar, Enter, or Tab key.
   - Enter the next valid punctuation, such as the semicolon, period, or equal sign keys.

Understanding Function Disambiguation

Many SAS functions have different versions of their syntax to accept different arguments depending on how they are being used in your program. For example, the SUBSTR function can be used on either the left side or the right side of an assignment statement based on what you are trying to accomplish. The Program Editor can help you resolve this ambiguity by presenting both versions of the syntax and enabling you to select between them.

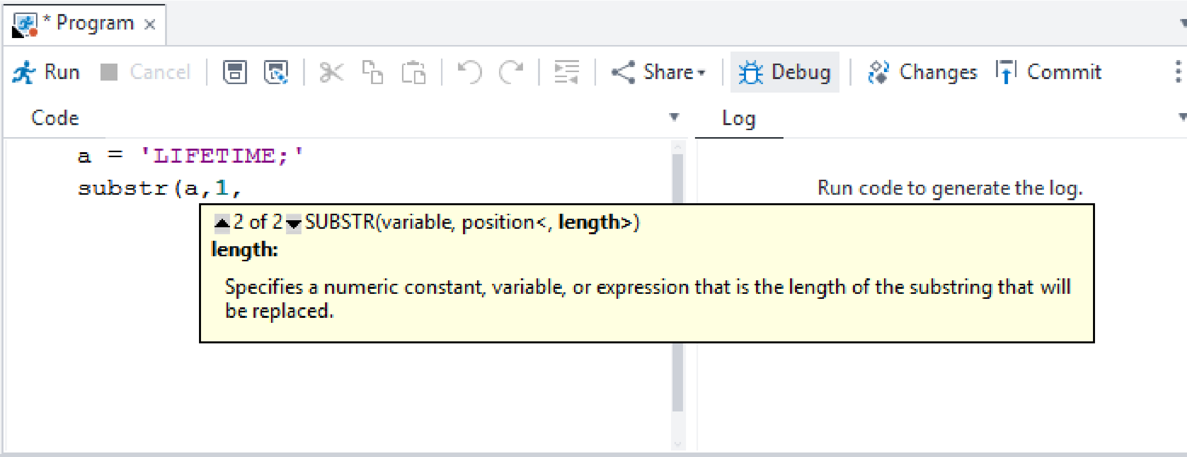
In the following example, the Program Editor displays the syntax for the SUBSTR function as it is used on the left side of the assignment statement. However, you can click the up or down arrows in the pop-up help window to display the syntax for the function when it is used on the right side of the assignment statement.



```
a = 'LIFETIME;'
substr(
```

▲2 of 2▼SUBSTR(**variable**, position<, length>)
variable:
Specifies a character variable.

The Program Editor then helps you complete the function reference by tracking the position of each argument as you type and by providing a hint about the purpose of each argument as you need it. In the following example, the Program Editor displays Help for the third argument in the SUBSTR function because you have already entered the first two arguments.



```
a = 'LIFETIME;'
substr(a,1,
```

Run code to generate the log.

▲2 of 2▼SUBSTR(variable, position<, **length>**)
length:
Specifies a numeric constant, variable, or expression that is the length of the substring that will be replaced.

Using the Syntax Help

The Program Editor displays brief SAS syntax documentation as you write and edit your programs. You can display the Help in any of these ways:

- Position the mouse pointer over a valid SAS keyword in your program.
- Position the cursor within a valid SAS keyword and press F1.
- Start entering a valid SAS keyword, and then position the mouse pointer over a suggested keyword in the autocompletion window.

The SAS language often uses keywords that have the same name but can be used in different contexts. For example, the LABEL keyword is used both within the LABEL statement and also as part of the LABEL= data set option. The Program Editor can distinguish between these uses and display the appropriate Help.

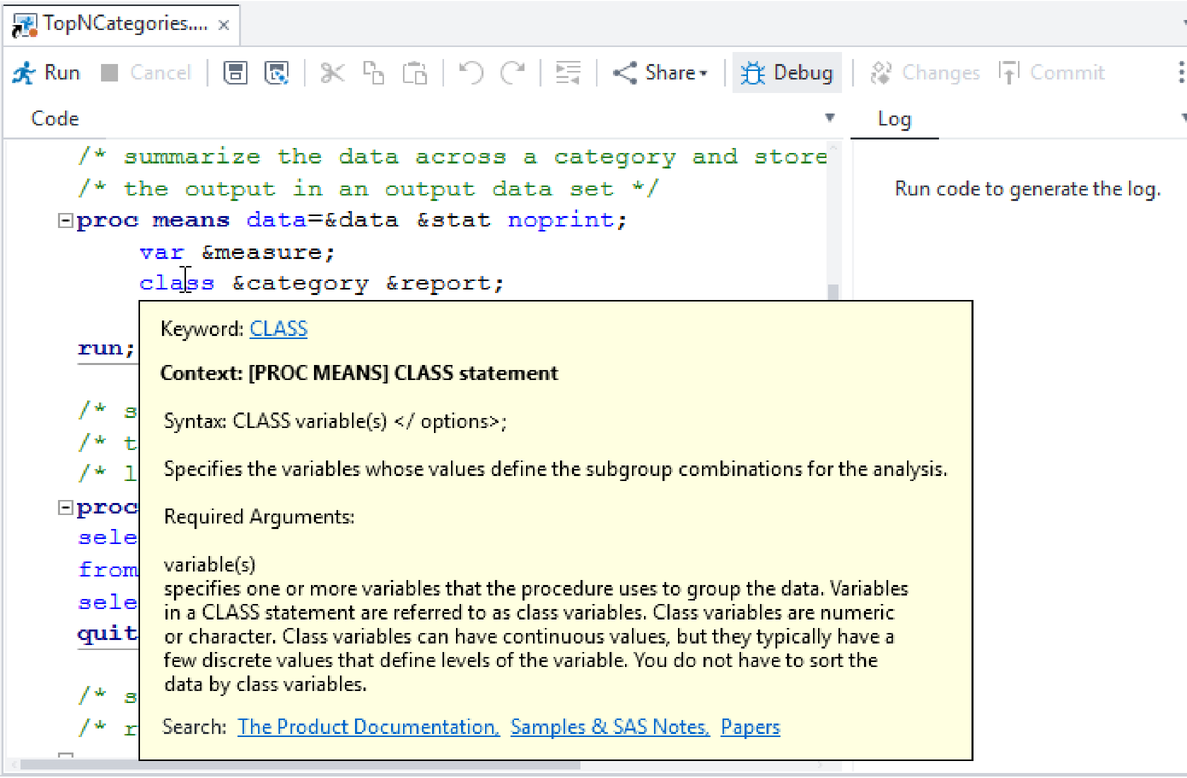
The SAS Product Documentation provides more comprehensive usage information about the SAS language, but the syntax Help in the Program Editor can get you started with a hint about the syntax or a brief description of a keyword.

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 support.sas.com for the keyword. You can customize the search engine that is used in the search by using the **Autocomplete** tab of the Enhanced Editor options. For more information, see "Setting Autocomplete Options" on page 510.
- 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.

Note: You can also search the documentation by using shortcut keys. For more information, see "SAS Program Editor Shortcuts" in *SAS Enterprise Guide: Accessibility Features*.

In the following example, the Program Editor displays Help for the CLASS option in the PROC MEANS statement.



```
/* summarize the data across a category and store
/* the output in an output data set */
proc means data=&data &stat noprint;
    var &measure;
    class &category &report;

run;

/* s
/* t
/* l
proc
sele
from
sele
quit

/* s
/* r
```

Keyword: CLASS

Context: [PROC MEANS] CLASS statement

Syntax: CLASS variable(s) </ options>;

Specifies the variables whose values define the subgroup combinations for the analysis.

Required Arguments:

variable(s)
specifies one or more variables that the procedure uses to group the data. Variables in a CLASS statement are referred to as class variables. Class variables are numeric or character. Class variables can have continuous values, but they typically have a few discrete values that define levels of the variable. You do not have to sort the data by class variables.

Search: The Product Documentation, Samples & SAS Notes, Papers

Automatically Formatting Code

You can use the Program 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.

Note: You can customize much of the formatting behavior by using the Indenter options. For more information, see "Setting Indenter Options" on page 511.

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;
```

To format your code, open the program that you want to format in the Program Editor. Click ≣ in the program toolbar, or right-click anywhere in the Program Editor and select **Format code**.

Note: You can undo the automatic formatting by selecting **Edit** ⇨ **Undo** or by pressing Ctrl+Z.

Splitting the Program Editor

You can split the Program Editor so that you can view and edit different parts of a large program at the same time. You can choose to split the window into two stacked windows, two side-by-side windows, or four windows.

To split the Program Editor, open the program that you want to edit. Right-click anywhere in the Program Editor and select **Split**. You can choose from these options:

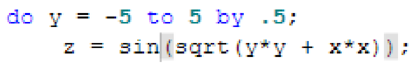
- **None** displays the Program Editor as one window.
- **Stacked** displays the Program Editor as two windows that are displayed vertically, with one window on top of the other.
- **Side By Side** displays the Program Editor as two windows that are displayed horizontally, with the two windows side by side.
- **Both** displays the Program Editor as four windows.

Note: You can also split the window by selecting **Program** ⇨ **Split**.

You can resize the two windows by positioning the mouse pointer over the separator bars between the windows until the pointer becomes a double arrow. Then drag the pointer upward or downward until the windows are the size that you want.

Using Parenthesis Matching

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



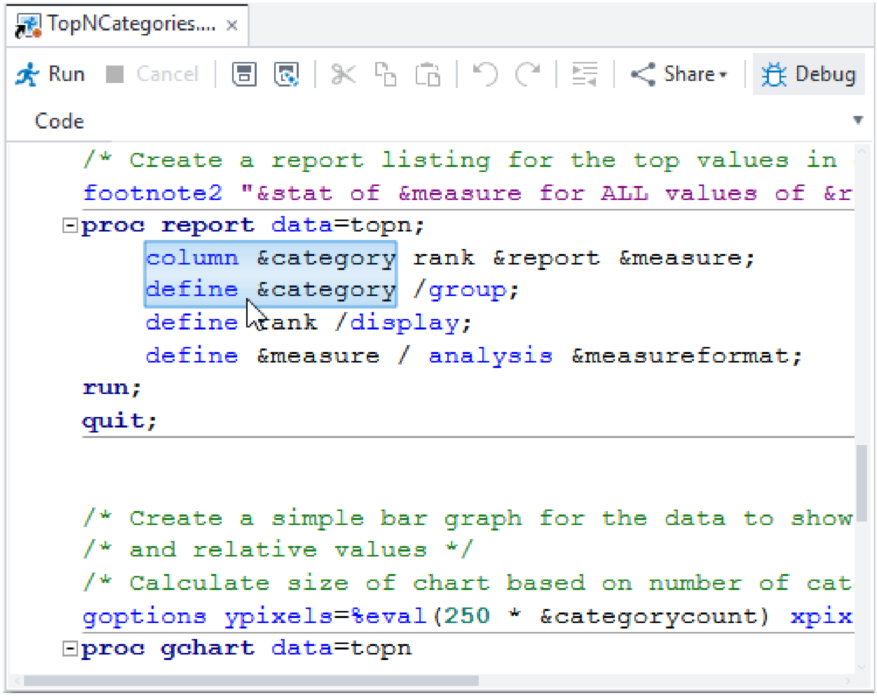
```
do y = -5 to 5 by .5;
    z = sin(sqrt(y*y + x*x));
```

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 in gray. If the parenthesis does not have a match, it is not highlighted.

Select Columns of Text

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, press the Alt key while you select the text with the left mouse button.



```
/* Create a report listing for the top values in
footnote2 "&stat of &measure for ALL values of &r
proc report data=topn;
    column &category rank &report &measure;
    define &category /group;
    define rank /display;
    define &measure / analysis &measureformat;
run;
quit;


/* Create a simple bar graph for the data to show
/* and relative values */
/* Calculate size of chart based on number of cat
goptions ypixels=%eval(250 * &categorycount) xpix
proc gchart data=topn
```

Collapsing and Expanding SAS Steps

With the Program Editor, you can collapse and expand each SAS step that a file contains. When the steps are collapsed, only the first line of each step is displayed.

To collapse steps:

- To collapse a step, click the minus sign (–) next to the first line of the step.

To expand steps:

- To expand a collapsed step, click the plus sign (+) next to the first line of the step.

Converting the Case of Your Text

You can convert the text in your program files to all uppercase or all lowercase.

To convert the case:

1. Use the left mouse button to select the text that you want to convert.
2. To change the selected text to uppercase, press Ctrl+Shift+U.

   To change the selected text to lowercase, press Ctrl+Shift+L.

Copying and Moving Text

You can drag selected sections of text between two files, or you can use the standard copy, cut, and paste commands. You can copy or move sections of text within a single file, among different open files in SAS Enterprise Guide, or to external editors such as Microsoft WordPad or Microsoft Word.

To copy or move text:

1. If you are dragging text between two files, position the open files so that you can see both the source file and the destination file. Both of these files must be visible before you can drag text between them.

   Note: You can change the layout of the SAS Enterprise Guide work area to view two files at once by selecting **View** ⇨ **Apply tab layout** ⇨ **layout option**.

2. Use the mouse to select the section of text that you want to copy or move.

3 To copy the text, press the Ctrl key while you drag the selected section to the new location.

To move the text, use the left mouse button to drag the selected section to the new location.

Using Editor Snippets

You can use editor snippets to store lines of code or text that you can easily insert into a program by typing the first few characters of the abbreviation name.

To create a snippet:

1 Click **Program** ⇨ **New snippet**.

2 In the **Abbreviation** box, enter the abbreviated name that you want to associate with the snippet (for example, `univ`).

3 In the **Text to insert** box, enter the text that you want the snippet to represent (for example, `proc univariate data=`). You can enter as much text as you want.

4 If you want to use the features of the Program Editor as you type the snippet text, select **SAS code editing mode**.

5 Click **OK** to save the snippet.

To insert a snippet:

1 At any location in the Program Editor, begin typing the abbreviated name of the snippet. A pop-up window opens with a list of suggested options that start with those letters.

2 Select the name of the snippet that you want to use, and press Enter to replace the snippet with the full text.

Viewing Keyboard Shortcuts in the Program Editor

By default, the Program Editor has many keyboard shortcuts that can make it easier for you to perform the most common commands when you are editing a program. You can view a complete list of the commands and their descriptions along with their corresponding keyboard shortcuts.

Note: You can also assign your own keyboard shortcuts to commands in the Program Editor. For more information, see “Assigning Keyboard Shortcuts in the Program Editor” on page 344.

To view the keyboard shortcuts:

1 Select **Program** ⇨ **Enhanced editor keys**. The Enhanced Editor Keys window displays a list of commands with keyboard shortcuts.

2 You can change the commands that are displayed in these ways:

   - To change the category of commands that are displayed, select another category from the **Category** drop-down list. By default, commands from all categories are displayed.
   - To view all of the commands for the selected category, including commands that are not currently assigned to keyboard shortcuts, select **Include commands without key assignments**.

Note: To reset all keyboard shortcuts to their default assignments, click **Reset All**. When you click **Reset All**, any keyboard shortcuts that you have changed or created are removed.

Working with SAS Macro Variables

Global SAS Macro Variables in SAS Enterprise Guide

Macro variables can be used to add information that is obtained when the task is run, such as the current time and date. 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 code:

```
Generated by &_CLIENTUSERNAME (&_CLIENTUSERID) with SAS Enterprise Guide &_CLIENTVERSION
```

Note: In addition to the macro variables listed below, you can use SAS macro functions and variables. For information about SAS macro functions and variables, see *SAS Macro Language: Reference*.

Table 9.1 *SAS Macro Variables in SAS Enterprise Guide*

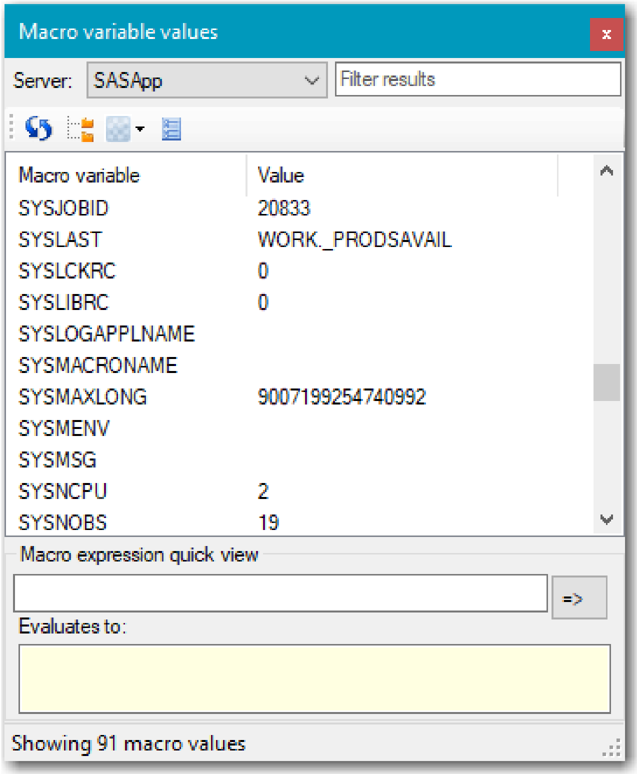
Macro Variable	Description
SYSSTREAMINGLOG	Streams content to the log while a procedure or DATA step is executing rather than displaying the log after the program completes.
_CLIENTAPP	Name of the client application

Macro Variable	Description
_CLIENTMACHINE	Client machine node name
_CLIENTPROCESSFLOWNAME	Name of the process flow
_CLIENTPROJECTNAME	File name for the project
_CLIENTPROJECTPATH	Full path and file name for the project
_CLIENTTASKFILTER	Filter that is defined for the task. You can use this macro variable in the titles and footnotes of the task so that the filter information is displayed in the title or footnote of your results.
_CLIENTTASKLABEL	Label for the current task. This is the text label that is displayed in the Project pane and the process flow.
_CLIENTUSERID	User ID of the client user
_CLIENTUSERNAME	Full user name, if that information is available
_CLIENTVERSION	Application version, including build number
_SASHOSTNAME	Server node name (IP address, DNS name, or DCOM name)
_SASPROGRAMFILE	Full path and file name of the SAS program that is currently being run. **Note:** In versions of SAS Enterprise Guide prior to SAS Enterprise Guide 7.13 HF2 and SAS Enterprise Guide 7.15, this macro variable is available only for SAS program files that are saved on the same server on which your SAS Enterprise Guide code is being run.
_SASSERVERNAME	Name of the logical server

Note: Previous versions of SAS Enterprise Guide used macro variables with names that began with _EG. These macro variables are no longer generated. Instead, use the macro variables in the previous table.

Viewing the SAS Macro Variables in Your Current SAS Session

To view all of the SAS macro variables that are defined in your current SAS session, select **Tools** ⇨ **SAS macro variable viewer**.



Server: SASApp

Macro variable	Value
SYSJOBID	20833
SYSLAST	WORK._PRODSAVAIL
SYSLCKRC	0
SYSLIBRC	0
SYSLOGAPPLNAME	
SYSMACRONAME	
SYSMAXLONG	9007199254740992
SYSMENV	
SYSMSG	
SYSNCPU	2
SYSNOBS	19

Showing 91 macro values

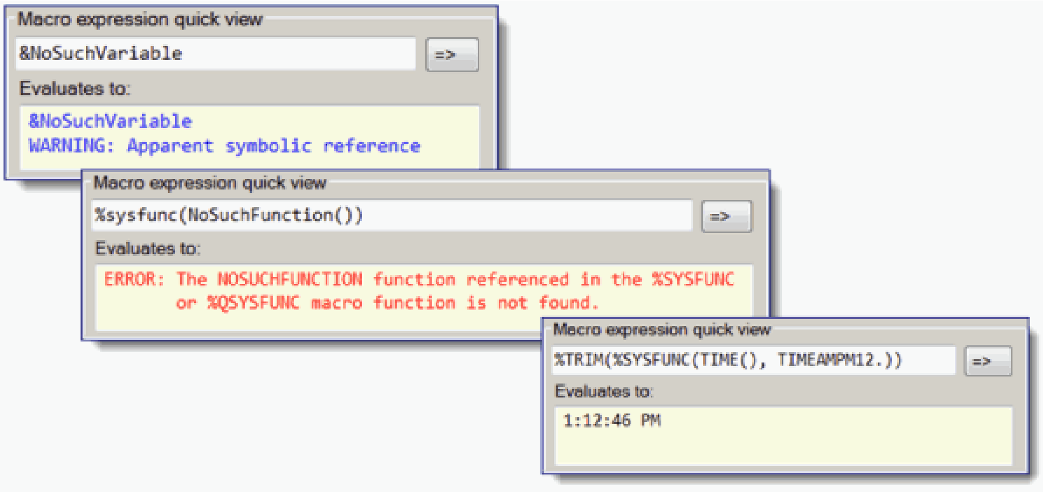
By default, the macro variables are displayed by groups. The two groups are Automatic and Global.

You can filter the results to subset the list of macro variables to only those variables of interest.

This window can remain open during your entire SAS Enterprise Guide session. You can easily switch between other SAS Enterprise Guide windows and the SAS Macro Variable Viewer to see any changes.

If your SAS environment contains connections to multiple SAS Workspace Servers, you can use the **Server** drop-down list to see the values for the macro variables on multiple machines.

You can also use the SAS Macro Variable Viewer to quickly evaluate an expression. This functionality enables you to get immediate feedback on a macro expression without adding it to a longer SAS program. The expression can be any macro expression that is valid for the right side of a macro variable assignment. If the expression is problematic, a SAS warning or error appears in the feedback window.



```
&NoSuchVariable
```

```
&NoSuchVariable
WARNING: Apparent symbolic reference
```

```
%sysfunc(NoSuchFunction())
```

```
ERROR: The NOSUCHFUNCTION function referenced in the %SYSFUNC
       or %QSYSFUNC macro function is not found.
```

```
%TRIM(%SYSFUNC(TIME(), TIMEAMPM12.))
```

```
1:12:46 PM
```

Customizing Your Programs

Using Prompts in a Program

Prompts in a program file require the user to select or enter a value when the program runs. For example, suppose that you have a program that lists sales by Product Name and Quarter. You could add a numeric prompt to the program so that the user can specify the value for the Quarter variable when the program runs.

```
PROC PRINT DATA=SASHELP.PRDSALE;
   VAR Product Quarter Actual Predict;
   Where Quarter EQ &Quarter_prompt;
RUN;
```

For a program, you can create a prompt in two ways:

- To create a prompt before you create the program, use the Prompt Manager. For more information, see “About Creating Prompts” on page 385.
- To create a prompt after you create the program, edit the program properties.

After you create a prompt, you must associate the prompt with the program. If you do not associate the prompt with the program, the user is not prompted for a value when the program runs.

To create a prompt by editing the program properties:

1. Open the program item in which you want to use a prompt and click ☰ on the work area toolbar.
2. Select **Prompts** in the selection pane, and then click **Prompt Manager**.

3. To create a prompt, click **Add**. For more information, see “About Creating Prompts” on page 385.

To associate an existing prompt with a program:

1. Open the program item in which you want to use a prompt and click ☰ on the work area toolbar.

2. Click **Prompts** in the selection pane, and then click **Add**.

3. Select the prompt that you want to associate with the program item and click **OK**.

4. Click **OK** to close the Properties window.

5. Edit the program file in which you want to use a prompt. The prompt should be added to the code as ***&prompt-name***.

Adding Status Updates to a SAS Program

About the SYSECHO Statement

You can use the SYSECHO statement to display text that provides the status of a running program. You can view the status text in the Submission Status pane. For more information, see “Using the Submission Status Pane” on page 328.

The status text is displayed until the next status event, which could be another SYSECHO statement, the start of a DATA step, or the start of a PROC step. Therefore, in order to be effective, you must position the SYSECHO statements within DATA steps or PROC steps or prior to other long-running statements.

Here is the syntax for the SYSECHO statement:

```
SYSECHO <"text">
```

text is a text string that must be between 1 and 64 characters long. This text must be enclosed in double quotation marks.

For more information about the SYSECHO statement, see *SAS Global Statements: Reference*.

Example 1: How to Place the SYSECHO Statement in a Program

In the following example, the SYSECHO status message, "My update," is never visible because it is immediately replaced by the "Running DATA step" status message that is generated when SAS Enterprise Guide runs a DATA step.

```
SYSECHO "My update";
```

```
DATA _NULL_;
/* do work */
RUN;
```

However, in this example, the SYSECHO status message is displayed until the next SYSECHO statement or DATA or PROC step boundary.

```
DATA _NULL_;
SYSECHO "My update";
/* do work */
RUN;
```

Example 2: How to Use the SYSECHO Statement in a Loop

In the following example, the status message from the SYSECHO statement provides information about what step is being processed in an iterative loop.

```
/* tracking in a loop */
%macro loop;
%do i=1 %to 10;
data _null_;
sysecho "In my data step - loop &i";
x = sleep(1);
run;
%end;
%mend;

%loop;
```

Example 3: How to Use the SYSECHO Statement in a Long Program

In the following example, the status message from the SYSECHO statement provides information about what step is being processed in a long program.

```
/* tracking in a long-running proc */
ods graphics on;
proc reg data=sashelp.class;
sysecho "in proc reg, model age=height";
model age=height;
run;
proc reg data=sashelp.class;
sysecho "in proc reg, model age=weight";
model age=weight;
run;
quit;
```

```
ods graphics off;
```

Using Editor Macros

About Editor Macros

An editor macro is a series of Program Editor commands and instructions that you group together as a single command to accomplish a task automatically. Instead of manually performing a series of time-consuming, repetitive actions, you can create and run a single macro. You run a macro from the Program menu or by using a keyboard shortcut.

You can share macro files between SAS users by using the export and import functionality.

Recording Macros

You can create a macro by recording it from the Program Editor or by adding commands in the Create Keyboard Macro window.

To record from the Program Editor:

1 Start recording either by pressing Alt+Shift+R or by selecting **Program** ⇨ **Editor macros** ⇨ **Record new**.

2 Execute the sequence of actions to accomplish the task. You can record any sequence of keystrokes, including the Delete and Backspace keys, function keys, cursor control keys, and shortcut keys for other editor macros.

3 Stop recording either by pressing Alt+Shift+R or by selecting **Program** ⇨ **Editor macros** ⇨ **Stop recording**.

4 In the Save Keyboard Macro window, enter a name and description for the editor macro.

5 If desired, click **Assign Keys** to define a keyboard shortcut to play the macro. For more information, see "Assigning Keyboard Shortcuts to Macros" on page 322.

6 Click **OK**.

To add commands by using the Create Keyboard Macro window:

1 Select **Program** ⇨ **Manage macros and snippets**. The Manage Macros and Snippets window opens.

2. Click **Create**. The Create Keyboard Macro window opens.
3. Enter the name and description of the macro.
4. To add a command, select the command from the **Commands** box and click ➡.
5. Specify the order in which the commands are executed. The commands are executed in the order in which they appear in the **Keyboard macro contents** box.
6. When all of the desired commands are present in the list and are ordered correctly, click **OK**.
7. In the Manage Macros and Snippets window, click **Close**.

Assigning Keyboard Shortcuts to Macros

You can assign a shortcut key to a macro immediately after you record it or later after you have saved it. You can also assign shortcut keys to the commands in the SAS language editor. For more information, see "Assigning Keyboard Shortcuts in the Program Editor" on page 344.

To assign a shortcut key to a macro after you record it:

1. After you record and name your macro, click **Assign Keys** in the Save Keyboard Macro window. The Assign Keys window appears with the new macro selected in the Commands box.
2. Position the cursor in the **Press new shortcut key** box, and press the key or combination of keys to use as a shortcut for the macro. You can use many keys and combinations of letter, numeric, function, and cursor control keys with the Shift, Ctrl, and Alt keys. Valid shortcut key sequences appear in the **Press new shortcut key** box and the assignment status for that key appears at the bottom of the window. If the value of **Currently assigned to** is `None`, no other command is assigned to this keyboard shortcut.
3. Click **Assign** to assign the keyboard shortcut.

   Note: If you assign a macro to a keyboard shortcut that is already assigned to another command, the previous command for that shortcut is replaced. For example, if you assign Ctrl+F to the `Add a new snippet` command, pressing Ctrl+F opens the Add Snippet window. You can no longer use Ctrl+F to open the Find window.

To assign a shortcut key to an existing macro:

1. Open the Manage Macros and Snippets window by selecting **Program** ⇨ **Manage macros and snippets**.
2. Select the macro to which you want to assign a shortcut key and click **Assign Keys**. The Assign Keys window opens.

3 Position the cursor in the **Press new shortcut key** box, and press the key or combination of keys to use as a shortcut for the macro. You can use many keys and combinations of letter, numeric, function, and cursor control keys with the Shift, Ctrl, and Alt keys. Valid shortcut key sequences appear in the **Press new shortcut key** box and the assignment status for that key appears at the bottom of the window. If the value of **Currently assigned to** is `None`, no other command is assigned to this keyboard shortcut.

4 Click **Assign** to assign the keyboard shortcut.

Note: If you assign a macro to a keyboard shortcut that is already assigned to another command, the previous command for that shortcut is replaced. For example, if you assign Ctrl+F to the `Add a new snippet` command, pressing Ctrl +F opens the Add Snippet window. You can no longer use Ctrl+F to open the Find window.

Running a Macro

1 Position the cursor in the Program Editor at the point where you want the macro to run.

2 Select **Program** ⇨ **Editor macros** ⇨ **Run macros**.

3 In the Run Keyboard Macro window, select a macro from the **Commands** list and click **Run**.

Running and Saving Programs

Analyzing a Program

About Analyzing a Program

You can use SAS Enterprise Guide to analyze the contents of a program. You might want to analyze a SAS program for these reasons:

- to quickly identify the different parts of the program and see how each part is related
- to take advantage of grid computing

- to determine whether there are any possible internationalization issues

Analyzing a Program to Create a Process Flow

Using SAS Enterprise Guide, you can create a process flow from a program. Using this process flow, you can quickly identify the different parts of the program and see how the parts are related.

Because SAS Enterprise Guide must run the program to perform the analysis, the amount of time that it takes to analyze the program is the same as the amount of time to run the program. During the analysis, the SCAPROC procedure identifies the parts (DATA steps and procedures) in the program. Then the results are displayed in SAS Enterprise Guide.

To analyze a program and create a process flow:

1. Open the program in the SAS Enterprise Guide work area.
2. Click ⋮ and select **Analyze program** ⇨ **Analyze program flow**. The Analyze SAS Program window opens.
3. To expand the macro invocations into separate tasks, click **Expand macros**.
4. Click **Begin analysis**. The results show each DATA step and procedure in the program.
5. To create a process flow from this program, specify a name for the process flow and click **Create process flow**. The new process flow appears in the main work area and in the Project pane.

Analyzing a Program for Grid Computing

When analyzing a program for grid computing, SAS Enterprise Guide identifies the parts of the program that are not dependent on one another. These parts can run simultaneously on multiple computers, which means that SAS Enterprise Guide returns the results faster. When analyzing a program, lines of SAS/CONNECT code are added to your original program. Therefore, you must have a license for SAS Grid Manager or SAS/CONNECT to analyze a program for grid computing.

Because SAS Enterprise Guide must run the program to perform the analysis, the amount of time that it takes to analyze the program is the same as the amount of time to run the program. During the analysis, the SCAPROC procedure identifies the parts (DATA steps and procedures) in the program. Then the results are displayed in SAS Enterprise Guide.

To analyze a program for grid computing:

1. Open the program in the SAS Enterprise Guide work area.
2. Click ⋮ and select **Analyze program** ⇨ **Analyze for grid computing**. The Analyze SAS Program for Grid Computing window opens.

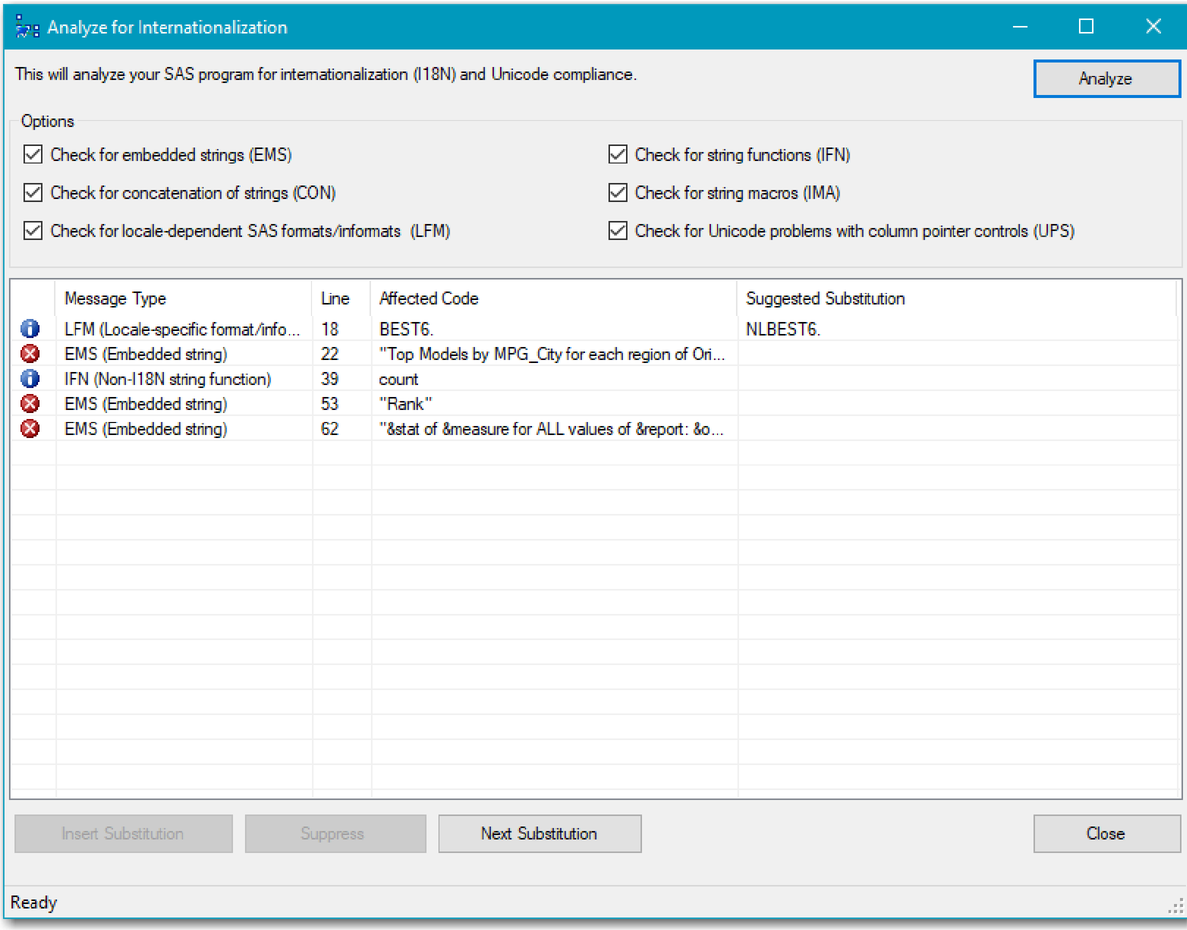
3 To expand the macro invocations into separate tasks, click **Expand macros**.

4 Click **Begin analysis**. The resulting program includes SAS/CONNECT code that is required for grid computing.

5 To save this new program to your project, specify a name for the program in the **Add grid-enabled program as** box. The new program appears in the Project pane and in the process flow for the project. Your original program is also still part of your project.

Analyzing a Program for Internationalization

You can analyze a SAS program to determine whether there are any possible internationalization issues. Internationalization is the process by which a program is optimized so that it can be adapted to any language and region without being rewritten. When you analyze a program for internationalization, SAS Enterprise Guide lists the lines of code that might be affected and suggests substitutions when possible.



This will analyze your SAS program for internationalization (I18N) and Unicode compliance.

Options

☑ Check for embedded strings (EMS)

☑ Check for concatenation of strings (CON)

☑ Check for locale-dependent SAS formats/informats (LFM)

☑ Check for string functions (IFN)

☑ Check for string macros (IMA)

☑ Check for Unicode problems with column pointer controls (UPS)

	Message Type	Line	Affected Code	Suggested Substitution
ⓘ	LFM (Locale-specific format/info...	18	BEST6.	NLBEST6.
⊗	EMS (Embedded string)	22	"Top Models by MPG_City for each region of Ori...	
ⓘ	IFN (Non-I18N string function)	39	count	
⊗	EMS (Embedded string)	53	"Rank"	
⊗	EMS (Embedded string)	62	"&stat of &measure for ALL values of &report: &o...	

Ready

When you analyze a program for internationalization, SAS Enterprise Guide checks for these items:

- **Embedded strings (EMS)** - text strings that are embedded in a program allow the application to display only one language to the user. Any strings that are visible in the user interface should be externalized, or stored in a separate properties file, so they can be translated. For information about replacing strings in your SAS program, see the SASMSG function in *SAS National Language Support (NLS): Reference Guide*.
- **Concatenated text strings (CON)** - concatenation is the operation in which two or more text strings are combined into one string. However, because grammatical structures of language can be very different, concatenation of strings can cause grammatical conflicts. It is best to avoid concatenation of strings that you intend to translate.
- **Locale-dependent SAS formats/informats (LFM)** - formats and informats can present locale dependencies for types of data that include dates, time, currency, and numerical values. You can use SAS NL formats to avoid such locale-specific dependencies.
- **String functions (IFN)** - SAS provides many string functions and CALL routines that can be used to manipulate characters and strings. Some string functions in SAS assume that the size of a single character is always 1 byte. However, in some environments, a single character can be up to 4 bytes. SAS provides additional K string functions that do not make assumptions about the size of a single character. Regular functions are byte-oriented, but K functions are character-oriented. For more information about K functions, see *SAS National Language Support (NLS): Reference Guide*.
- **String macros (IMA)** - SAS provides many macros that can be used to manipulate characters and strings. Some string macros in SAS assume that the size of a single character is always 1 byte. However, in some environments, a single character can be up to 4 bytes. SAS provides additional NLS macro functions that do not make assumptions about the size of a single character.
- **Column pointer controls in the PUT and INPUT statements in a UTF-8 SAS session (UPS)** - the SAS column pointer controls assume that each character can be displayed in one column. However, a character in a multi-byte encoding, such as UTF-8, is not equivalent to 1 byte as characters are in single-byte encodings. In UTF-8 encoding, all non-ASCII characters have a varying length between 2 and 4 bytes, which can cause the SAS column pointer controls to return unexpected results.

When you analyze your code, you might find that some of the reported issues occur in code that does not need to be changed. You can prevent SAS Enterprise Guide from including those issues by adding a comment to a single line or around a block of code. When you add the comment to your code, you can specify a particular issue that you want to exclude from the analysis, or you can exclude all issues.

This example shows how you can exclude embedded strings (EMS) from the analysis of one line of code.

```
data a;
   x="text string"; /* ignore for i18n analyzer - ID:EMS */
run;
```

This example shows how to exclude an entire block of code from all analysis.

```
/*  ignore for i18n analyzer - ID:ALL:begin */
```

```
data a;
  x='string 1';
  y='string 2';
  z=today();
  put z date.;
run;
/*  ignore for i18n analyzer - ID:ALL:end */
```

To analyze a program for internationalization:

1 Open the program in the SAS Enterprise Guide work area.

2 Click ⋮ and select **Analyze program** ⇨ **Analyze for internationalization**. The Analyze for Internationalization window opens.

3 Select the check boxes for the types of issues that you want to include in the analysis. By default, all of the check boxes are selected.

4 Click **Analyze**. The results show each affected line of code in the program. You can click a line in the Analyze for Internationalization window to highlight the affected line in the Program Editor.

5 When you view the results of the analysis, you can select these options:

   - **Insert Substitution** replaces the affected code in your program with the suggested substitution. This option is available only if SAS Enterprise Guide has a suggested substitution.
   - **Suppress** inserts a comment on the affected line of code so that it is excluded from analysis.
   - **Next Substitution** advances to the next substitution.

Selecting a Server for Running a Program

You can run a program file on the currently selected server, or you can select another server to run it on. Programs run on the default server unless you select another server.

To select another server:

- From the toolbar, select the server that you want to use from the server drop-down list. This server designation applies only to the active program.

Note: To change to a different SAS server environment, click 🔌.

Running a Program

You can run new or existing program files. You can run a program file on the currently selected server, or you can select another server to run it on. You can also select a subset of the code in a file, and run only the selection.

Note: The maximum line length for each line in a program is 6000 bytes.

Note: You can specify the maximum number of output data sets that are added to a project from a single program file. By default, only 50 data sets can be added to a project from a single program file. If your program creates more than the specified maximum number of data sets, a note that warns you of this limit is automatically added to your project. You can access the additional data sets on the SAS server on which the code is run, but they are not added to your project. For more information, see "Setting Project and Process Flow Options" on page 484.

After your program runs, a log is generated. The log lists any messages from SAS. If you encounter errors, see the log for messages.

To run a program:

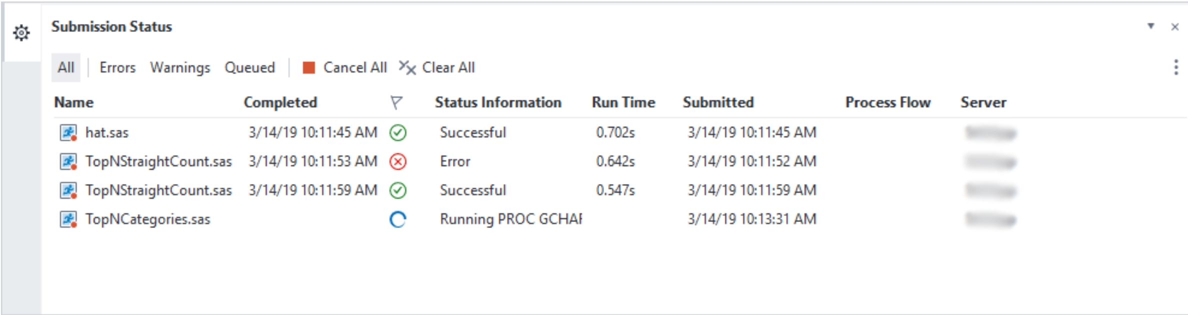
1 On the Program Editor toolbar, click **Run**.

   Note: To run a subset of the code, select the lines of code that you want to run before you click **Run**.

Using the Submission Status Pane

After you run a program in the Program Editor, you can view a summary of the messages that were generated in the log. The submission status lists all the errors, warnings, and notes that were generated when the program ran, as well as related line numbers and a sample of the affected code. From the Submission Status pane, you can click the message description and view the message in the program or log source file. You can also search the online Help for the message text.

By default, the Submission Status pane is not displayed. To open the Submission Status pane, select **View** ⇨ **Submission Status**.



Name	Completed		Status Information	Run Time	Submitted	Process Flow	Server
hat.sas	3/14/19 10:11:45 AM	✓	Successful	0.702s	3/14/19 10:11:45 AM		[redacted]
TopNStraightCount.sas	3/14/19 10:11:53 AM	⊗	Error	0.642s	3/14/19 10:11:52 AM		[redacted]
TopNStraightCount.sas	3/14/19 10:11:59 AM	✓	Successful	0.547s	3/14/19 10:11:59 AM		[redacted]
TopNCategories.sas		↻	Running PROC GCHAI		3/14/19 10:13:31 AM		[redacted]

Saving Programs

By default, program files are saved as SAS Source Files (*.sas).

When you save a new or existing program and choose the name and location, SAS Enterprise Guide automatically saves the program using its default character encoding when possible. You can also specify an encoding to use when you save a program on your local computer.

Note: When you save a program file to a specified location, SAS Enterprise Guide automatically updates the reference to the program file in the project, if the program is included in a project.

To save a program and specify the encoding:

1. Select **File** ⇨ **Save "*program-name*" as**.
2. In the Save window, select **My Computer**.
3. From the **Save as** drop-down list, select **Select encoding**.
4. Click **Browse** to select the folder in which to save the program file, and then click **Save**. The Select Encoding window opens.
5. Select the encoding that you want to use and click **Select**. The program file is saved with the encoding that you specified.

Using the DATA Step Debugger

About the DATA Step Debugger

The DATA Step Debugger is a tool that enables you to find logic errors in a DATA step program. With the DATA Step Debugger, you can watch the variable values in a program change as the program runs. You can execute the program line by line, and you can also set specific breakpoints in the program.

Note: The DATA Step Debugger works only with DATA step programs.

Getting Started with the DATA Step Debugger

What Is Debugging?

Debugging is the process of removing logic errors from a program. Unlike syntax errors, logic errors do not stop a program from running. Instead, logic errors cause the program to produce unexpected results. For example, if you create a DATA step that tracks inventory, and your program shows that you are out of stock when your warehouse is actually full, you have a logic error in your program.

Without the DATA Step Debugger, you could debug your program by adding code, such as PUT statements, to your program and examining the results and log. Although the SAS log helps you identify data errors, the DATA Step Debugger offers you an easier, interactive way to identify logic errors, and sometimes data errors, in DATA steps.

How Debugging Works

When you open a DATA step in the DATA Step Debugger, SAS compiles the step, displays the code in the DATA Step Debugger window, and pauses until you begin execution. Each time execution is paused, the current values of each variable are displayed in the variables pane. For more information, see “Inspecting Variable Values” on page 335.

You can execute your program in the DATA Step Debugger in several ways:

- If you begin execution by clicking 🏃, SAS executes each statement in the DATA step.
- To suspend execution at a particular line in the DATA step program, select one or more statements on which to set breakpoints. When you click 🏃, SAS executes the program until a breakpoint is reached. For more information, see “Setting and Clearing Breakpoints” on page 333.
- To execute the DATA step one statement at a time, click ⤓. You can examine the variable values after each step in the program. For more information, see “Executing the Program Step by Step” on page 334.
- To execute the DATA step until the value of a specific variable changes, select the variable to watch and click 🏃. SAS executes the program until the value of the specified variable changes. For more information, see “Watching a Variable Value” on page 335.

Note: When you use the DATA Step Debugger, note these reminders:

- When you are debugging a DATA step, statements in the DATA step can iterate as many times as they would outside the debugging session. When the last iteration has finished, a message appears in the Debug Console.
- DATA steps that include the DATALINES and CARDS statements cannot be debugged. You can work around this limitation in these ways:
  - Store your data in a flat file, and use the FILENAME statement to read the data into the DATA step.
  - Create two separate DATA steps: one that includes the DATALINES or CARDS statement and another debuggable DATA step to process the data.
- You can debug only one DATA step at a time. You can use the DATA Step Debugger only with a DATA step and not with a PROC step.

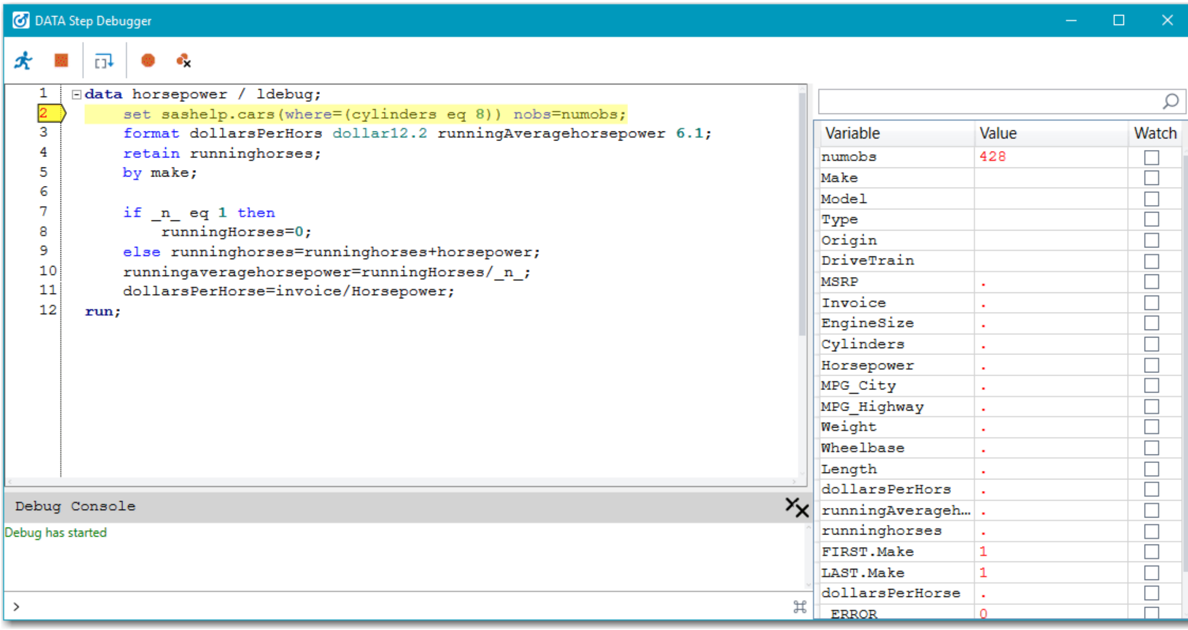
Using the DATA Step Debugger

About the DATA Step Debugger Window

The DATA Step Debugger window includes many features that make it easier to debug your DATA step program:

- The currently executing line is highlighted in yellow.
- The variables pane on the right side of the window lists each variable in the program with its current value.
- You can use the filter and search box to narrow the list of variables in the variables pane.
- As you step through a program, variable values that change are displayed in red.

- The Debug Console at the bottom of the window displays logging information as you execute the program.
- You can enter debugger commands directly on the command line at the bottom of the window. To view a list of valid commands, click ⌘.



```
data horsepower / ldebug;
    set sashelp.cars(where=(cylinders eq 8)) nobs=numobs;
    format dollarsPerHors dollar12.2 runningAveragehorsepower 6.1;
    retain runninghorses;
    by make;

    if _n_ eq 1 then
        runningHorses=0;
    else runninghorses=runninghorses+horsepower;
    runningaveragehorsepower=runningHorses/_n_;
    dollarsPerHorse=invoice/Horsepower;
run;
```

Debug Console

Debug has started

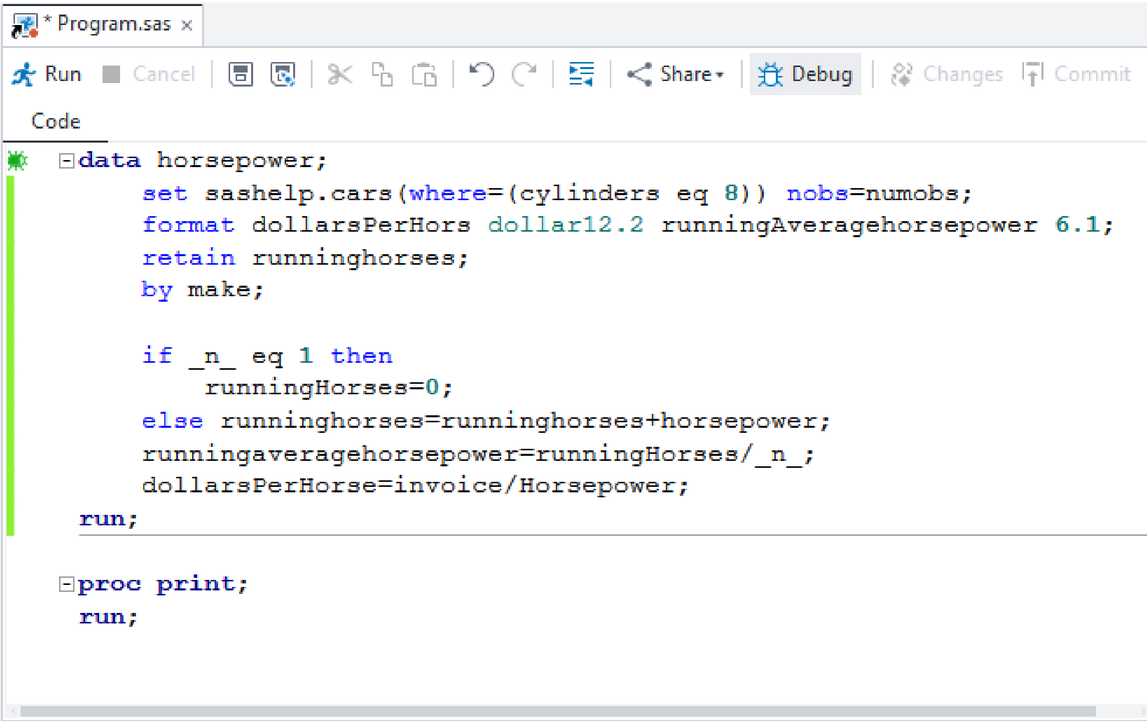
Variable	Value	Watch
numobs	428	□
Make		□
Model		□
Type		□
Origin		□
DriveTrain		□
MSRP	.	□
Invoice	.	□
EngineSize	.	□
Cylinders	.	□
Horsepower	.	□
MPG_City	.	□
MPG_Highway	.	□
Weight	.	□
Wheelbase	.	□
Length	.	□
dollarsPerHors	.	□
runningAverageh...	.	□
runninghorses	.	□
FIRST.Make	1	□
LAST.Make	1	□
dollarsPerHorse	.	□
ERROR	0	□

Opening the DATA Step Debugger

You can run the DATA Step Debugger on any program that contains a DATA step. To get started using the DATA Step Debugger, you must first enable the DATA Step Debugger and select the valid DATA step code that you want to debug.

To run the DATA Step Debugger:

1 Open your program in the Program Editor and click **Debug** on the toolbar. All sections of DATA step code in the program are highlighted with a green bar in the margin to indicate that they can be debugged.

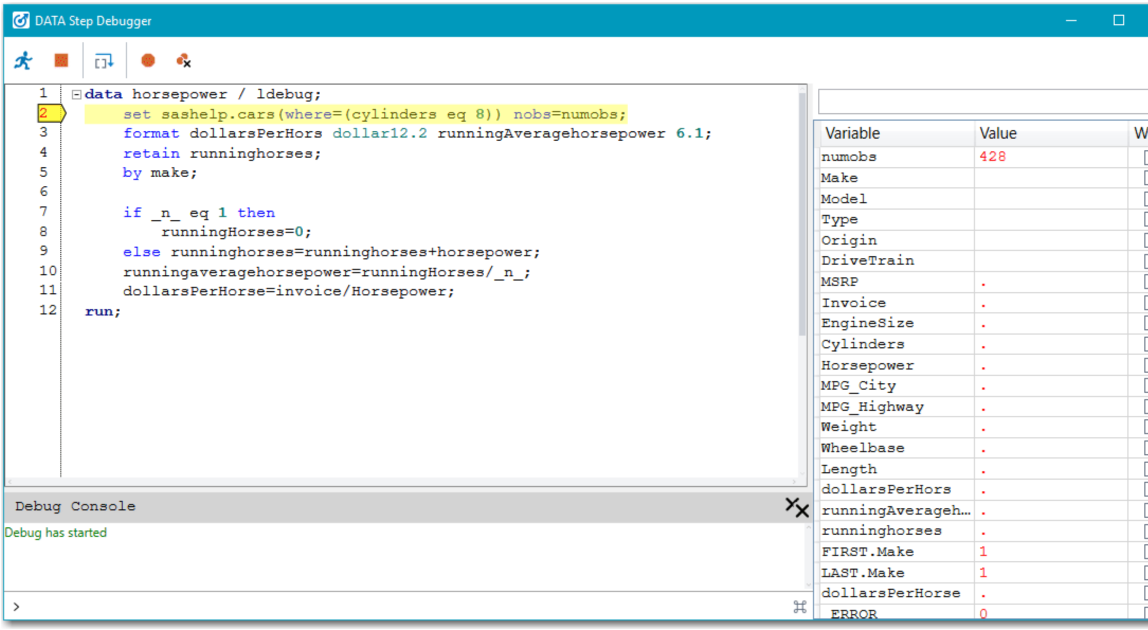


```
data horsepower;
      set sashelp.cars(where=(cylinders eq 8)) nobs=numobs;
      format dollarsPerHors dollar12.2 runningAveragehorsepower 6.1;
      retain runninghorses;
      by make;

      if _n_ eq 1 then
          runningHorses=0;
      else runninghorses=runninghorses+horsepower;
      runningaveragehorsepower=runningHorses/_n_;
      dollarsPerHorse=invoice/Horsepower;
run;

proc print;
run;
```

2 Click in the margin beside the section of code that you want to debug. The DATA step code opens in the DATA Step Debugger window.



```
data horsepower / ldebug;
    set sashelp.cars(where=(cylinders eq 8)) nobs=numobs;
    format dollarsPerHors dollar12.2 runningAveragehorsepower 6.1;
    retain runninghorses;
    by make;

    if _n_ eq 1 then
        runningHorses=0;
    else runninghorses=runninghorses+horsepower;
    runningaveragehorsepower=runningHorses/_n_;
    dollarsPerHorse=invoice/Horsepower;
run;
```

Debug has started

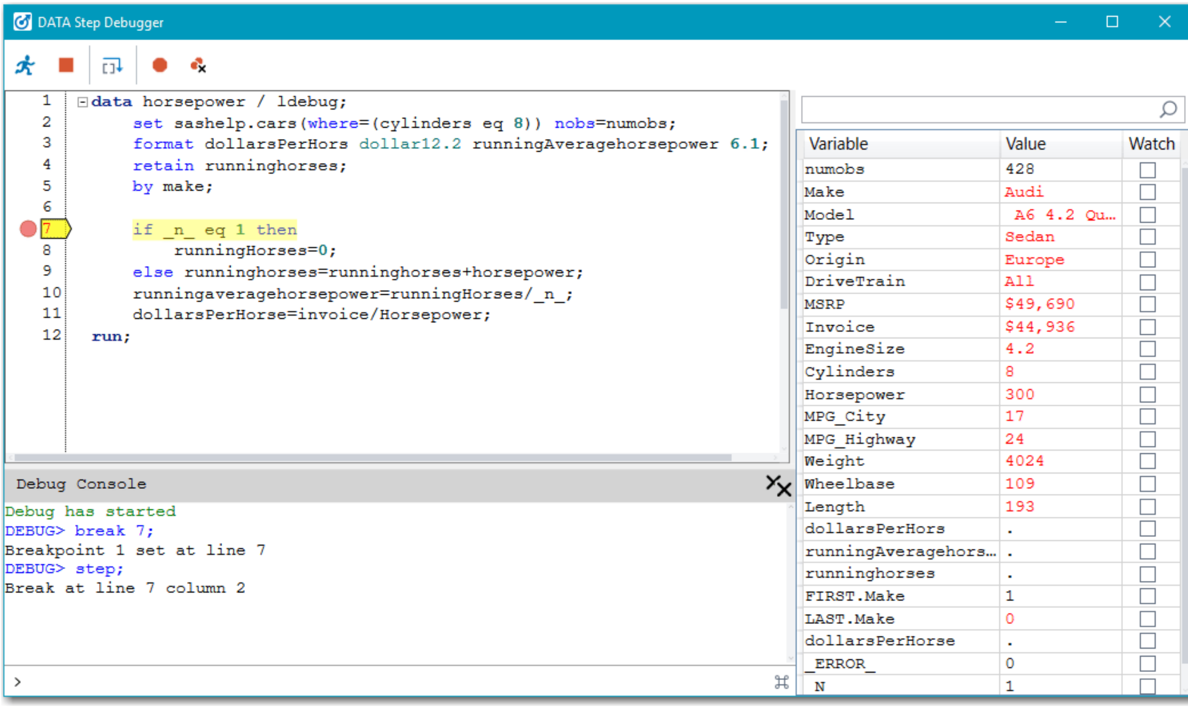
Variable	Value	W
numobs	428	
Make		
Model		
Type		
Origin		
DriveTrain		
MSRP	.	
Invoice	.	
EngineSize	.	
Cylinders	.	
Horsepower	.	
MPG_City	.	
MPG_Highway	.	
Weight	.	
Wheelbase	.	
Length	.	
dollarsPerHors	.	
runningAverageh…	.	
runninghorses	.	
FIRST.Make	1	
LAST.Make	1	
dollarsPerHorse	.	
ERROR	0	

Setting and Clearing Breakpoints

You can set or clear a breakpoint on any executable line in the program. A breakpoint suspends execution of the program at the specified line. Each time you set or clear a breakpoint, information about the breakpoint is displayed in the Debug Console at the bottom of the window.

To set and clear breakpoints:

- You can toggle a breakpoint on and off by clicking the line number on which you want to set a breakpoint. You can also select the line and click ● on the toolbar.
- To clear all breakpoints in the program, click ●✕.



```
 1  data horsepower / ldebug;
 2      set sashelp.cars(where=(cylinders eq 8)) nobs=numobs;
 3      format dollarsPerHors dollar12.2 runningAveragehorsepower 6.1;
 4      retain runninghorses;
 5      by make;
 6
 7      if _n_ eq 1 then
 8          runningHorses=0;
 9      else runninghorses=runninghorses+horsepower;
10      runningaveragehorsepower=runningHorses/_n_;
11      dollarsPerHorse=invoice/Horsepower;
12  run;
```

```
Debug has started
DEBUG> break 7;
Breakpoint 1 set at line 7
DEBUG> step;
Break at line 7 column 2
```

Variable	Value	Watch
numobs	428	□
Make	Audi	□
Model	A6 4.2 Qu…	□
Type	Sedan	□
Origin	Europe	□
DriveTrain	All	□
MSRP	$49,690	□
Invoice	$44,936	□
EngineSize	4.2	□
Cylinders	8	□
Horsepower	300	□
MPG_City	17	□
MPG_Highway	24	□
Weight	4024	□
Wheelbase	109	□
Length	193	□
dollarsPerHors	.	□
runningAveragehors…	.	□
runninghorses	.	□
FIRST.Make	1	□
LAST.Make	0	□
dollarsPerHorse	.	□
ERROR	0	□
N	1	□

Executing the Program Step by Step

You can execute the statements in the program one at a time and view the variable values with each step. As you step through your program, any variable values that change are displayed in red.

To execute the statement starting at the point at which execution was suspended, click [ ]↓. Only one statement is executed at a time.

Starting and Stopping Execution of the Program

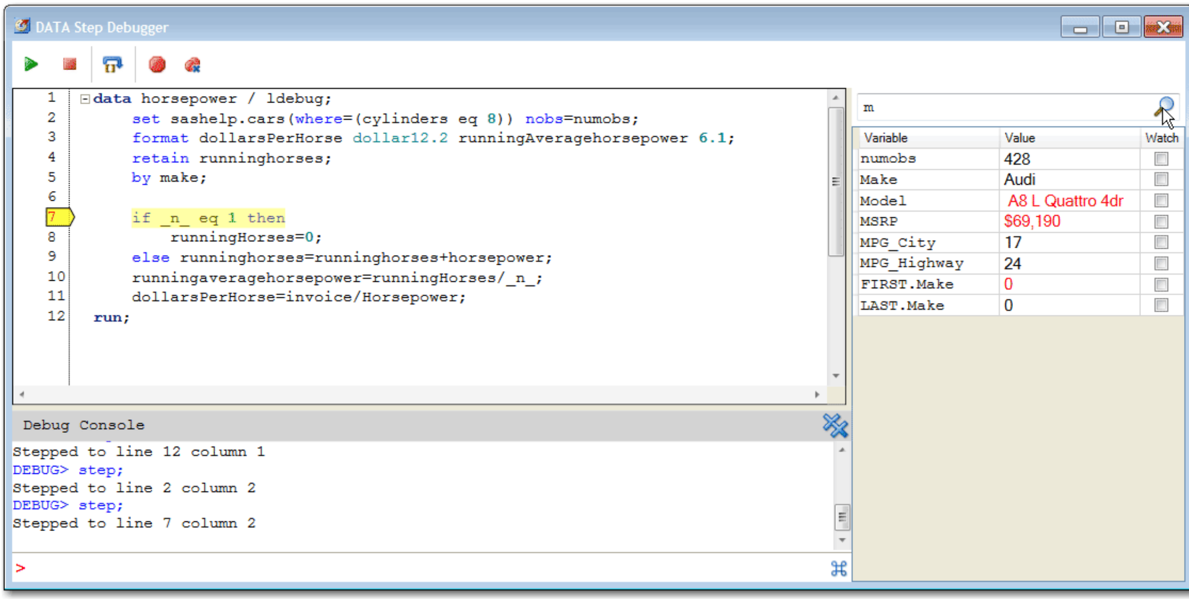
When you start execution of the program in the DATA Step Debugger, the program runs until it reaches a breakpoint or until the value of a watched variable changes. If no breakpoints or watched variables are specified, the program runs to completion.

- To start or continue execution of the program, click 🏃.
- To stop execution of the program, click ■.

Note: You can specify a line in the program at which to restart execution by right-clicking the appropriate line in the program and selecting **Jump execution to this line**.

Inspecting Variable Values

As you debug a program, you can view the values of each variable in the program. You can view the complete list of variables in the variables pane, and you can use the filter and search box to narrow the list of variables. Variable values that have changed are displayed in red.



```
data horsepower / ldebug;
     set sashelp.cars(where=(cylinders eq 8)) nobs=numobs;
     format dollarsPerHorse dollar12.2 runningAveragehorsepower 6.1;
     retain runninghorses;
     by make;

     if _n_ eq 1 then
         runningHorses=0;
     else runninghorses=runninghorses+horsepower;
     runningaveragehorsepower=runningHorses/_n_;
     dollarsPerHorse=invoice/Horsepower;
run;
```

```
Stepped to line 12 column 1
DEBUG> step;
Stepped to line 2 column 2
DEBUG> step;
Stepped to line 7 column 2
>
```

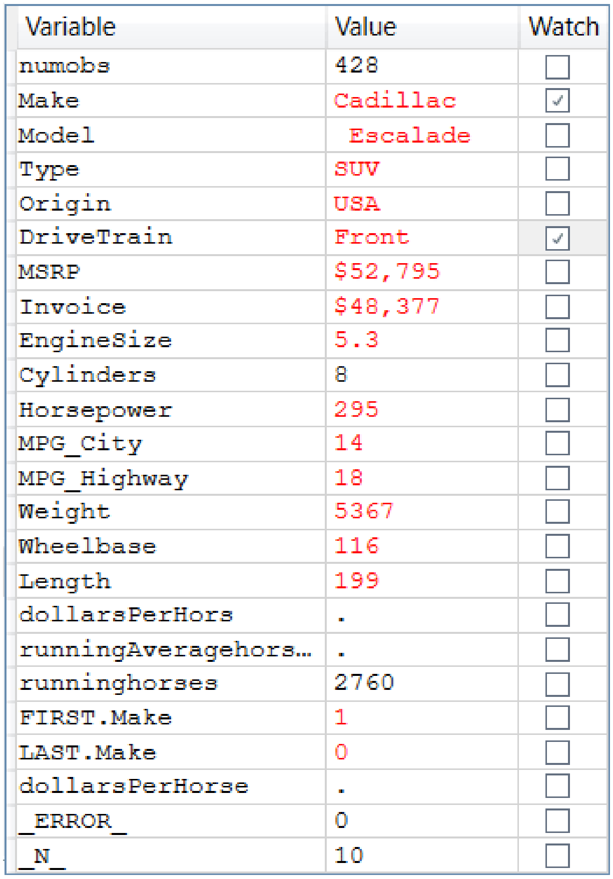
m

Variable	Value	Watch
numobs	428	☐
Make	Audi	☐
Model	A8 L Quattro 4dr	☐
MSRP	$69,190	☐
MPG_City	17	☐
MPG_Highway	24	☐
FIRST.Make	0	☐
LAST.Make	0	☐

Watching a Variable Value

You can watch any variable in the program. SAS suspends execution of the program whenever the value of a watched variable changes. Variable values that have changed are displayed in the variables pane in red.

To watch a variable, select the check box for the variable in the variables pane.



Variable	Value	Watch
numobs	428	□
Make	Cadillac	☑
Model	Escalade	□
Type	SUV	□
Origin	USA	□
DriveTrain	Front	☑
MSRP	$52,795	□
Invoice	$48,377	□
EngineSize	5.3	□
Cylinders	8	□
Horsepower	295	□
MPG_City	14	□
MPG_Highway	18	□
Weight	5367	□
Wheelbase	116	□
Length	199	□
dollarsPerHors	.	□
runningAveragehors…	.	□
runninghorses	2760	□
FIRST.Make	1	□
LAST.Make	0	□
dollarsPerHorse	.	□
ERROR	0	□
_N_	10	□

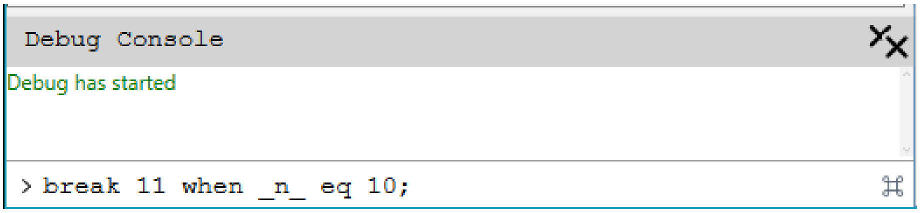
Debugging with the Command Line

You can use the command line to enter debugger commands directly. To view a list of valid commands, click ⌘.

Note: For information about DATA step debugger commands, see “Issuing Commands” in *SAS Code Debugger and SAS DATA Step Debugger: User’s Guide*.

For example, you can use the command line to set a conditional breakpoint so that execution of the program is stopped when a specified condition is met. The following example uses this debugger command to set a breakpoint at line 11 when the value of the variable `_n_` is `10`:

```
break 11 when _n_ eq 10
```

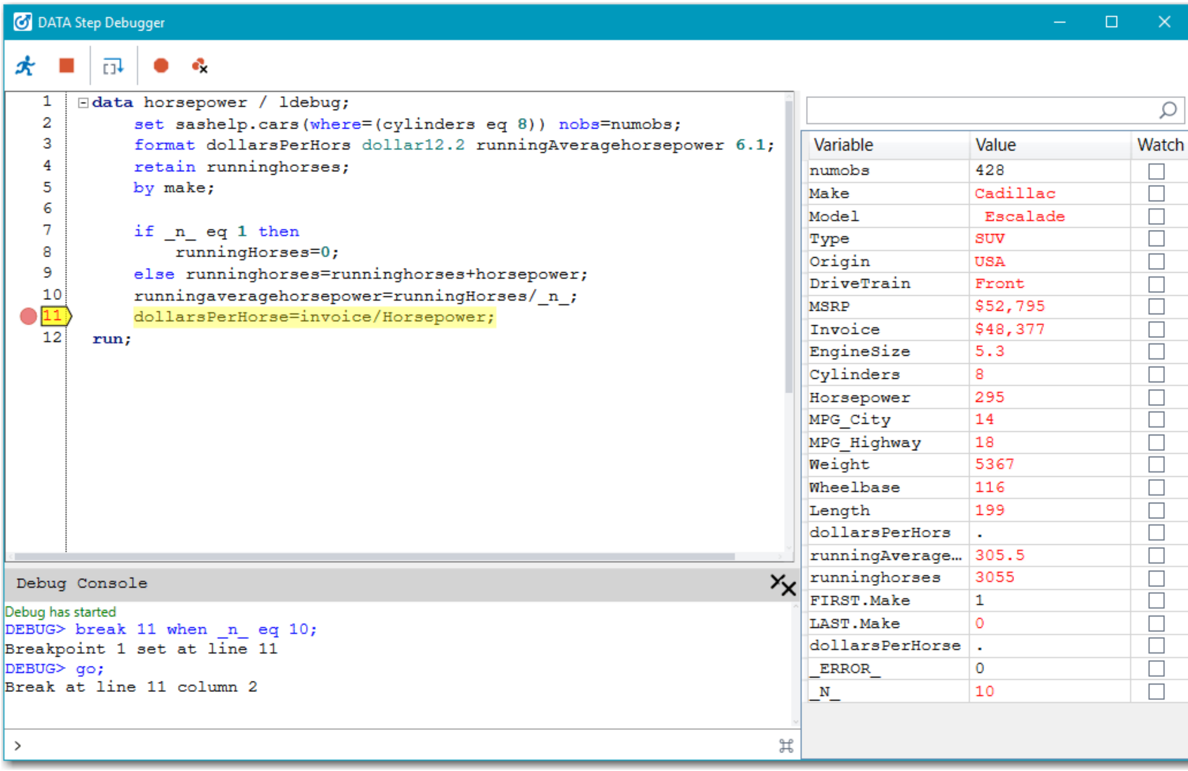


Debug Console

Debug has started

```
> break 11 when _n_ eq 10;
```

The program execution stops when the value of `_n_` is `10`.



```
1   data horsepower / ldebug;
2       set sashelp.cars(where=(cylinders eq 8)) nobs=numobs;
3       format dollarsPerHors dollar12.2 runningAveragehorsepower 6.1;
4       retain runninghorses;
5       by make;
6
7       if _n_ eq 1 then
8           runningHorses=0;
9       else runninghorses=runninghorses+horsepower;
10      runningaveragehorsepower=runningHorses/_n_;
11      dollarsPerHorse=invoice/Horsepower;
12  run;
```

```
Debug has started
DEBUG> break 11 when _n_ eq 10;
Breakpoint 1 set at line 11
DEBUG> go;
Break at line 11 column 2
```

Variable	Value	Watch
numobs	428	☐
Make	Cadillac	☐
Model	Escalade	☐
Type	SUV	☐
Origin	USA	☐
DriveTrain	Front	☐
MSRP	$52,795	☐
Invoice	$48,377	☐
EngineSize	5.3	☐
Cylinders	8	☐
Horsepower	295	☐
MPG_City	14	☐
MPG_Highway	18	☐
Weight	5367	☐
Wheelbase	116	☐
Length	199	☐
dollarsPerHors	.	☐
runningAverage...	305.5	☐
runninghorses	3055	☐
FIRST.Make	1	☐
LAST.Make	0	☐
dollarsPerHorse	.	☐
ERROR	0	☐
_N_	10	☐

Viewing Program and Log Properties

Viewing Program Properties

You can view and change many of the properties for each program in the Project pane. By default, program items use settings from the Options window, which is accessible from the Tools menu. You can choose to override those settings for individual program items.

You can view the properties for a program in these ways:

- Right-click the program item in the Project pane or in a process flow and select **Properties**.
- Open the program and click ☰ in the work area toolbar.

Table 9.2 General Properties

Property Name	Description
Label	Specifies the name of the program in the Project pane and in the process flow. You cannot change the name of an existing program that you have added to your project; you can change only the name of a program file that you have created in SAS Enterprise Guide.
Code will run on server	Specifies the name of the server on which SAS Enterprise Guide executes the program. You can change the server by selecting a new server from the drop-down list. The list of servers includes all of the servers that you can access.
Last Execution Time	Displays how long it took for the program to run the last time it was executed. This time is rounded to the nearest second and does not include the time it takes to connect to the server.
File path	Indicates whether changes that you make to the program are made to the original program file or whether a copy of the program is embedded in the project. If you are viewing the properties for an existing program file, the location lists the complete path name of the program file. If you are viewing the properties for a program file that you have created in SAS Enterprise Guide, the file path is listed as **(Embedded In Project)**. ■ **Embed** – Embeds the program in the SAS Enterprise Guide project so that any changes that you make to the program in SAS Enterprise Guide are not applied to the original program file. This option is available only for existing program files that you have inserted into your project. ■ **Save As** – Saves the program to the location that you specify.

Table 9.3 Results Properties

Property Name	Description
Use preferences from Tools -> Options	Specifies that SAS Enterprise Guide uses the settings that are specified in the Options window for this program. By default, the HTML format is used with the HTML style. The other results options are disabled if you select this option.
Customize result formats, styles and behavior	Enables you to override the results settings that are specified in the Options window for the active program. The results options are disabled until you select this option. **Note:** To generate results in Microsoft Excel or Microsoft PowerPoint, the SAS server must be running SAS 9.4M3 or later.

Property Name	Description
Graph Format	Determines the image format for graphs in results for this program. To change the format, select another format from the drop-down list. For more information, see "Setting Graph Results Options" on page 494.
Reset Options	Resets your results options back to their default values.

Table 9.4 *Code Submission*

Property Name	Description
Use preferences from project properties	Specifies that the code submission settings for the task should use the settings that are specified in the **Code Submission** section of the Properties for *project-name* window. For more information, see "Viewing Project Properties" on page 37.
Customize code submission options	Specifies that you want to customize the code submission options for this task. You can choose from the following options: - **Use grid if available** – Enables SAS Enterprise Guide to run tasks in a SAS grid computing environment. For an overview of SAS grid support in SAS Enterprise Guide, see "Using SAS Enterprise Guide with a SAS Grid" on page 12. - **Allow parallel execution on the same server** – Enables SAS Enterprise Guide to create one or more additional workspace server connections so that parallel process flow paths can be run in parallel on the same server. For more information, see "Using SAS Enterprise Guide with Parallel Execution" on page 13. **Note:** Because there might be more than one workspace server connection, any changes that you make in one workspace session, such as library assignments or changes to macro variables or options, might not be available to tasks that are running in another parallel workspace session.
Action to take on error in the execution of this program	Specifies how to handle an error in the execution of this program. - **Stop all items** – Stops the execution of all subsequent items in the project or process flow. - **Stop current branch** – Stops the execution of all subsequent items in the current branch of the process flow. This is the default value. - **Continue execution** – Continues the execution of all subsequent items in the project or process flow.

Table 9.5 Prompts Properties

Property Name	Description
Project prompts used	Displays the SAS name, display name, and data type of each prompt that you have associated with the program. You can add and remove prompts from the program and create prompts by using the Prompt Manager. For more information, see "Using Prompts in a Program" on page 318. ■ To associate existing project prompts with the program, click **Add** and select the prompt that you want to use. ■ To remove the association between a prompt and the program, select the prompt that you want to remove and click **Remove**. ■ To move the prompts up or down the list, select the prompt that you want to move and click the up or down arrow. The order of the prompts in the **Project prompts used** list determines the order in which the prompts are displayed to the user in the prompt window when the program is run. ■ To create a prompt, click **Prompt Manager**, and then click **Add**. For more information, see "About Creating Prompts" on page 385.

This is a summary of all the properties that are associated with the program. These properties are for informational purposes only; you are not able to edit any of these properties from this location.

Table 9.6 Summary Properties

Property Name	Description
Label	The name of the program.
Created	The date on which the program was created.
Last modified	The date on which the program was last modified.
Modified by	The user name of the person who last modified the program.

Viewing Log Properties

To view the summary properties for a log item, click the **Log** tab in the work area, and then click **Properties** on the toolbar. These properties are for informational purposes only; you cannot edit any of these properties from this location.

Sharing Programs with Other Users

Exporting Programs

About Exporting Programs

You can export a single program, or you can export all of the programs in a project into a single program file that you save to a specified location. You can also export programs as a step in the process flow of a project. When you export a program to a specified location, SAS Enterprise Guide does not change the reference to the program file in the project.

Exporting a Single Program

1 Select the program that you want to export in the work area or in the process flow.

2 From the toolbar, select **Share** ⇨ **Code** ⇨ **Export**.

3 In the Export window, select the appropriate export options for the file and click **Export**.

Export All Code

You can export all of the code in a project or process flow into a single program file that you save to a specified location. When you export a program to a specified location, SAS Enterprise Guide does not change the reference to the program file in the project.

Note: Exporting all of the code in a project is not the same as saving the project itself. You cannot use the exported code to re-create a project and its tasks. To save all of the tasks in a project, you need to save the project.

To export all code in a project or process flow:

1 From the Project tab of the navigation pane, click ⋮, and select **Share** ⇨ **Export all code in project**.

From the toolbar of the process flow, select **Share** ⇨ **Export all code in process flow**.

2 To change the location where the code file will be saved, enter the correct path in the **Export to file** box, or click **Browse** to find the correct location.

3 From the Encoding drop-down list, select the type of encoding that you want to use for the exported data. By default, the exported data is encoded in the same way that the original data was encoded.

4 In the **Code available for export** list, clear the check boxes next to any code items that you do not want to export.

5 In the **Code preview** area, you can click **Preview All Selected** to preview all of the selected code items, or you can click **Preview of *code-item*** to view only the selected code item.

6 You can select the following options:

- **Include headers** to include headings that provide descriptive information about the code.
- **Include generated wrapper code** to include wrapper code that SAS generates.
- **Include custom SAS code from the Tools --> Options dialog** to include SAS code that is specified in the Options window.
- **Include library assignment code** to include the code that SAS generates to assign the library.

7 Click **Export** to export the selected code items.

Send a Program

You can send a copy of your SAS Enterprise Guide programs to another user through electronic mail. To send documents through email, you need to have installed an email application such as Microsoft Outlook or Lotus cc:Mail that supports either the Messaging Application Programming Interface (MAPI) or Vendor Independent Messaging (VIM). You can also send a copy of the active program to Microsoft Word.

To send a program to another user:

1 Open the program that you want to send. From the work area toolbar, select **Share** ⇨ **Code** ⇨ **Email**. The active program is added as an attachment to the email message.

2 Enter the email address of whomever you want to receive the file and send the message.

To send a program to Microsoft Word:

- Open the program that you want to send. From the work area toolbar, select **Share** ⇨ **Code** ⇨ **Send to Microsoft Word**. The active program is inserted into a new Microsoft Word document.

Export a Program to HTML

You can export your program to an HTML file so that you can easily include the program in another document or on a Web page. When you export the program, all of the indenting and keyword coloring are retained.

To export a program in HTML:

1 In the Program Editor, open the program that you want to export. Right-click anywhere in the Program Editor window and select **Export As HTML**.

2 Specify the name and location for the file and click **Save**.

Customizing the Program Editor

Customizing the Editor Appearance

By default, the SAS language editor highlights your text in different colors and font styles. This highlighting helps you easily distinguish the different elements of a SAS program and detect basic syntax errors more quickly. You can adjust the text color, font, font size, and font style to create and save your own appearance schemes by using the Enhanced Editor options. For more information, see "Setting SAS Program and Editor Options" on page 507.

Creating and Saving a New Scheme

By default, the SAS language editor highlights your text in different colors to help you distinguish the different elements of a SAS program. You can adjust the text color, font, font size, and font style to create and save your own appearance schemes.

To create and save a scheme:

1 Use the Editor Appearance options to customize the editor appearance for each element whose appearance you want to adjust. For more information, see "Setting Editor Appearance Options" on page 509.

2 Click **Save As**.

3 Enter a scheme name and click **OK**.

Note: You cannot delete the built-in schemes.

Assigning Keyboard Shortcuts in the Program Editor

By default, the Program Editor has many keyboard shortcuts that can make it easier for you to perform the most common commands when you are editing a program. In addition to the default shortcuts, you can also assign your own keyboard shortcuts to the commands in the Program Editor.

Note: If you want to assign a keyboard shortcut to a combination of commands, you can record an editor macro, and then assign a keyboard shortcut to the macro. For more information, see "Using Editor Macros" on page 321.

To assign a keyboard shortcut to a command in the SAS language editor:

1 Select **Program** ⇨ **Enhanced editor keys**.

2 Click **Assign Keys**.

3 Select a category of commands from the **Category** drop-down list.

4 From the **Commands** box, select the command for which you want to create a shortcut. If a keyboard shortcut is already defined for the command, the shortcut is displayed in the **Current keys** box.

   Note: You can assign more than one keyboard shortcut to a command.

5 Click inside the **Press new shortcut key** box. Press the key sequence that you want to use for the selected command. The key sequence is displayed in the **Press new shortcut key** box.

   Note: After you enter a value in the **Press new shortcut key** box, a message indicates whether the specified key sequence is currently assigned as a keyboard shortcut. If the key sequence is not currently assigned, you see **None**.

6 Click **Assign** to assign the shortcut to the command.

   Note: To reset the selected command to the default keyboard shortcut, click **Reset**.

To remove a keyboard shortcut:

1 Select **Program** ⇨ **Enhanced editor keys**.

2 Click **Assign Keys**.

3 Select a category of commands from the **Category** drop-down list.

4 From the **Commands** box, select the command for which you want to remove a shortcut.

5 In the **Current keys** box, select the shortcut key that you want to remove.

6 Click **Remove**.

Note: To reset all keyboard shortcuts to their default assignments, click **Reset All** in the Enhanced Editor Keys window. When you click **Reset All**, any keyboard shortcuts that you have changed or created are removed.

Creating User-Defined Keywords

You can create a list of user-defined keywords whose appearance you can customize. You might want to create user-defined keywords for programming elements such as SAS procedure statements, variables, and other user-defined formats.

To create user-defined keywords:

1 Select **Tools** ⇨ **Options**.

2 In the Options window, select **SAS Programs** in the selection pane. Click **Editor Options** and ensure that the **General** tab is selected.

3 Click **User-Defined Keywords**.

4 To add a new keyword, click **Add** and enter the name of the keyword in the name box. The new keyword is added to the list.

5 Click **OK** to save the keyword.

Note: You can customize the appearance of user-defined keywords by using the Enhanced Editor Appearance options. Select a file type of **SAS Program File** and a file element of **User-defined keywords**, and then specify the appearance options that you want to use. For more information, see "Setting Editor Appearance Options" on page 509.

10

Working with Stored Processes

About Stored Processes

A stored process is a SAS program that is designed to be stored on a central server and accessed by multiple remote clients. Stored processes consist of two distinct parts: the SAS code, and the stored process definition that resides on a metadata server.

Stored processes offer five primary benefits:

- Centralized storage of SAS code simplifies the process of managing and sharing code.
- The ability to include prompts in the code increases the capability to reuse code.
- SAS programs can be deployed in a variety of clients—for example, a client tool such as SAS Enterprise Guide or the SAS Add-In for Microsoft Office, a web service, a Java applet, or an application server that uses Enterprise JavaBeans technology.
- Users with little or no SAS programming experience can take advantage of the SAS Enterprise Guide drag-and-drop interface to build complex SAS programs that can be centrally located and executed.
- Security is improved by storing the code on a central server. Although users have permission to execute a program, developers can limit who has permission to view or edit the original SAS code.

Stored processes can be used to perform the following actions:

- stream HTML output to a web browser
- create output that can be consumed by the client or a middle-tier application
- publish results to channels or repositories
- create SAS data sets for subsequent processing by other applications or other stored processes
- serve as the basis for a web service

You must have an open project to create a stored process. For more information, see Chapter 4, “Working with Projects,” on page 27.

Adding an Existing Stored Process to an Open Project

To add an existing stored process to your open project:

1. Select **View** ⇨ **SAS Folders**. The list of SAS folders opens in the Resources pane.
2. Navigate through the tree view until you find the stored process that you want to add.
3. Right-click the stored process that you want to add and select **Add to Project**.

Note: You can also double-click a stored process to add it to the open project.

Creating Stored Processes

About Creating a Stored Process

You can create a stored process from the following sources:

- a code item that you created.
- a task item.

  Note: Before you can create a stored process from a task item, you must run or save the task.

- a process flow. You can create a stored process by using the entire process flow or a branch of the process flow. Creating a stored process from a process flow enables you to combine several tasks or queries into a single SAS program.

  Note: When creating a stored process from a process flow, options that are included in the tasks cannot be made into stored process prompts.

When you create a stored process from a task or a process flow, the SAS code from the task or process flow is automatically included. You can also create a blank stored process and use the Create New SAS Stored Process Wizard to add your SAS code. The SAS code is run when the stored process is executed.

You create a stored process by using the Create New SAS Stored Process Wizard. How you open the Create New SAS Stored Process Wizard depends on the source that you are using to create the stored process.

Opening the Create New Stored Process Wizard

The Create New SAS Stored Process Wizard guides you through the steps for creating a stored process. You can open the Create New SAS Stored Process Wizard in these ways:

- To create a stored process using the SAS code from a program or task, right-click on the program item or task item, and select **Create Stored Process**.
- To create a stored process using the SAS code from a process flow, right-click in the process flow and select **Create Stored Process**.
- To create a blank stored process, select **File** ⇨ **New** ⇨ **Stored Process** or right-click in the process flow and select **New** ⇨ **Stored Process**. You can add the SAS code for the stored process by using the Create New SAS Stored Process Wizard.

Note: If your data is accessed through the file system rather than through a SAS library, then the Data Not Contained in a SAS Library window might open before the Create New SAS Stored Process Wizard opens.

This window warns that the stored process might fail if the data for the code or task is not accessible from a SAS library. Because SAS Enterprise Guide is a desktop application, it has full access to your file system. The stored process server might not have the same access.

To ensure that the stored process server can access your data, insert it into your project by using a SAS library. If the data is saved in a SAS library, then the stored process server should be able to access the data, and the stored process should run correctly.

Name the Stored Process

1 Before you can name the stored process, you must open the Create New SAS Stored Process Wizard.

Note: To edit an existing stored process, double-click on the stored process in the Project pane or the process flow. The Stored Process Manager appears. In the selection pane, select **Name and Description** to change the name, description, or keywords for the stored process.

2 Specify the name of the stored process. The name is used to identify your stored process on the SAS Metadata Server. When you are creating a new stored process, by default the **Name** text box contains the name of the item that was selected when the Create New SAS Stored Process Wizard was invoked.

3 Specify a location in which to save the stored process on the metadata server.

4 (Optional) Provide a brief description of the stored process. A description helps other users better understand the function of the stored process. The description cannot exceed 200 characters.

5 (Optional) Specify the keywords that can be used to describe the stored process. Keywords help search utilities identify the stored process. Keywords are case sensitive, so "Key1" and "key1" are two different keywords.

Note: If you are defining a stored process to be called by a SAS BI Web Service, enter `XMLA Web Service` as a keyword. SAS BI Web Services does not recognize any stored processes without this keyword present in the metadata.

6 (Optional) Specify one or more users who are responsible for this stored process. Contact these users if you have questions or need to make changes to the stored process.

7 If you are connected to a SAS 9.3 (or later) server, you can also specify these options:

- **Upgrade to latest version** – This option is available in the Stored Process Manager when you are modifying a stored process that was created using SAS 9.2. If you select this option, then the stored process is upgraded to use the stored process features that are available in the latest SAS release. After you upgrade the stored process, it cannot be run on SAS 9.2 servers. If you do not select this option, then the stored process can be run on SAS 9.2 (and later) servers.
- **Make 9.2 compatible version** – If you select this option, then the stored process is created using only the options that were available in SAS 9.2. If you do not select this option, then the stored process that you create can be run only on SAS 9.3 (or later) servers.
- **Hide from user** – If you select this option, then users will not see this stored process in the client application.

  Note: Because SAS Enterprise Guide can be used to author stored processes, hidden stored processes are visible in the user interface. An example of a SAS product where these stored processes are hidden is SAS Add-In for Microsoft Office.

8 Click **Next**.

- If you opened the wizard from a task item, clicking **Next** enables you to specify the code options.

- If you opened the wizard from a SAS program or process flow, or if you are creating a blank stored process, clicking **Next** enables you to add SAS code to the stored process.

Add SAS Code to the Stored Process

1 Before you can add SAS code, you must specify the name of the stored process.

Note: If you are using the Stored Process Manager to edit a stored process, then select **SAS Code** in the selection pane to edit the SAS code.

2 If necessary, enter the SAS code for the stored process. If you are creating a stored process from a code item, task item, or process flow, then this code has already been added.

You can add code in these ways:

- Enter the SAS code in the Program Editor.
- Click **Replace with code from** to replace the existing code with code from an external file. You can select a file on your computer, on a SAS server, on the execution server, or from the project.

Note: The functionality of this Program Editor is similar to the functionality of the SAS language editor that is included in the work area.

3 Specify whether to include the code for stored process macros, global macro variables, LIBNAME references, or stored process definitions. If you are adding code from an external file, click **Include code for** to include the code for any of these items in the Program Editor. For more information, see Specifying code options.

4 Click **Next** to specify the execution options.

Specify Code Options

1 Before you can specify the code options for the stored process, you must specify the name of the stored process.

Note: If you are using the Stored Process Manager to edit a stored process, then select **SAS Code** in the selection pane to change the code options for your stored process.

2 Select the code that you want to include. You can choose from these options:

- **Stored process macros** includes the SAS macro variables in the code. By default, the macro variables are included.
- **Global macro variables** includes the %GLOBAL statement in the code. The %GLOBAL statement creates one or more global macro variables that are available during the entire execution of the stored process. These variables correspond to the prompts that you created. By default, the global macro variables are included.
- **LIBNAME references** includes the SAS LIBNAME statements in the section of code that is generated by SAS Enterprise Guide. Any LIBNAME statements that you include in your SAS code are included.

3 Click **Next** to specify the execution options.

Specifying the Execution Options

Specify the Execution Options

1 Before you can specify the execution options, you must specify the code options.

If you opened the wizard from a SAS program or process flow, or if you are creating a blank stored process, you must add SAS code to the stored process before you can specify the execution options.

Note: If you are using the Stored Process Manager to edit a stored process, then select **Execution Options** in the selection pane to specify the execution options.

2 Specify the execution options for your stored process. The options are different depends on whether you are creating a stored process that is compatible with SAS 9.2 or a stored process that is not compatible.

3 Click **Next**. If you opened the Create New SAS Stored Process Wizard from a code item or a task item that is not accessing data through a SAS server, then clicking **Next** enables you to create prompts for a stored process.

If you opened the wizard from a task item that is accessing data through a SAS server, then clicking **Next** enables you to assign a libref to your data.

Note: If you select the SAS Stored Process Server as the execution server, a warning might appear when you click **Next**. This warning appears because the SAS Stored Process Server might not be able to run a stored process that uses the Work or Sasuser library.

- If you are running code that reads from or writes to the Work library, the contents in the Work library might not be available either before or after the stored process is run.

- The Sasuser library on the stored process server is different from the Sasuser library that is used by the tasks. If your stored process code writes the output data set from a task, that output data is saved to the Sasuser library by default. However, the Sasuser library on the stored process server is read-only. To ensure that the stored process runs successfully, change all references to the Sasuser library to a library that you have set up to be available for future use.

Specify the Execution Options for a Stored Process That Is Compatible with SAS 9.2

Note: These options are available in the Create New Stored Process wizard and in the SAS Process Manager if you are working with a stored process that is compatible with SAS 9.2.

1 From the **Execution server** drop-down list, select the server where you want to run the stored process. Your SAS source code is saved on this server.

2 From the **Source filepath** drop-down list, select the location where you want to save your SAS code.

   Note: The root location that appears in the Add SAS Source Code Location to Metadata dialog box depends on the file navigation settings that are defined for the workspace server. Your administrator defines these settings by using SAS Management Console or SAS Enterprise Guide Explorer.

   Note: Stored processes can be stored only in locations that are defined on the metadata server. References to the locations in the **Source filepath** drop-down list are stored on the metadata server, but they refer to actual locations on the execution server.

3 In the **Source filename** box, enter the file name for the source code. By default, this file name is the stored process name with a file extension of .sas.

4 Review the result types. By default, SAS Enterprise Guide selects the result types that are supported by the stored process. In general, leave these defaults. If you need more control over the output options, use a shared prompt. For more information, see Creating prompts and output parameters for a stored process and About shared groups.

   Here are the result types that are available in SAS Enterprise Guide:

   Streaming

   A single stream of output data (for example, a web page) is returned to the application that ran the stored process. No output file is stored on the server. Instead, you view the output immediately after it is generated. New output is generated each time you run the stored process.

By default, an output stream called _WEBOUT is automatically defined.

Note: The Streaming option is available only if your execution server is a stored process server.

Note: Streaming output returns only one file with the result set. Some SAS procedures embed JPEG or GIF images in their ODS results. If the tasks in your stored process use any of these procedures, and if you selected streaming output, then the image files are not returned when you execute the stored process. Broken image links appear in your HTML results.

To prevent broken images, set the output type of the stored process to **Package**. This output type combines the HTML page and any image files into a single package before returning the results to the client.

Package

One or more files are created by the execution server and combined into one result package that is returned to the client application. The client application then unpacks the contents of the package and displays the files appropriately. This option is the preferred output method for SAS Enterprise Guide.

Note: If the stored process supports a transient package of files, then SAS Enterprise Guide forces the stored process to return output in a transient package even if the stored process is defined to create another type of output.

5 Click **Next**. If you opened the Create New SAS Stored Process Wizard from a code item or a task item that is not accessing data through a SAS server, then clicking **Next** enables you to create prompts for a stored process.

If you opened the wizard from a task item that is accessing data through a SAS server, then clicking **Next** enables you to assign a libref to your data.

Note: If you select the SAS Stored Process Server as the execution server, a warning might appear when you click **Next**. This warning appears because the SAS Stored Process Server might not be able to run a stored process that uses the Work or Sasuser library.

- If you are running code that reads from or writes to the Work library, the contents in the Work library might not be available either before or after the stored process is run.
- The Sasuser library on the stored process server is different from the Sasuser library that is used by the tasks. If your stored process code writes the output data set from a task, that output data is saved to the Sasuser library by default. However, the Sasuser library on the stored process server is read-only. To

ensure that the stored process runs successfully, change all references to the Sasuser library to a library that you have set up to be available for future use.

Specify the Execution Options for a SAS 9.3 Stored Process

Note: These options are available in the Create New Stored Process Wizard and in the SAS Process Manager if you are connected to a SAS 9.3 (or later) server.

1 From the **Application server** drop-down list, select the name of the default application server context that will run the stored process. The application server context defines the environment in which the stored process executes. Application server contexts typically contain multiple server definitions. This field is required.

2 From the **Server type** area, specify the type of server that will run the stored process. The default selection is **Default server**. When you select this option, the client application chooses the best server (either SAS Stored Process Server or SAS Workspace Server) to use.

   Note: Your SAS Stored Process Server and your SAS Workspace Server probably run under different credentials and might not have the same access permissions. As a best practice, store your SAS source code in your metadata. If you write your SAS source code to a specific server, select that server and not the **Default server** option.

   Select **Stored process server only** if you want only the stored process to be executed by a stored process server. Select this option if the stored process uses sessions or if it uses replay (for example, to produce graphics in streaming output).

   Select **Workspace server only** if you want only the stored process to be executed by a workspace server. Select this option if the stored process must be run under the client identity. This option can provide better protection for secure resources.

3 Specify the source code location and execution. You can choose from these options:

   - To run the stored process on different application servers, select **Allow execution on other application servers**, so the source code is saved in the metadata.
   - To run the stored process only on the specified application server, select **Allow execution on selected application server only**. Then you can specify whether to save the source code in the metadata or on the application server.

4 Review the result types. By default, SAS Enterprise Guide selects the result types that are supported by the stored process. In general, leave these defaults. If you need more control over the output options, use a shared prompt. For more information, see Creating prompts and output parameters for a stored process and About shared groups.

Here are the result types that are available in SAS Enterprise Guide:

Streaming

A single stream of output data (for example, a web page) is returned to the application that ran the stored process. No output file is stored on the server. Instead, you view the output immediately after it is generated. New output is generated each time you run the stored process.

By default, an output stream called _WEBOUT is automatically defined.

Note: The Streaming option is available only if your execution server is a stored process server.

Note: Streaming output returns only one file with the result set. Some SAS procedures embed JPEG or GIF images in their ODS results. If the tasks in your stored process use any of these procedures, and if you selected streaming output, then the image files are not returned when you execute the stored process. Broken image links appear in your HTML results.

To prevent broken images, set the output type of the stored process to **Package**. This output type combines the HTML page and any image files into a single package before returning the results to the client.

Package

One or more files are created by the execution server and combined into one result package that is returned to the client application. The client application then unpacks the contents of the package and displays the files appropriately. This option is the preferred output method for SAS Enterprise Guide.

Note: If the stored process supports a transient package of files, then SAS Enterprise Guide forces the stored process to return output in a transient package even if the stored process is defined to create another type of output.

5 Click **Next**. If you opened the Create New SAS Stored Process Wizard from a code item or a task item that is not accessing data through a SAS server, then clicking **Next** enables you to create prompts for a stored process.

If you opened the wizard from a task item that is accessing data through a SAS server, then clicking **Next** enables you to assign a libref to your data.

Assigning a Libref to Access Your Data

What Is a Libref?

A libref is an alias for the location of a SAS library. You can map a libref to a SAS library by using a SAS LIBNAME statement. The syntax for this statement is `LIBNAME` *`libref`* `'`*`full-path-to-SAS-library`*`'`.

When Should You Use This Step?

Because a task depends on data to run successfully, it is important that the stored process can find and access the data at run time. If your data is stored in either the Maps, Sashelp, Sasuser, or Work library, then you do not need to create a libref to access your data. If your data is not stored in one of these SAS libraries, then you must use this step to assign a libref that points to the location of your data.

When you run or save a task that uses data that is not stored in one of these SAS libraries, SAS Enterprise Guide automatically includes in the code a LIBNAME statement that maps a libref to the location of the data. When you initially create a stored process, you can probably use the LIBNAME statement that SAS Enterprise Guide generated. If you move the data to a new location, then you need to use the Stored Process Manager to edit the LIBNAME statement.

Assign a Libref to Access Your Data

1 Before you can assign a libref, you must specify the execution options.

Note: If you are using the Stored Process Manager to edit a stored process, then select **Librefs** in the selection pane to assign a libref.

2 From the **References to libraries requiring the generation of a LIBNAME statement** box, select a library. SAS Enterprise Guide adds to the code a LIBNAME statement that identifies this library. This library must be available on all servers that run the stored process.

Note: The **References to built-in libraries** box shows libraries that are accessible without using a LIBNAME statement. Examples of the libraries include Maps, Sashelp, Sasuser, and Work libraries. This list of libraries can differ from server to server, so verify that these libraries are available on all the servers that run the stored process. If you do not need to specify an additional

LIBNAME statement, click **Next** to create prompts and output parameters for a stored process.

3 To define a custom LIBNAME statement, select the **Use custom LIBNAME statement** check box. Enter your changes in the text box. These changes automatically appear in the **LIBNAME statement** box.

4 Click **Next** to create prompts and output parameters for a stored process.

Creating Prompts and Output Parameters for a Stored Process

What Are Prompts?

Including prompts in your code increases your ability to reuse code and also enables the code to be customized when the prompt is run. Stored process prompts typically correspond to macro variables that are included in the SAS code.

You can group similar prompts into a single category. How these groups are displayed depends on the interface that you use when the prompt is run. If you run the stored process in SAS Enterprise Guide, then each group is a separate pane in the user interface. The label for the pane is the name that you specified for the group. If you run the stored process from a Web application (such as SAS Web Report Studio), then the groups are represented by tables. The title for the table is the name that you specified for the group.

What Are Output Parameters?

Including output parameters in your code enables you to submit values to the client application when a stored process runs successfully. Output parameters are used mainly with SAS BI Web Services. The name of the output parameter is the name of the SAS macro variable that is associated with the output parameter. The output parameter can be any of the following types: Date, Double, Integer, Time, TimeStamp, or String.

Add Prompts and Output Parameters to Your Stored Process

1 If you opened the wizard from a code item or a task item that is not accessing data through a SAS server, then before you can create a prompt, you must specify the execution options.

If you opened the wizard from a task item that is accessing data through a SAS server, then you must first assign a libref to access your data.

Note: If you are using the Stored Process Manager to edit a stored process, then select **Prompts** in the selection pane to add, edit, or delete prompts.

2 Add prompts to the stored process.

- To create a new prompt, click **New** and select **New Prompt**. The New Prompt dialog box appears.
- To add a prompt from the SAS code, you can perform the following tasks:
  - □ To add a single prompt from the SAS code, click **New** and select **Prompt from SAS Code for ⇨ *prompt-name***.
  - □ To add all of the prompts from the SAS code, click **New** and select **Prompt from SAS Code for ⇨ Multiple**. The Edit Prompt dialog box appears for each prompt that you are adding.
- To add a prompt from the project, click **New** and select **Project prompt copy ⇨ *prompt-name***. A copy of the project prompt is added to the stored process. This copy of the prompt is unique to the stored process. Any changes that you make to this prompt are not saved to the project prompt.

Note: Nested groups are not supported by all applications that run stored processes. Consequently, the prompts that you grouped might not be displayed in nested groups in certain applications. However, these prompts are still available to users who are running the stored process.

3 Add output parameters.

a Click **New**. The Create an Output Parameter dialog box appears.

b Specify a name for the SAS macro variable that holds the value of the output parameter. SAS macro variable names must start with an alphabetic character (A–Z or a–z) or an underscore (_) and can contain only alphanumeric characters (A–Z, a–z, or 0–9) or underscores. SAS macro variable names are limited to 32 characters, and must be unique within a stored process.

c Select the type for the output parameter. You can choose from the following types: Date, Double, Integer, Time, TimeStamp, or String.

d (Optional) Specify a label for the output parameter. If you specify a label, then this text is displayed to the user instead of the name of the macro variable. Slashes, backslashes, and control characters cannot be used in this field. The label can contain up to 60 characters.

e (Optional) Specify a description for the output parameter.

For more information about output parameters, see SAS Stored Process: Developer's Guide.

4 Click **Next** to define the data sources to use as input and specify the data targets for output.

Defining the Data Sources to Use as Input and Specifying the Data Targets for Output

You can define any data sources that you want to use as input when the stored process runs. These data sources are also called input streams. You can also specify any data targets where you want to send output when the stored process runs. These data targets are also called output streams.

1 Before you can add a data source or data target, you must create prompts for the stored process.

Note: If you are using the Stored Process Manager to edit a stored process, then select **Data Sources and Targets** in the selection pane to specify an input data source or data target.

2 To create a new data source or data target, click **New**. The Create a New Data Source dialog box appears.

3 To create generic (byte-stream) data or an XML stream:

a In the **Form of Data** area, select **Generic (byte-stream)** data or **XML based data**. A generic stream can contain any value (such as binary data or text). An XML stream should contain well-formed XML. You can also attach an XML schema to the input stream.

b Specify the file reference for the input stream. The fileref is limited to eight characters, and can contain alphanumeric characters (A–Z, a–z, or 0–9), dollar signs ($), number signs (#), at signs (@), and underscores (_). A fileref is required.

c (Optional) To specify whether to allow the stored process to make multiple passes over the data, select the **Allow rewinding stream** check box. Select this option when the stored process uses the XML library engine to read the input data without specifying an XMLMap of the input data. When an XMLMap is not specified, the XML library engine requires multiple passes over the data.

d (Optional) Specify a label for the input stream.

e (Optional) Specify a brief description of the input stream. A description helps your users to better understand the function of the input stream.

f To attach an XML schema for XML-based data, click the **Schema** tab and specify these options:

- the schema URI (the location of the XML schema). To use a published schema when defining a data source, click **Browse**.

- the namespace where the data type or element is defined.
- the name of the data type or element that defines the data source.
- whether the name is a schema element or a schema type.
- whether to embed the schema fragment inside the generated Web Services Description Language (WSDL) or whether to reference the fragment with a URI.

4 To create a data table:

   a In the **Form of Data** area, select **Data Table**.

   b (Optional) Specify a label for the input stream.

   c (Optional) Specify a brief description of the input stream. A description helps your users to better understand the function of the input stream.

   d In the **Table parameter name** box, specify the macro name of the SAS table. The macro variables are prefixed with "_SOURCE" or "_TARGET," and the value of the macro variable is the name of the template. This field is required.

   e (Optional) Specify the name of the table to use as a template. Note You can create a data table only if you are connected to a SAS 9.3 server.

5 In the Create a Data Source dialog box, click **OK**. The name of the data source or data target now appears in the list of input or output streams.

6 Click **Next** to review a summary of the stored process.

Review a Summary of the Stored Process

1 Before you can review the summary information, you must create prompts for a stored process.

Note: If you are using the Stored Process Manager to edit a stored process, then select **Summary** in the selection pane to view a summary of the stored process.

2 Review the information in the text box. You can choose to scroll through the summary or to display the results by page. By default, only a portion of the SAS code is shown. To view all of the SAS code for the stored process, select **Show full SAS code**.

3 To change a value, click **Back**.

Note: To change a value in the Stored Process Manager, select the page that contains the option that you want to change from the selection pane.

4 Select **Run stored process when finished** so that SAS Enterprise Guide can run the stored process when you click **Finish**. By default, this option is not selected, and SAS Enterprise Guide saves the stored process but does not run it.

Note: This option is available only from the Create New SAS Stored Process Wizard.

5 Click **Finish** to run or save the stored process.

Note: If you copied and pasted the stored process code from an email, then the stored process code is different from the code in your project. When you create a stored process from a code item, SAS Enterprise Guide must modify the code before saving the code to the stored process server. To update the code in your project, click **Yes** in the message that appears.

Troubleshooting: Creating Stored Processes

Problem	Solutions
When I try to create a stored process, I get an error message that says that I do not have permission to access the specified location. I know that I have permission to this network location because I have accessed other files there. What happened?	You could have insufficient authorization to access a file if your user account is not the account that was created when the server was configured. When configuring the BI Server plug-in in SAS Management Console, your site's SAS administrator created a user account for stored processes (for example, sassrv). The purpose of this user account is to enable users at your site to create and run stored processes. If you create a file and are using your user account, then the file is associated with your user account, not the user account for stored processes. Therefore, you might not be able to access that file when you create a stored process. If you cannot access a file during the creation of a stored process, then contact your SAS administrator.
When I specify the execution options, a .sas extension is added to the file name for the source code in the **Source filename** box. How do I prevent this from happening?	If you are saving this file to a Windows location, then you can enter the file name in double quotation marks without the .sas extension. SAS Enterprise Guide automatically adds the extension when it saves the file. If you are saving this source code to a server that cannot handle file names with extensions, such as an MVS server, then you must enter the file name in double quotation marks. For example, if you enter `mystoredprocess`, the mystoredprocess file is created on the server, and the .sas extension is not added.

Editing Stored Processes

Editing an Existing Stored Process

After a stored process has been added to the current project, you can use the Stored Process Manager to modify an existing stored process. To open the Stored Process Manager, double-click the stored process in the Project pane or the process flow or right-click the stored process in the Project pane or process flow and select **Modify *name-of-stored-process***.

Changing the Format of the Results

By default, the format of your results is specified in the Options dialog box. By using the Properties window, you can customize the results that appear when the stored process is run within SAS Enterprise Guide.

Note: Changing the result format affects only the selected stored process. It does not change the value that is specified in the Options dialog box.

1. Right-click on the stored process and select **Properties**.
2. In the Properties window, select **Customize results formats, styles, and behavior**.
3. Select the format for your results and the style.
4. Select the graph format to determine the image format for graphs in your results.
5. Click **OK**.

Running Stored Processes

Running a Stored Process in SAS Enterprise Guide

1 Right-click on the stored process and select **Run**.

2 Specify the values of any prompts.

3 Click **Run**.

Running a Stored Process in SAS Add-In for Microsoft Office

Stored processes that are stored in a central repository at your site can be run in several SAS applications, including the SAS Add-In for Microsoft Office. The _MSOFFICECLIENT macro variable is unique to the SAS add-in. This macro variable specifies the Microsoft application that is currently executing the stored process. Valid values for this macro variable are Excel, Word, PowerPoint, and Outlook.

Running a Stored Process in JMP

In JMP 7 or later, you can run SAS stored processes. By using the JMP Stored Process Packager in SAS Enterprise Guide, you can add additional JMP functionality to the results package for the stored process. This functionality is available only when you run the stored process in JMP. The SAS program that is generated by the JMP Stored Process Packager is not run as part of the SAS Enterprise Guide project.

The JMP Stored Process Packager acts like any other task in SAS Enterprise Guide. You can create a stored process from this task by right-clicking on the object in the process flow and selecting **Create Stored Process**. You can link this object to other objects (such as SAS tasks or queries) in the process flow. If you create a stored process from a task that links to the JMP Stored Process Packager, then the stored process code automatically includes the task code and the code for the additional JMP functionality.

To create the package:

1 Select **Tools** ⇨ **JMP Stored Process Packager**.

2 Enter a name for the package.

3 Add a data set to the package contents.

4 Add a JSL script to the package contents. In JMP, copy the code for the JSL script. Then in SAS Enterprise Guide, paste the code in the Script Properties dialog box.

5 From the **Link the package step after this task** drop-down list, select the task that you want to link the package to.

6 Click **OK** to create the package.

Note: If you are creating a custom package, do not include the stored process macros in your code. The stored process macros save your output in the Stpwork library. To remove the stored process macros from your code, clear the **Stored process macros** check box when you create the stored process.

If you need to change the source data set or JSL script after you have created the stored process, you can specify these changes in the JMP Stored Process Packager and then update the code on the stored process server.

Note: You can edit the code for this stored process in other SAS applications. SAS Enterprise Guide automatically adds comments to the code when the stored process is created. Review these comments carefully to understand each line of code in the stored process.

Troubleshooting: Running Stored Processes

Problem	Solution
When I try to run a stored process, I get an error message that says that I do not have permission to access the specified location. I know that I have permission to this network location because I have saved other files there. What happened?	The execution of a stored process could fail if the stored process server cannot read from or write to the network location that you specified for your SAS source code. When configuring the BI Server plug-in in SAS Management Console, your site's SAS administrator created a user account, such as sassrv. If the sassrv user account does not have permission to read from or write to the specified network location, then the stored process fails. To execute the stored process, ask your SAS administrator to give Read and Write permission to the sassrv account for the network location that you specified.
When I try to run a stored process, I get an error in the log	If the data set that you are using for the stored process is open in SAS Enterprise Guide, then this

Problem	Solution
that says that the delete lock for the data set is not available and that the data set is being used by another process. What happened?	error appears in the SAS log when you try to run the stored process. To successfully run the stored process, you must first close any input data sets.

Updating the SAS Code in a Stored Process

Previewing the Stored Process Code

You can preview the stored process code while you are creating a stored process or after you save a stored process.

- In the Create New Stored Process Wizard, you can preview the code for the task or program that is the basis for the stored process. However, the code does not include the macro variables for any prompts that you might add. These macro variables are added to the code when you save the stored process.

  Note: You cannot edit the code of a task after it becomes part of a stored process. If you need to add code or modify the existing code for a task, then these changes must be made in the task code before it is added to the stored process.

- In the Stored Process Manager, you can preview the code for the task or program that is the basis for the stored process. The code elements that are displayed depend on the code options that you specified when you created the stored process. For more information, see “Specify Code Options” on page 352.

  You can also preview the code for any prompts that you added to your stored process. For more information, see “Understanding Macro Variables and Prompts” on page 373.

Synchronizing the Program in the Project with the Stored Process

After you create a stored process from a program item, you might need to change your SAS code. You can update the program in your project and then save the program to your stored process without having to open the Stored Process Manager.

To change the program on the stored process server :

1 Double-click on the program item in the Project pane or the code object in the process flow. The program opens in the Program Editor.

2 Edit your program.

Note: You do not have to save your changes before proceeding to the next step.

3 Right-click the stored process, and select **Modify**. If the code in the project is different from the code on the server, the Compare SAS Code Files window appears. The **EG Code** and **Server Code** boxes show the differences between the project code and the server code.

4 To save the project code to the stored process server, click **Use EG Code**.

Note: If you click **Use Server Code**, the code in the project is changed to match the code on the server.

Updating Task Code in a Stored Process

After you create a stored process from a task item, you might need to make changes to the task code. You can write these code changes to your stored process.

To change the code on the stored process server:

1 Right-click the task item and select **Modify**. The task window appears.

2 Run or save the task.

3 Right-click the stored process, and select **Modify**. If the code in the project is different from the code on the server, the Compare SAS Code Files window appears. The **EG Code** and **Server Code** boxes show the differences between the project code and the server code.

4 To save the project code to the stored process server, click **Use EG Code**.

Note: If you click **Use Server Code**, the code in the project is changed to match the code on the server.

Emailing a Stored Process

Moving stored processes from one server to another can be difficult. An easier solution is to email the stored process code to share the code with another user. When you email the stored process code, the SAS code is appended to the metadata definitions for the stored process. The email recipient can copy and paste the stored process code into a code item in SAS Enterprise Guide and create a stored process from this code.

To email a stored process:

1 Right-click the stored process and select **Modify**. The Stored Process Manager appears.

2 View the code for the stored process.

   Note: You cannot copy the code from the Program Editor in the Create New SAS Stored Process Wizard or the SAS Code pane in the Stored Process Manager. This code does not include the metadata that is required to run the stored process.

3 Copy the stored process code and paste it into an email.

To create a stored process from code that is sent by email

1 Select **File** ⇨ **New** ⇨ **Program** to open a new code item.

2 Copy the stored process code from the email and paste it into the new code item.

3 After the code is added, right-click the code item and select **Create Stored Process**. The Create New SAS Stored Process Wizard appears.

Note: Some settings for stored processes depend on the specific metadata environment. The settings are preserved when a stored process is being moved between two similar environments. If the stored process is being moved to a different type of environment (such as one that has servers with different names), then default values are used for these options.

Stored Process Properties

To view the properties for a stored process item, right-click the stored process name and select **Properties**.

- **Use preferences from Tools->Options** uses the settings that are specified in the Options dialog box for this stored process. The other results options are disabled if this option is selected.
- **Customize result formats, styles, and behavior** enables you to override the settings that are specified in the Options dialog box for this stored process.
- **Result Format**
  - **SAS Report** generates the results as a SAS Report. You can also select a style from the drop-down list. The style is applied to the SAS Report results for this task. SAS Report results can be used to create a report.
  - **HTML** generates results in HTML format. You can also select a style from the drop-down list. The style is applied to the HTML results for this task.
  - **PDF** generates results in PDF format. You can also select a style from the drop-down list. The style is applied to the PDF results for this task.
  - **RTF** generates results in RTF format. You can also select a style from the drop-down list. The style is applied to the RTF results for this task.
  - **Graph Format** determines the image format for graphs in results. To change the format, select another format from the drop-down list.

  Note: You can select the drop-down list next to the result format to specify the style of the results.

- **Last Execution Time** displays how long it took for the stored process to run the last time it was executed. This time is rounded to the nearest second and does not include the time it takes to connect to the server.

See Also

“About the Style Manager” on page 531

11

Working with Prompts

About Prompts

Prompts enable you to request input from the user when code is run. Including prompts in your code increases your ability to reuse code and also enables the code to be customized using the value that you enter for the prompt. You can define prompts that require the user to select or enter one or more values when a task, query, program, or stored process is run. Prompts can be used to provide the name of a variable in order to run a task or to provide a value for a filter expression in a query or program. When the prompt is run, one or more macro variables are automatically generated for each type of prompt.

Without prompts, the code that you run is static even though the data might change. For example, you could have a static query to determine which products had sales greater than USD 1500 in the 4th quarter of 2017. With prompts, both the data and the code can change. For example, you could create a query with prompts to determine which products had sales between *x* and *y* for the *n*th quarter of 2017. The values of *x*, *y*, and *n* are supplied by the user each time the query is run.

If you are creating prompts for a stored process, then you can organize similar prompts into a group. When you run the stored process in SAS Enterprise Guide, the groups are represented by tabs in the user interface.

For stored process prompts, you can also create dependencies between prompts. A dependency occurs when the values for one prompt depend on the value that was selected for another prompt. For example, you create a prompt called Divisions and a prompt called Departments. You want your users to see only the departments for the selected division, so you create a dependency between these two prompts. When the prompts run, the user first selects a value for the Divisions prompt. Because the prompts are dependent, the only available values for the Departments prompt are the departments in that division.

For more information about creating a prompt, see “About Creating Prompts” on page 385.

After you create a prompt, you can copy and paste it within the same project or into another SAS Enterprise Guide project.

Note: The Prompt Manager is available only if you create a project.

Understanding Macro Variables and Prompts

Macro Variables for Prompt Types

SAS macro variables are used by programs such as stored processes. When a prompt runs, one or more global macro variables are automatically generated in the SAS code. Macro variables are generated for prompt types and for special values. To view the macro variables for a specific prompt, right-click on the name of the prompt in the Prompt Manager and select **Show Macro Variables**.

In the stored process code, here is the syntax for the macro variable: `%LET` *`PROMPT-NAME`* `= user-value;`

In this syntax, *PROMPT-NAME* is the name that you specified when creating the prompt, and user-value is the value that the user specifies when the prompt is run. You can also use the **Prompt Manager** pane to view the macro variables that are associated with a prompt.

If no value is specified for a single value prompt, an empty macro variable is generated. The value of this empty macro variable is a period (.) for numeric prompts and a space ( ) for text prompts. If no value is specified for multiple value prompts, then a macro variable named *PROMPT-NAME*_COUNT is generated. The value of this macro variable is `0`.

For many prompt types, multiple macro variables are generated. Extensions and an ordinal number, if necessary, are added to the name of the macro variable in order to create a unique name for each macro variable. The following table lists the extensions for each prompt type and gives an example of each.

Text, Numeric, Single Value Prompts

These prompts have no extension.

Suppose that you create a single, user-entered value text prompt called Products. When the prompt is run, the user sees a list of products. The user selects Cars as the product, and the following macro variable is generated:

```
%LET PRODUCTS = Cars;
```

Text, Numeric, Multiple Value Prompts

If you create a multiple value prompt, then multiple macro variables are generated. The base macro variable is the name of the prompt without any extensions. Extensions are used to create a unique name for additional macro variables. For multiple value prompts, these extensions are `_COUNT` and `0` through ***n***.

Suppose that you create a multiple value text prompt called Products. When the prompt is run, the user selects Cars, Dolls, and Games from a list of values. Six macro variables are generated for the three values that the user selected. Extensions are added to the prompt name, so the name of each macro variable is unique.

Syntax of Macro Variables	Explanation of Value
`%LET PRODUCTS = Cars;`	The base macro variable contains the first value that is specified for the prompt.
`%LET PRODUCTS_COUNT = 3; %LET PRODUCTS0 = 3;`	The number of values that are specified for the prompt.
`%LET PRODUCTS1 = Cars;`	The first value that the user specified for the prompt.
`%LET PRODUCTS2 = Dolls;`	The second value that the user specified for the prompt.
`%LET PRODUCTS3 = Games;`	The third value that the user specified for the prompt.

Text Range, Numeric Range

Here are the extensions:

- `_MIN` - the lower bound of the range.
- `_MAX` - the upper bound of the range.

Suppose that you create a numeric range prompt called Age. When the prompt is run, the user specifies a maximum value of 20 and a minimum value of 5.

Syntax of Macro Variables	Explanation of Value
`%LET AGE_MAX = 20;`	The macro variable for the maximum value.

Syntax of Macro Variables	Explanation of Value
`%LET AGE_MIN = 5;`	The macro variable for the minimum value.

Date Prompts

Here are the extensions for date prompts:

- `_END` - the end date in the format (*ddmmyyy*) of the week. month, quarter, or year. This extension is used only if the date type is **Week**, **Month**, **Quarter**, or **Year**.
- `_LABEL` - the formatted or relative value of the date. Here are the formats for each date type.

Date Type	Format
Day	*month-name dd, yyyy*
Week	Week *ww yyyy*
Month	*month-name yyyy*
Quarter	*quarter-name* quarter *yyyy*
Year	*yyyy*

- `_REL` - the value of the relative date. This extension is used only if the user selects a relative date (such as yesterday and *N* weeks) for the prompt.

Suppose that you create a date prompt called ClosingDate. When the prompt is run, the user selects the last date in the sales period.

Syntax of Macro Variables	Explanation of Value
`%LET CLOSINGDATE = 04Apr2010;%LET CLOSINGDATE_LABEL = April 04, 2010;`	If the date type is **Day** and the user selects a specific date (such as April 04, 2010), then two macro variables are generated. - The CLOSINGDATE macro variable displays the date in the *ddmmmyyyy* format. - The CLOSINGDATE_LABEL macro variable displays the full name of the month, a two-digit integer for the day, and a four-digit integer for the year.

Syntax of Macro Variables	Explanation of Value
`%LET CLOSINGDATE = 01Jan2009;%LET CLOSINGDATE_END = 31Jan2009;%LET CLOSINGDATE_LABEL = January 2009`	If the date type is **Week**, **Month**, **Quarter**, or **Year**, then three macro variables are generated. In this example, the date type is **Month**. When the prompt is run, the user selects January 2009 as the value of the prompt. ■ The CLOSINGDATE macro variable is the first day of the month (01Jan2009), and the CLOSINGDATE_END macro variable is the last day of the month (31Jan2009). ■ The CLOSINGDATE_LABEL macro variable shows the full name of the month and a four-digit integer for the year.
`%LET CLOSINGDATE = 04Apr2010;%LET CLOSINGDATE_LABEL = Yesterday;%LET CLOSINGDATE_REL = D-D1;`	If the user selects a relative date, then three macro variables are generated. In this example, the user selects **Yesterday** as the relative date. The CLOSINGDATE macro variable is today's date (04Apr2010), and the CLOSINGDATE_REL macro variable is D-D1 to represent yesterday's date.

Date Range

Here are the extensions for Date range prompts:

- `_MIN` - the first day of the week, month, quarter, or year that is the minimum value of the range. If the date type is **Day**, then the value of `_MIN` is the first day of the range.
- `_MAX` - the first day of the week, month, quarter, or year that is the maximum value of the range. If the date type is **Day**, then the value of `_MAX` is the last day of the range.
- `_MIN_END` - the last day of the week, month, quarter, or year that is the minimum value of the range.
- `_MAX_END` - the last day of the week, month, quarter, or year that is the maximum value of the range.
- `_MIN_LABEL` - the formatted or relative minimum value of the range. Here are the formats for each date type.

Date Type	Format
Day	*month-name dd, yyyy*

Date Type	Format
Week	Week *ww yyyy*
Month	*month-name yyyy*
Quarter	*quarter-name* quarter *yyyy*
Year	*yyyy*

- `_MAX_LABEL` - the formatted or relative maximum value of the range. Here are the formats for each date type.

Date Type	Format
Day	*month-name dd, yyyy*
Week	Week *ww yyyy*
Month	*month-name yyyy*
Quarter	*quarter-name* quarter *yyyy*
Year	*yyyy*

- `_MIN_REL` - the value of the relative date for the minimum value. This extension is used only if the user selects a relative date (such as yesterday and *N* weeks) for the prompt.
- `_MAX_REL` - the value of the relative date for the maximum value. This extension is used only if the user selects a relative date (such as yesterday and *N* weeks) for the prompt.

Suppose that you create a date range prompt called SalesRange. When the prompt is run, the user specifies the beginning and end dates in a sales period.

Syntax of Macro Variables	Explanation of Value
`%LET SALESRANGE_MAX = 31Dec2006%LET SALESRANGE_MAX_LABEL = December 31, 2006%LET SALESRANGE_MIN = 01Jan2006%LET SALESRANGE_MIN_LABEL = January 01, 2006`	If the user selects specific beginning and end dates (such as January 01, 2006, and December 31, 2006), then four macro variables are generated. ■ The SALESRANGE_MAX and SALESRANGE_MIN macro variables display the date in *ddmmmyyyy* format. ■ The SALESRANGE_MAX_LABEL and SALESRANGE_MIN_LABEL macro variables display the full name of the month, a two-digit

	integer for the day, and a four-digit integer for the year.
%LET SALESRANGE_MAX = 26Jul2010;%LET SALESRANGE_MAX_END = 01Aug2010;%LET SALESRANGE_MAX_LABEL = Week 30 2010;%LET SALESRANGE_MIN = 24May2010;%LET SALESRANGE_MIN_END = 30May2010;%LET SALESRANGE_MIN_LABEL = Week 21 2010;	If the date type is **Week**, **Month**, **Quarter**, or **Year**, then four macro variables are generated. In this example, the date type is **Week**. When the prompt is run, the user selects a date range of Week 21 2010, to Week 30 2010. - The SALESRANGE_MAX and SALESRANGE_MAX_END show the first and last days for the maximum value. - The SALESRANGE_MAX_LABEL macro variable shows that the user selected Week 30 2010, for the maximum value. - The SALESRANGE_MIN and SALESRANGE_MIN_END macro variables show the first and last days for the minimum value. - The SALESRANGE_MIN_LABEL shows that the user selected Week 21 2010, for the minimum value.
%LET SALESRANGE_MAX = 01Jan2008;%LET SALESRANGE_MAX_END = 31Dec2008;%LET SALESRANGE_MAX_LABEL = Current year;%LET SALESRANGE_MAX_REL = Y0Y;%LET SALESRANGE_MIN = 01Jan1998;%LET SALESRANGE_MIN_END = 31Dec1998;%LET SALESRANGE_MIN_LABEL = 10 years ago%LET SALESRANGE_MIN_REL = Y-10Y;	If the user selects relative dates for the beginning and end dates, then eight macro variables are generated. In this example, the date type is **Year**. When the prompt is run, the user selects a date range of 10 years ago to the current year. - The SALESRANGE_MAX and SALESRANGE_MIN macro variables display the first and last day of the current year (2008). - The SALESRANGE_MAX_LABEL and SALESRANGE_MIN_LABEL display the values that the user selected when the prompt is run. The macro variables for the minimum value show the first and last day *N* years ago, where *N* equals 10. - The SALESRANGE_MIN_REL macro variable is Y-10Y to represent 10 years ago.

Time Prompts

Here are the extensions for time prompts:

- `_LABEL` - the relative or formatted time value. The format for time value is *hh:mm:ss* AM\|PM.

- _REL - the value of the relative time. This extension is used only if the user selects a relative time (such as the current time) for the prompt.

Suppose that you create a time prompt called ClosingTime. When the prompt is run, the user specifies the closing time in the sales period.

Syntax of Macro Variables	Explanation of Value
`%LET CLOSINGTIME = 13:00:00;%LET CLOSINGTIME_LABEL = 01:00:00 PM;`	If the user selects a specific time, then two macro variables are generated. ■ The CLOSINGTIME macro variable displays the time in a 24-hour format. ■ The CLOSINGTIME_LABEL macro variable displays the time in a 12-hour format and includes the AM or PM label.
`%LET CLOSINGTIME = 13:00:00;%LET CLOSINGTIME_LABEL = Next hour;%LET CLOSINGTIME_REL = H1H;`	If the user selects a relative time (such as next hour), then three macro variables are generated. When the prompt is run, the user selects **Next hour** as the relative time. If the current time is 12:54:36 PM, then the value of the next hour is 01:00:00 PM. ■ The CLOSINGTIME macro variable displays the time in a 24-hour format (13:00:00), and the CLOSINGTIME_LABEL macro variable displays the value that the user selected when the prompt is run. ■ The CLOSINGTIME_REL macro variable is H1H to represent the next hour.

Timestamp Prompts

Here are the extensions for timestamp prompts:

- _LABEL - the relative or formatted value. The format for the timestamp value is *mmm dd, yyyy hh:mm:ss* AM|PM.
- _REL - the value of the relative timestamp. This extension is used only if the user selects a relative timestamp (such as the current date and time) for the prompt.

Suppose that you create a timestamp prompt called ClosingTimestamp. When the prompt is run, the user specifies the closing date and time in the sales period.

Syntax of Macro Variables	Explanation of Value
`%LET CLOSINGTIMESTAMP = 21May2008 14:11:24;%LET CLOSINGTIMESTAMP_LABEL = May 21, 2008 2:11:24 PM;`	If the user selects a specific date and time (such as May 21, 2008 2:11:24 PM), two macro variables are generated. ■ The CLOSINGTIMESTAMP macro variable displays the time in a 24-hour format. ■ The CLOSINGTIMESTAMP_LABEL macro variable displays the time in a 12-hour format and includes the AM or PM label.
`%LET CLOSINGTIMESTAMP = 21May2009 14:11:24;%LET CLOSINGTIMESTAMP_LABEL = Current date and time of next year;%LET CLOSINGTIMESTAMP_REL = T1Y;`	If the user selects a relative time, then three macro variables are generated. When the prompt is run, the user selects **Current date and time next year** as the relative date. If the current date and time is May 21, 2008 2:11:24 PM, then the value of the current date and time in the next calendar year is 21May2009 14:11:24. ■ The CLOSINGTIMESTAMP macro variable displays the relative time in the *ddmmmyyyy hh:mm:ss* format, and the CLOSINGTIMESTAMP_LABEL macro variable displays the value that the user selected when the prompt is run. ■ The CLOSINGTIMESTAMP_REL macro variable is T1Y to represent the same date and time in the next calendar year.

Time Range, Timestamp Range

Here are the extensions for a time range or timestamp range prompt:

- `_MIN` - the lower bound of the range.
- `_MAX` - the upper bound of the range.
- `_MIN_LABEL` - the time value or timestamp value of the lower bound. This value can be a relative value or a specific value that uses as *mmm dd, yyyy hh:mm:ss* AM|PM format.
- `_MAX_LABEL` - the time value or timestamp value of the upper bound. This value can be a relative value or a specific value that uses as *mmm dd, yyyy hh:mm:ss* AM|PM format.
- `_MIN_REL` - the value of the relative date and time for the lower bound. This extension is used only if the user selects a relative date (such as the current date and time) for the prompt.

- _MAX_REL - the value of the relative date and time for the upper bound. This extension is used only if the user selects a relative date (such as the current date and time) for the prompt.

Suppose that you create a timestamp prompt called SalesRange. When the prompt is run, the user specifies the beginning and end dates and time in the sales period.

Syntax of Macro Variables	Explanation of Value
`%LET SALESRANGE_MAX = 31Dec2019 12:00:00;%LET SALESRANGE_MAX_LABEL = December 31, 2019 12:00:00 PM;%LET SALESRANGE_MIN = 01Jan2019 00:00:00;%LET SALESRANGE_MIN_LABEL = January 1, 2019 12:00:00 AM;`	If the user selects specific minimum and maximum dates (such as January 1, 2019 and December 31, 2019), four macro variables are generated. - The SALESRANGE_MAX and SALESRANGE_MIN macro variables display the time in a 24-hour format. - The SALESRANGE_MAX_LABEL and SALESRANGE_MIN_LABEL macro variables display the time in a 12-hour format and include the AM or PM label.
`%LET SALESRANGE_MAX = 21May2019 15:25:26;%LET SALESRANGE_MAX_LABEL = 15 minutes from now;%LET SALESRANGE_MAX_REL = T15m;%LET SALESRANGE_MIN = 21MAY2019 15:10:00;%LET SALESRANGE_MIN_LABEL = Current minute;%LET SALESRANGE_MIN_REL = m0m;`	If the user selects a relative date and time, six macro variables are generated. When the prompt is run, the user selects a timestamp range of the current minute to 15 minutes from now. - The macro variables for the maximum value show the time 15 minutes from now (May 21, 2019 3:25:56 PM). The SALESRANGE_MAX_REL macro variable is T15m to represent 15 minutes from now. - The macro variables for the minimum value show the current time (May 21, 2019 3:10:00 PM).

Color Prompts

There are no extensions for color prompts.

Suppose that you create a color prompt called BarColor. When the prompt is run, the user selects the color of the bars in a bar chart. Colors can be selected from a color palette or by specifying the Red, Green, Blue (RGB) value. When the macro variable is generated, the hexadecimal value for the color is used. The user selects ″Red″ as the color, and the following macro variable is generated: `%LET BARCOLOR=cxff0000;`.

Data Source Prompts

A data source prompt has the `_TYPE` extension. This value specifies the type of data source that you want the user to select from when the prompt is run. Each type of data source has a unique value:

- `1` for tables
- `2` for OLAP cubes
- `4` for relational information maps
- `8` for OLAP information cubes

Suppose that you create a data source prompt called InputData. When the prompt is run, the user specifies the data source to use for the analysis.

Syntax of Macro Variables	Explanation of Value
`%LET INPUTDATA = /Shared Data/sashelp/CLASS;%LET INPUTDATA_TYPE = 1;`	Two macro variables are generated for the type of data source that the user selects when the prompt is run. When the prompt is run, the user selects the CLASS data set. The INPUTDATA macro variable lists the path to this data set, and the INPUTDATA_TYPE macro variable is set to 1 to indicate that the selected data source is a SAS data set.

Data Source Item Prompts

Here are the extensions for data source item prompts:

- `_PATH` - the location of the SAS folder that contains the data source.
- `_SOURCE_TYPE` - the type of data source that you want the user to select from when the prompt is run. Each type of data source has a unique value:
  - `1` for tables
  - `2` for OLAP cubes
  - `4` for relational information maps
  - `8` for OLAP information cubes
- `_TYPE` - the data type of the data source item that you want the user to select from when the prompt is run. Each data type has a unique value:
  - `1` for character values

- □ `2` for numeric values
- □ `4` for date values
- □ `8` for time values
- □ `16` for timestamp values

Suppose that you create the InputData prompt for a data source item. When the prompt is run, the user specifies the column name in the data source or the data item ID in the information map to use in the analysis.

Syntax of Macro Variables	Explanation of Value
`%LET INPUTDATA_SOURCE_TYPE = 1;%LET INPUTDATA = Age;%LET INPUTDATA_PATH = /Shared Data/sashelp/CLASS;%LET INPUTDATA_TYPE = 2;`	Four macro variables are generated for the data source item. When the prompt is run, the user selects the Age column from the CLASS data set. ■ The INPUTDATA macro variable contains the name of the column (Age). ■ The INPUTDATA_PATH macro variable lists the path to the CLASS data set. ■ The INPUTDATA_SOURCE_TYPE macro variable is set to 1 to indicate that the selected data source is a SAS data set. ■ The INPUTDATA_TYPE macro variable is set to 2 to indicate that Age contains numeric data.

File or Directory Prompts

File or directory prompts have the `_SERVER` extension. This value specifies the name of the server that contains the file or directory.

Suppose that you create a file prompt called InputFile. When the prompt is run, the user selects the file to use for the analysis.

Syntax of Macro Variables	Explanation of Value
`%LET INPUTFILE = D:\Public\Customer.sas;%LET INPUTFILE_SERVER = SASMain;`	Two macro variables are generated when the user selects an input file or directory. ■ The INPUTFILE macro variable displays the path of the file or directory. ■ The INPUTFILE_SERVER macro variable lists the SAS Workspace Server that contains the files or directories that the user could select from.

Data Library Prompts

Data library prompts have the _PATH extension. This value specifies the location of the SAS folder where the library is registered.

Suppose that you create a data library prompt called Library. When the prompt is run, the user selects the SAS library that contains the data to use in the analysis.

Suppose that you create a file prompt called InputFile. When the prompt is run, the user selects the file to use for the analysis.

Syntax of Macro Variables	Explanation of Value
`%LET LIBRARY = SASHELP;%LET LIBRARY_PATH = /Shared Data/sashelp/SASHELP;`	Two macro variables are generated when the user selects a SAS library. For example, if the user selects SASHELP as the library, then the LIBRARY macro variable contains the libref for the Sashelp library. The LIBRARY_PATH macro variable displays the path of the library.

Variable Prompts

Variable prompts do not have extensions.

Suppose that you create a variable prompt called Customers. When the prompt is run, the user selects the variable that contains the list of customers for the report. The user selects Europe as the value of the variable prompt, and the following macro variable is generated: `%LET CUSTOMERS = Europe;`

Macro Variables for Special Values

For several prompt types, you can specify whether to include all possible values or missing values in the selection list that appears when the prompt is run. If the user selects one of these special values, then the following macro variables appear in the SAS code:

Special Value	Value in Generated Code	Example
All possible values	`_ALL_VALUES_`	Suppose that you create a text prompt called Products. When the prompt is run, the user selects **(all possible values)**, and the following macro variable is generated: `%LET PRODUCTS = _ALL_VALUES_;`
Missing values	. (period) for numeric prompts (single space) for character prompts	Suppose that you create a numeric prompt called Sales. The values for this prompt are numeric values. When the prompt is run, the user selects **(missing values)**, and the following macro variable is generated: `%LET SALES = .;` Suppose that you create a text prompt called Products. The values for this prompt are character values. When the prompt is run, the user selects **(missing values)**, and the following macro variable is generated: `%LET PRODUCTS = ;`

Creating Prompts

About Creating Prompts

You can create prompts in these ways:

- by using the **Prompt Manager** pane
- when you are working in the Query Builder
- when you are writing a SAS program
- when you are creating a stored process

When you create a prompt, you must first specify the general properties of the prompt, such as the name and description. Then you can select the type of prompt that you want to create and specify the types of values that are valid for the prompt and how users can enter or select those values.

To create a prompt:

1. Specify the general properties of the prompt.
2. Specify the prompt type and its values.
3. Create dependencies between prompts.

Note: You can create a dependency only for prompts in stored processes.

Specify the General Properties of a Prompt

1 On the **General** tab, enter a unique name for the prompt in the **Name** box. Names can contain only alphanumeric characters and underscores, cannot begin with a number, and cannot exceed 32 characters.

Note: SAS might append additional characters to the name (such as numbers for lists or "_REL" for relative dates), so the actual name that you specify might need to be less than 32 characters.

Note: If you are creating a masked text prompt, then the name of the prompt should include _PASSWORD. If you do not include _PASSWORD in the prompt name, then any user-entered values for masked prompts appear in the SAS log.

2 In the **Displayed text** box, enter the text that the user sees when the prompt is run. By default, the displayed text is the same as the prompt name.

3 In the **Description** box, enter the description to display to the user when the prompt is run.

4 If you are using an application or software features that use prompt groups, then you can specify the parent group that contains this prompt in the **Parent group** box. If the prompt is contained by a non-shared group, then you can change the parent group (that is, move the prompt) by selecting a different value from the drop-down list.

5 You can select any of these options:

- **Hide at run time** specifies whether the user is able to view the prompt and its value.

  Note: If you select this option and the **Requires a non-blank value** option, you must specify a default value on the **Prompt Type and Values** tab.

- **Requires a non-blank value** specifies whether the user is required to provide a non-blank value.

  Note: If you select this option and the **Hide at run time** or **Read-only values** option, you must specify a default value on the **Prompt Type and Values** tab. If you select this option, the **Minimum value count** option on the **Prompt Type and Values** tab is automatically set to 1. If you do not select this option, the **Minimum value count** option is cleared.

- **Read-only values** specifies whether the user can change the value of the prompt.

  Note: If you select this option and the **Requires a non-blank value** option, then you must specify a default value on the **Prompt Type and Values** tab.

- **Use prompt value throughout project** - specifies whether the value or values that a user assigns to a prompt can be used by other tasks, queries, and programs in the project. If you do not select this option, then the values are deleted after the code is run.

Selecting the Prompt Type and Its Values

Create a Text Prompt with a Single, User-Entered Line of Text

A text prompt contains character strings. You can create a text prompt where the user specifies a single value for the prompt.

To create a text prompt with a single, user-entered line of text:

1. On the **Prompt Type and Values** tab, select **Text** from the **Prompt type** drop-down list.

2. From the **Method for populating prompt** drop-down list, select **User enters values**. For more information, see “Understanding the Methods for Populating Prompts” on page 423.

3. From the **Number of values** drop-down list, select **Single value**.

4. From the **Text type** drop-down list, select **Single line**.

5. In the **Minimum length** and **Maximum length** boxes, specify the minimum and maximum number of characters that can be entered for the prompt. Trailing blanks are included when the length of a prompt value is calculated and validated.

   - If you do not specify a minimum value, the default value is 1.
   - If you do not specify a maximum value, the default value is 2,147,483,647.

6. Select any special values that can be included. You can choose from these options:

   - **All possible values** - adds **(all possible values)** to the list of values that the user sees when the prompt is run. You usually select this option when you have other prompts that are dependent on this prompt. If the user selects this special value for the prompt when the prompt is run, then the prompt processes all of the values that you defined for it.

- **Missing values** - enables the user to select all of the missing values from the data source. The value **(missing values)** is added to the list of prompt values. For character prompts, the missing value is represented by a blank. For more information, see "Understanding Missing Values" on page 424.

7 In the **Default value** box, specify the default value for the prompt. If you selected any of the options in the **Include Special Values** area, you can select those special values from the drop-down list. If a default value does not appear in the drop-down list, you can enter any character value. Blank spaces and nonprintable characters can be used, but the value cannot consist completely of these characters.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

8 To help users know what type of value to enter for the prompt, enter a hint in the **Hint** box.

Create a Text Prompt with a Masked, User-Entered Single Line of Text

A text prompt contains character strings. You can create a text prompt where the user specifies a single line of masked text for the prompt. You might want to use masked text if you are creating a prompt that stores passwords. When the prompt is run, the user-entered value is masked by asterisks.

To create a text prompt with a masked, user-entered single line of text:

1 On the **Prompt Type and Values** tab, select **Text** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User enters values**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Single value**.

4 From the **Text type** drop-down list, select **Masked single line**.

Note: When you select this option, a message about changes to the prompt name might appear. If the name of the prompt on the **General** tab does not include _PASSWORD, then any user-entered values for masked prompts appear in the SAS log. To automatically append _PASSWORD to the name of the prompt and to hide these user-entered values in the SAS log, click **Yes** in the message dialog box.

5 In the **Minimum length** and **Maximum length** boxes, specify the minimum and maximum number of characters that can be entered for the prompt. Trailing

blanks are included when the length of a prompt value is calculated and validated.

- If you do not specify a minimum value, the default value is 1.
- If you do not specify a maximum value, the default value is 2,147,483,647.

6 From the **Method for encoding masked text** drop-down list, select whether to encode the prompt value. You can choose from these options:

- **SAS002 - SAS proprietary encryption** - specifies to use the Base SAS algorithm that encrypts with 32-bit keys. This method is selected by default.
- **SAS001 - base64 encoding** - specifies to use base64 encoding.
- **(none)** - specifies that no encoding method is used.

Note: To reduce exposure, additional handling of prompts that contain sensitive data values might be needed. For example, a SAS Stored Process server can detect certain strings (like _PASSWORD) that you include in a prompt name. When the server detects such a string, it masks the prompt's value in the server log with XXXXXX. (For more information about hiding passwords and other sensitive data for stored processes, see *SAS Stored Processes: Developer's Guide*.)

Create a Text Prompt with Multiple, User-Entered Lines of Text

A text prompt contains character strings. You can create a text prompt where the user specifies multiple lines of text for the prompt.

To create a text prompt with multiple, user-entered lines of text:

1 On the **Prompt Type and Values** tab, select **Text** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User enters values**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Single value**.

4 From the **Text type** drop-down list, select **Multi-line text**.

5 In the **Maximum line count** box, specify the maximum number of lines of text that the user can enter when the prompt is run. By default, the maximum line count is 2,147,483,647.

6 In the **Minimum length** and **Maximum length** boxes, specify the minimum and maximum number of characters that can be entered for the prompt. Trailing blanks are included when the length of a prompt value is calculated and validated.

- If you do not specify a minimum value, the default value is 1.
- If you do not specify a maximum value, the default value is 2,147,483,647.

7 In the **Number of lines displayed** box, specify the number of lines of text to display to the user when the prompt is run.

8 In the **Default value** box, enter the default value for the prompt. Blank spaces and nonprintable characters can be used, but the value cannot consist completely of these characters.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

Create a Text Prompt with Multiple, User-Entered Values

A text prompt contains character strings. You can create a text prompt where the user specifies multiple values for the prompt.

To create a text prompt with multiple, user-entered values:

1 On the **Prompt Type and Values** tab, select **Text** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User enters values**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Multiple values** or **Multiple ordered values**. If you select **Multiple ordered values**, the user can reorder the selected values when the prompt is run.

4 In the **Minimum value count** and **Maximum value count** boxes, specify the minimum and maximum number of values that are required for the prompt.

- If you do not specify a minimum value count, the default is 0.
- If you do not specify a maximum value count, the default is 2,147,483,647.

Note: If you specify 0 as the minimum value, then the **Requires a non-blank value** check box on the **General** tab is automatically cleared. If you specify a value greater than 0, then the **Requires a non-blank value** check box is automatically selected.

5 In the **Minimum length** and **Maximum length** boxes, specify the minimum and maximum number of characters that can be entered for the prompt. Trailing blanks are included when the length of a prompt value is calculated and validated.

- If you do not specify a minimum value, the default value is 1.
- If you do not specify a maximum value, the default value is 2,147,483,647.

6 Select any special values that can be included. You can choose from these options:

- **All possible values** - adds **(all possible values)** to the list of values that the user sees when the prompt is run. You usually select this option when you have other prompts that are dependent on this prompt. If the user selects this special value for the prompt when the prompt is run, then the prompt processes all of the values that you defined for it.
- **Missing values** - enables the user to select all of the missing values from the data source. The value **(missing values)** is added to the list of prompt values. For character prompts, the missing value is represented by a blank. For more information, see “Understanding Missing Values” on page 424.

7 To create the list of default values for the prompt, click **Add** and enter the value. If you selected any of the options in the **Include Special Values** area, you can select those special values from a drop-down list in the **Default Values** box. Repeat step 7 or press Enter to add additional values to the list. The order that the values appear in the **Default values** list is how they appear to the user when the prompt is run. The default values are used if the user does not specify a value when prompted.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

8 To help users know what type of value to enter for the prompt, enter a hint in the **Hint** box.

Create a Text Prompt with a Single Value from a Static List

A text prompt contains characters. You can create a text prompt where the user selects a single value from a static list.

To create a text prompt with a single value from a static list:

1 On the **Prompt Type and Values** tab, select **Text** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User selects values from static list**. For more information, see “Understanding the Methods for Populating Prompts” on page 423.

3 From the **Number of values** drop-down list, select **Single value**.

4 In the **Minimum length** and **Maximum length** boxes, specify the minimum and maximum number of characters that can be entered for the prompt. Trailing

blanks are included when the length of a prompt value is calculated and validated.

- If you do not specify a minimum value, the default value is 1.
- If you do not specify a maximum value, the default value is 2,147,483,647.

5 Select any special values that can be included. You can choose from these options:

- **All possible values** - adds **(all possible values)** to the list of values that the user sees when the prompt is run. You usually select this option when you have other prompts that are dependent on this prompt. If the user selects this special value for the prompt when the prompt is run, then the prompt processes all of the values that you defined for it.
- **Missing values** - enables the user to select all of the missing values from the data source. The value **(missing values)** is added to the list of prompt values. For character prompts, the missing value is represented by a blank. For more information, see "Understanding Missing Values" on page 424.

6 To display each unformatted value in square brackets ([]) after the formatted value when the prompt runs, select the **Append formatted values with unformatted values** check box. SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended formatted values also appear in the **List of values** box. For more information, see "Examples of Appended Formatted Values" on page 431.

7 To create a static list for the prompt:

a Click **Add**, and enter the value that you want to use in the **Unformatted Value** column.

b To specify a different formatted value that is displayed to the user in the list, enter the formatted value in the **Formatted (Displayed) Value** column.

c To specify the value as the default value, select **Default**.

d Repeat steps 1–3 to add additional values to the list. Press Enter to add a new row. The order that the values appear in the **List of values** list is how they appear to the user when the prompt is run.

Note: If you selected any of the options in the **Include Special Values** area, those special values also appear in the **List of values** box.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

8 To allow the user to add additional unformatted values to the list of values in the prompt, select the **Allow user to specify additional (unformatted) values** check box.

Create a Text Prompt with Multiple Values from a Static List

A text prompt contains character strings. You can create a text prompt where the user specifies multiple lines of text for the prompt.

To create a text prompt with multiple values from a static list:

1 On the **Prompt Type and Values** tab, select **Text** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User selects values from static list**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Multiple values** or **Multiple ordered values**. If you select **Multiple ordered values**, the user can reorder the selected values when the prompt is run.

4 In the **Minimum value count** and **Maximum value count** boxes, specify the minimum and maximum number of values that are required for the prompt.

- If you do not specify a minimum value count, the default is 0.
- If you do not specify a maximum value count, the default is 2,147,483,647.

Note: If you specify 0 as the minimum value, then the **Requires a non-blank value** check box on the **General** tab is automatically cleared. If you specify a value greater than 0, then the **Requires a non-blank value** check box is automatically selected.

5 In the **Minimum length** and **Maximum length** boxes, specify the minimum and maximum number of characters that can be entered for the prompt. Trailing blanks are included when the length of a prompt value is calculated and validated.

- If you do not specify a minimum value, the default value is 1.
- If you do not specify a maximum value, the default value is 2,147,483,647.

6 Select any special values that can be included. You can choose from these options:

- **All possible values** - adds **(all possible values)** to the list of values that the user sees when the prompt is run. You usually select this option when you have other prompts that are dependent on this prompt. If the user selects this special value for the prompt when the prompt is run, then the prompt processes all of the values that you defined for it.
- **Missing values** - enables the user to select all of the missing values from the data source. The value **(missing values)** is added to the list of prompt values. For character prompts, the missing value is represented by a blank. For more information, see "Understanding Missing Values" on page 424.

7 To display each unformatted value in square brackets ([]) after the formatted value when the prompt runs, select the **Append formatted values with unformatted values** check box. SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended formatted values also appear in the **List of values** box. For more information, see "Examples of Appended Formatted Values" on page 431.

8 To create the list of values for the prompt:

a Click **Add**, and enter the value that you want to use in the **Unformatted value** column.

b To specify a different formatted value that is displayed to the user in the list, enter the formatted value in the **Formatted (Displayed) Value** column.

c To specify the value as the default value, select **Default**.

d Repeat steps 1–3 to add additional values to the list. Press Enter to add a new row. The order that the values appear in the **List of values** list is how they appear to the user when the prompt is run.

Note: If you selected any of the options in the **Include Special Values** area, those special values also appear in the **List of values** box.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

9 To allow the user to add additional unformatted values to the list of values in the prompt, select the **Allow user to specify additional (unformatted) values** check box.

Create a Text Prompt with a Single Value from a Dynamic List

A text prompt contains character strings. You can create a text prompt for which the user selects either single or multiple values from a list of unique values that is retrieved from the data source that you specify when you create the prompt. The list of values is populated when the prompt is run.

To create a text prompt with a single value from a dynamic list:

1 On the **Prompt Type and Values** tab, select **Text** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User selects values from a dynamic list**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Single value**.

4 In the **Minimum length** and **Maximum length** boxes, specify the minimum and maximum number of characters that can be entered for the prompt. Trailing blanks are included when the length of a prompt value is calculated and validated.

- If you do not specify a minimum value, the default value is 1.
- If you do not specify a maximum value, the default value is 2,147,483,647.

5 Select the data source from which the list of values is populated.

Note: Do not select a SAS Information Map table as the data source for a dynamic list. SAS Information Map tables are tables that are in SAS Information Map libraries. Prompts cannot retrieve values from this type of table.

6 In the **Unformatted Values** box, select the column that contains the unformatted values for the list. Only columns that contain valid values for the selected prompt type are available.

By default, only distinct values are displayed. To show all the unformatted values, clear the **Show only distinct values** check box.

7 In the **Formatted (Displayed) Values** box, select the column that contains the formatted values. To use the same column for the formatted and unformatted values, select the **Use 'Unformatted Values' column** option. If you select a different column for the formatted values, select the format of the column. By default, the format from the data source is used.

8 To display each unformatted value in square brackets ([]) after the formatted value when the prompt runs, select the **Append formatted values with unformatted values** check box. SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended formatted values also appear in the **List of values** box. For more information, see "Examples of Appended Formatted Values" on page 431.

9 To include a special value that enables the user to select all of the prompt values at one time, select the **All possible values** check box. The value **(all possible values)** is added to the list of prompt values.

10 From the **Sort order** drop-down list, select the sort order for the values in the list. By default, the values are sorted in the order which they appear in the data source. You can also sort the formatted or unformatted values in ascending or descending order.

11 From the **Default values** drop-down list, select the default values for the prompt. If you selected the **All possible values** check box, you can select those values from the drop-down list. If no values appear in the drop-down list, click **Select**.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

12 To allow the user to add additional unformatted values to the list of values in the prompt, select the **Allow user to specify additional (unformatted) values** check box.

Create a Text Prompt with Multiple Values from a Dynamic List

A text prompt contains character strings. You can create a text prompt for which the user selects either single or multiple values from a list of unique values that is retrieved from the data source that you specify when you create the prompt. The list of values is populated when the prompt is run.

To create a text prompt with multiple values from a dynamic list:

1 On the **Prompt Type and Values** tab, select **Text** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User selects values from a dynamic list**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Multiple values** or **Multiple ordered values**. If you select **Multiple ordered values**, the user can reorder the selected values when the prompt is run.

4 In the **Minimum value count** and **Maximum value count** boxes, specify the minimum and maximum number of values that are required for the prompt.

- If you do not specify a minimum value count, the default is 0.
- If you do not specify a maximum value count, the default is 2,147,483,647.

Note: If you specify 0 as the minimum value, then the **Requires a non-blank value** check box on the **General** tab is automatically cleared. If you specify a value greater than 0, then the **Requires a non-blank value** check box is automatically selected.

5 In the **Minimum length** and **Maximum length** boxes, specify the minimum and maximum number of characters that can be entered for the prompt. Trailing blanks are included when the length of a prompt value is calculated and validated.

- If you do not specify a minimum value, the default value is 1.
- If you do not specify a maximum value, the default value is 2,147,483,647.

6 Select the data source from which the list of values is populated.

Note: Do not select a SAS Information Map table as the data source for a dynamic list. SAS Information Map tables are tables that are in SAS Information Map libraries. Prompts cannot retrieve values from this type of table.

7 In the **Unformatted Values** box, select the column that contains the unformatted values for the list. Only columns that contain valid values for the selected prompt type are available.

By default, only distinct values are displayed. To show all the unformatted values, clear the **Show only distinct values** check box.

8 In the **Formatted (Displayed) Values** box, select the column that contains the formatted values. To use the same column for the formatted and unformatted values, select the **Use 'Unformatted Values' column** option. If you select a different column for the formatted values, select the format of the column. By default, the format from the data source is used.

9 To display each unformatted value in square brackets ([]) after the formatted value when the prompt runs, select the **Append formatted values with unformatted values** check box. SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended formatted values also appear in the **List of values** box. For more information, see "Examples of Appended Formatted Values" on page 431.

10 To include a special value that enables the user to select all of the prompt values at one time, select the **All possible values** check box. The value **(all possible values)** is added to the list of prompt values.

11 From the **Default values** drop-down list, select the default values for the prompt. If you selected the **All possible values** check box, you can select those values from the drop-down list. If no values appear in the drop-down list, click **Select**.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

12 To allow the user to add additional unformatted values to the list of values in the prompt, select the **Allow user to specify additional (unformatted) values** check box.

Create a Text Range Prompt

A text range prompt contains two character strings that are the upper and lower boundaries for a range of character values.

To create a text range prompt:

1 On the **Prompt Type and Values** tab, select **Text range** from the **Prompt type** drop-down list.

2 In the **Minimum length** and **Maximum length** boxes, specify the minimum and maximum lengths for the text string that the user provides when the prompt is run.

3 In the **Minimum value allowed** and **Maximum value allowe**d boxes, specify the minimum and maximum values that are allowed for the prompt.

The prompt uses the Unicode Collation Algorithm to determine the following:

- whether a prompt value is greater than the minimum value allowed. For example, with this algorithm, the value "a" is less than the value "z." If the specified prompt value and the minimum value allowed are identical except that one has extra characters, then the shorter string is considered to be the lesser value. For example, the value "ab" is less than the value "abc."
- whether a prompt value is less than the maximum value allowed. For example, with this algorithm, the value "z" is greater than the value "a." If the specified prompt value and the maximum value allowed are identical except that one has extra characters, then the longer string is considered to be the greater value. For example, the value "abc" is greater than the value "ab."

Note: If you do not enter a minimum or maximum value, then the default value is null, and the user can enter any text string for the prompt.

4 In the **From** and **To** boxes, specify default values for the lower and upper boundaries of the range.

Note: You can include spaces and nonprintable characters, but a default value cannot consist completely of these characters. Trailing blanks are stored as part of the value and are included when SAS Enterprise Guide validates the value against the minimum and maximum length requirements.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

Create a Hyperlink Prompt

A hyperlink prompt contains the web address for a link.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

To create a hyperlink prompt:

1 On the **Prompt Type and Values** tab, select **Hyperlink** from the **Prompt type** drop-down list.

2 In the **Default link text** box, enter the text for the link.

3 In the **Default link address (URL)** box, enter the web address for the link.

Create a Numeric Prompt with a Single, User-Entered Value

A numeric prompt accepts only numeric data as input. You can create a numeric prompt for which the user enters either single or multiple values when the prompt is run.

To create a numeric prompt with a single, user-entered value:

1 On the **Prompt Type and Values** tab, select **Numeric** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User enters values**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Single value**.

4 To specify that only integer values can be entered by the user, select **Allow only integer values**.

5 In the **Minimum number of decimal places displayed** box, enter the minimum number of decimal places that you want to display. For example, if you enter a value of 2, then the prompt values of 1.11, 2, 3.30, and 4.4 would be displayed to the user as 1.11, 2.00, 3.30, and 4.40. You can enter a value between 0 and 15. The default value is 0.

> **Note:** If you specify that only integer values can be included in the list, this option is disabled.

6 In the **Maximum number of decimal places allowed** box, enter the maximum number of decimal places that you want to allow. For example, if you enter a value of 4, then the following prompt values would be valid: 0, 1.1, 2.22, 3.333, 4.4444. However, 5.55555 would not be a valid prompt value. Negative values are not allowed. If you do not specify a value, then the default value is 2,147,483,647.

> **Note:** If you specify that only integer values can be included in the list, this option is disabled.

7 In the **Minimum value allowed** and **Maximum value allowed** boxes, enter the minimum and maximum values that can be entered by the user. For more information, see "Understanding Numeric Values" on page 424.

- If you do not specify a minimum value, then the default value is –1.7976931348623157E308.
- If you do not specify a maximum value, then the default value is 1.7976931348623157E308.

8 Select any special values that you want to include in the list of prompt values. You can choose from these options:

- **All possible values** - enables the user to select all of the prompt values at one time. The value **(all possible values)** is added to the list of prompt values.
- **Missing values** - enables the user to select all of the missing values from the data source. The value (missing values) is added to the list of prompt values. For numeric prompts, the missing value is represented by a period (.). For more information, see "Understanding Missing Values" on page 424.

Note: You can create a special numeric missing value for numeric prompts. The unformatted value that you specify must start with a period and be followed by an underscore or a letter of the alphabet (._ or .A through .Z). The letters can be uppercase or lowercase.

9 Enter the default value for the prompt in the **Default Value** box. If you selected any of the options in the **Include Special Values** area, you can select those special values from a drop-down list in the **Default Value** box. For more information, see "Understanding Numeric Values" on page 424. The default value is used if the user does not specify a value when prompted.

Create a Numeric Prompt with Multiple, User-Entered Values

A numeric prompt accepts only numeric data as input. You can create a numeric prompt for which the user enters either single or multiple values when the prompt is run.

To create a numeric prompt with multiple, user-entered values:

1 On the **Prompt Type and Values** tab, select **Numeric** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User enters values**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Multiple values** or **Multiple ordered values**. If you select **Multiple ordered values**, the user can reorder the selected values when the prompt is run.

4 To specify that only integer values can be entered by the user, select **Allow only integer values**.

5 In the **Minimum number of decimal places displayed** box, enter the minimum number of decimal places that you want to display. For example, if you enter a value of 2, then the prompt values of 1.11, 2, 3.30, and 4.4 would be displayed to the user as 1.11, 2.00, 3.30, and 4.40. You can enter a value between 0 and 15. The default value is 0.

Note: If you specify that only integer values can be included in the list, this option is disabled.

6. In the **Maximum number of decimal places allowed** box, enter the maximum number of decimal places that you want to allow. For example, if you enter a value of 4, then the following prompt values would be valid: 0, 1.1, 2.22, 3.333, 4.4444. However, 5.55555 would not be a valid prompt value. Negative values are not allowed. If you do not specify a value, then the default value is 2,147,483,647.

Note: If you specify that only integer values can be included in the list, this option is disabled.

7. In the **Minimum value count** and **Maximum value count** boxes, specify the minimum and maximum number of values that are required for the prompt.

   - If you do not specify a minimum value count, the default is 0.
   - If you do not specify a maximum value count, the default is 2,147,483,647.

Note: If you specify 0 as the minimum value, then the **Requires a non-blank value** check box on the **General** tab is automatically cleared. If you specify a value greater than 0, then the **Requires a non-blank value** check box is automatically selected.

8. In the **Minimum value allowed** and **Maximum value allowed** boxes, enter the minimum and maximum values that can be entered by the user. For more information, see “Understanding Numeric Values” on page 424.

   - If you do not specify a minimum value, then the default value is –1.7976931348623157E308.
   - If you do not specify a maximum value, then the default value is 1.7976931348623157E308.

9. Select any special values that you want to include in the list of prompt values. You can choose from these options:

   - **All possible values** - enables the user to select all of the prompt values at one time. The value **(all possible values)** is added to the list of prompt values.
   - **Missing values** - enables the user to select all of the missing values from the data source. The value (missing values) is added to the list of prompt values. For numeric prompts, the missing value is represented by a period (.). For more information, see “Understanding Missing Values” on page 424.

Note: You can create a special numeric missing value for numeric prompts. The unformatted value that you specify must start with a period and be followed by an underscore or a letter of the alphabet (._ or .A through .Z). The letters can be uppercase or lowercase.

10 To create the list of default values for the prompt, click **Add** and enter the value. If you selected any of the options in the **Include Special Values** area, you can select those special values from a drop-down list in the **Default Values** box. For more information, see "Understanding Numeric Values" on page 424.

To add additional values to the list, repeat this step or press Enter. The order that the values appear in the **Default Values** list is how they appear to the user when the prompt is run. The default values are used if the user does not specify a value when prompted.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

Create a Numeric Prompt with a Single Value from a Static List

A numeric prompt accepts only numeric data as input. You can create a numeric prompt for which the user selects either single or multiple values from a fixed set of values that you define when you create the prompt.

To create numeric prompt with a single value from a static list:

1 On the **Prompt Type and Values** tab, select **Numeric** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User selects values from static list**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Single value**.

4 To specify that only integer values can be entered by the user, select **Allow only integer values**.

5 In the **Minimum number of decimal places displayed** box, enter the minimum number of decimal places that you want to display. For example, if you enter a value of 2, then the prompt values of 1.11, 2, 3.30, and 4.4 would be displayed to the user as 1.11, 2.00, 3.30, and 4.40. You can enter a value between 0 and 15. The default value is 0.

Note: If you specify that only integer values can be included in the list, this option is disabled.

6 In the **Maximum number of decimal places allowed** box, enter the maximum number of decimal places that you want to allow. For example, if you enter a value of 4, then the following prompt values would be valid: 0, 1.1, 2.22, 3.333, 4.4444. However, 5.55555 would not be a valid prompt value. Negative values

are not allowed. If you do not specify a value, then the default value is 2,147,483,647.

Note: If you specify that only integer values can be included in the list, this option is disabled.

7 In the **Minimum value allowed** and **Maximum value allowed** boxes, enter the minimum and maximum values that can be entered by the user. For more information, see "Understanding Numeric Values" on page 424.

- If you do not specify a minimum value, then the default value is –1.7976931348623157E308.
- If you do not specify a maximum value, then the default value is 1.7976931348623157E308.

8 Select any special values that you want to include in the list of prompt values. You can choose from these options:

- **All possible values** - enables the user to select all of the prompt values at one time. The value **(all possible values)** is added to the list of prompt values.
- **Missing values** - enables the user to select all of the missing values from the data source. The value (missing values) is added to the list of prompt values. For numeric prompts, the missing value is represented by a period (.). For more information, see "Understanding Missing Values" on page 424.

Note: You can create a special numeric missing value for numeric prompts. The unformatted value that you specify must start with a period and be followed by an underscore or a letter of the alphabet (._ or .A through .Z). The letters can be uppercase or lowercase.

9 To display each unformatted value in square brackets ([]) after the formatted value when the prompt runs, select the **Append formatted values with unformatted values** check box. SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended formatted values also appear in the **List of values** box. For more information, see "Examples of Appended Formatted Values" on page 431.

10 To create the list of values for the prompt:

a Click **Add**, and enter the value that you want to use in the **Unformatted value** column.

b To specify a different formatted value that is displayed to the user in the list, enter the formatted value in the **Formatted (Displayed) Value** column.

c To specify the value as the default value, select **Default**.

d Repeat steps 1–3 to add additional values to the list. Press Enter to add a new row. The order that the values appear in the **List of values** list is how they appear to the user when the prompt is run.

Note: If you selected any of the options in the **Include Special Values** area, those special values also appear in the **List of values** box.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

Create a Numeric Prompt with Multiple Values from a Static List

A numeric prompt accepts only numeric data as input. You can create a numeric prompt for which the user selects either single or multiple values from a fixed set of values that you define when you create the prompt.

To create a numeric prompt with multiple values from a static list:

1 On the **Prompt Type and Values** tab, select **Numeric** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User selects values from static list**. For more information, see “Understanding the Methods for Populating Prompts” on page 423.

3 From the **Number of values** drop-down list, select **Multiple values** or **Multiple ordered values**. If you select **Multiple ordered values**, the user can reorder the selected values when the prompt is run.

4 To specify that only integer values can be entered by the user, select **Allow only integer values**.

5 In the **Minimum number of decimal places displayed** box, enter the minimum number of decimal places that you want to display. For example, if you enter a value of 2, then the prompt values of 1.11, 2, 3.30, and 4.4 would be displayed to the user as 1.11, 2.00, 3.30, and 4.40. You can enter a value between 0 and 15. The default value is 0.

Note: If you specify that only integer values can be included in the list, this option is disabled.

6 In the **Maximum number of decimal places allowed** box, enter the maximum number of decimal places that you want to allow. For example, if you enter a value of 4, then the following prompt values would be valid: 0, 1.1, 2.22, 3.333, 4.4444. However, 5.55555 would not be a valid prompt value. Negative values are not allowed. If you do not specify a value, then the default value is 2,147,483,647.

Note: If you specify that only integer values can be included in the list, this option is disabled.

7 In the **Minimum value allowed** and **Maximum value allowed** boxes, enter the minimum and maximum values that can be entered by the user. For more information, see "Understanding Numeric Values" on page 424.

- If you do not specify a minimum value, then the default value is –1.7976931348623157E308.
- If you do not specify a maximum value, then the default value is 1.7976931348623157E308.

8 Select any special values that you want to include in the list of prompt values. You can choose from these options:

- **All possible values** - enables the user to select all of the prompt values at one time. The value **(all possible values)** is added to the list of prompt values.
- **Missing values** - enables the user to select all of the missing values from the data source. The value (missing values) is added to the list of prompt values. For numeric prompts, the missing value is represented by a period (.). For more information, see "Understanding Missing Values" on page 424.

Note: You can create a special numeric missing value for numeric prompts. The unformatted value that you specify must start with a period and be followed by an underscore or a letter of the alphabet (._ or .A through .Z). The letters can be uppercase or lowercase.

9 To display each unformatted value in square brackets ([]) after the formatted value when the prompt runs, select the **Append formatted values with unformatted values** check box. SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended formatted values also appear in the **List of values** box. For more information, see "Examples of Appended Formatted Values" on page 431.

10 To create the list of values for the prompt:

a Click **Add**, and enter the value that you want to use in the **Unformatted value** column.

b To specify a different formatted value that is displayed to the user in the list, enter the formatted value in the **Formatted (Displayed) Value** column.

c To specify the value as the default value, select **Default**.

d Repeat steps 1–3 to add additional values to the list. Press Enter to add a new row. The order that the values appear in the **List of values** list is how they appear to the user when the prompt is run.

Note: If you selected any of the options in the **Include Special Values** area, those special values also appear in the **List of values** box.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

11 To allow the user to add additional unformatted values to the list of values in the prompt, select the **Allow user to specify additional (unformatted) values** check box.

Create a Numeric Prompt with a Single Value from a Dynamic List

A numeric prompt accepts only numeric data as input. You can create a numeric prompt for which the user selects either single or multiple values from a list of unique values that is retrieved from the data source that you specify when you create the prompt. The list of values is populated when the prompt is run.

To create a numeric prompt with a single value from a dynamic list:

1 On the **Prompt Type and Values** tab, select **Numeric** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User selects values from a dynamic list**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Single value**.

4 To specify that only integer values can be entered by the user, select **Allow only integer values**.

5 In the **Minimum number of decimal places displayed** box, enter the minimum number of decimal places that you want to display. For example, if you enter a value of 2, then the prompt values of 1.11, 2, 3.30, and 4.4 would be displayed to the user as 1.11, 2.00, 3.30, and 4.40. You can enter a value between 0 and 15. The default value is 0.

Note: If you specify that only integer values can be included in the list, this option is disabled.

6 In the **Minimum value allowed** and **Maximum value allowed** boxes, enter the minimum and maximum values that can be entered by the user. For more information, see "Understanding Numeric Values" on page 424.

- If you do not specify a minimum value, then the default value is –1.7976931348623157E308.
- If you do not specify a maximum value, then the default value is 1.7976931348623157E308.

7 Select the data source from which the list of values is populated.

Note: Do not select a SAS Information Map table as the data source for a dynamic list. SAS Information Map tables are tables that are in SAS Information Map libraries. Prompts cannot retrieve values from this type of table.

8 From the **Unformatted Values** drop-down list, select the column that contains the unformatted values for the list. Only columns that contain valid values for the selected prompt type are available.

9 From the **Formatted (Displayed) Values** drop-down list, select the column that contains the formatted values. To use the same column for the formatted and unformatted values, select the **Use 'Unformatted Values' column** option. To select a different column for the formatted values, click **Select**. By default, the format from the data source is used.

10 To display each unformatted value in square brackets ([]) after the formatted value when the prompt runs, select the **Append formatted values with unformatted values** check box. SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended formatted values also appear in the **List of values** box. For more information, see "Examples of Appended Formatted Values" on page 431.

11 To include a special value that enables the user to select all of the prompt values at one time, select the **All possible values** check box. The value **(all possible values)** is added to the list of prompt values.

12 From the **Sort order** drop-down list, select the sort order for the values in the list. By default, the values are sorted in the order which they appear in the data source. You can also sort the formatted or unformatted values in ascending or descending order.

13 From the **Default values** drop-down list, select the default values for the prompt. If you selected the **All possible values** check box, you can select those values from the drop-down list. If no values appear in the drop-down list, click **Select**.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

14 To allow the user to add additional unformatted values to the list of values in the prompt, select the **Allow user to specify additional (unformatted) values** check box.

Create a Numeric Prompt with Multiple Values from a Dynamic List

A numeric prompt accepts only numeric data as input. You can create a numeric prompt for which the user selects either single or multiple values from a dynamic list that is populated by a data source that you specify when you create the prompt. The list of values is populated when the prompt is run.

To create a numeric prompt with multiple values from a dynamic list:

1 On the **Prompt Type and Values** tab, select **Numeric** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User selects values from a dynamic list**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Multiple values** or **Multiple ordered values**. If you select **Multiple ordered values**, the user can reorder the selected values when the prompt is run.

4 To specify that only integer values can be entered by the user, select **Allow only integer values**.

5 In the **Minimum number of decimal places displayed** box, enter the minimum number of decimal places that you want to display. For example, if you enter a value of 2, then the prompt values of 1.11, 2, 3.30, and 4.4 would be displayed to the user as 1.11, 2.00, 3.30, and 4.40. You can enter a value between 0 and 15. The default value is 0.

Note: If you specify that only integer values can be included in the list, this option is disabled.

6 In the **Maximum number of decimal places allowed** box, enter the maximum number of decimal places that you want to allow. For example, if you enter a value of 4, then the following prompt values would be valid: 0, 1.1, 2.22, 3.333, 4.4444. However, 5.55555 would not be a valid prompt value. Negative values are not allowed. If you do not specify a value, then the default value is 2,147,483,647.

Note: If you specify that only integer values can be included in the list, this option is disabled.

7 In the **Minimum value count** and **Maximum value count** boxes, specify the minimum and maximum number of values that are required for the prompt.

- If you do not specify a minimum value count, the default is 0.
- If you do not specify a maximum value count, the default is 2,147,483,647.

Note: If you specify 0 as the minimum value, then the **Requires a non-blank value** check box on the **General** tab is automatically cleared. If you specify a value greater than 0, then the **Requires a non-blank value** check box is automatically selected.

8 In the **Minimum value allowed** and **Maximum value allowed** boxes, enter the minimum and maximum values that can be entered by the user. For more information, see "Understanding Numeric Values" on page 424.

- If you do not specify a minimum value, then the default value is –1.7976931348623157E308.
- If you do not specify a maximum value, then the default value is 1.7976931348623157E308.

9 Select the data source from which the list of values is populated.

Note: Do not select a SAS Information Map table as the data source for a dynamic list. SAS Information Map tables are tables that are in SAS Information Map libraries. Prompts cannot retrieve values from this type of table.

10 From the **Unformatted Values** drop-down list, select the column that contains the unformatted values for the list. Only columns that contain valid values for the selected prompt type are available.

11 From the **Formatted (Displayed) Values** drop-down list, select the column that contains the formatted values. To use the same column for the formatted and unformatted values, select the **Use 'Unformatted Values' column** option. To select a different column for the formatted values, click **Select**. By default, the format from the data source is used.

12 To display each unformatted value in square brackets ([]) after the formatted value when the prompt runs, select the **Append formatted values with unformatted values** check box. SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended formatted values also appear in the **List of values** box. For more information, see "Examples of Appended Formatted Values" on page 431.

13 To include a special value that enables the user to select all of the prompt values at one time, select the **All possible values** check box. The value **(all possible values)** is added to the list of prompt values.

14 From the **Sort order** drop-down list, select the sort order for the values in the list. By default, the values are sorted in the order which they appear in the data source. You can also sort the formatted or unformatted values in ascending or descending order.

15 From the **Default values** drop-down list, select the default values for the prompt. If you selected the **All possible values** check box, you can select those values from the drop-down list. If no values appear in the drop-down list, click **Select**.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

16 To allow the user to add additional unformatted values to the list of values in the prompt, select the **Allow user to specify additional (unformatted) values** check box.

Create a Numeric Range Prompt

A numeric range prompt contains two numeric values that are the upper and lower boundaries for a range of numeric values.

To create a numeric range prompt:

1 On the **Prompt Type and Values** tab, select **Numeric range** from the **Prompt type** drop-down list.

2 To specify that only integer values can be entered by the user, select **Allow only integer values**.

3 In the **Minimum number of decimal places displayed** box, enter the minimum number of decimal places that you want to display. For example, if you enter a value of 2, then the prompt values of 1.11, 2, 3.30, and 4.4 would be displayed to the user as 1.11, 2.00, 3.30, and 4.40. You can enter a value between 0 and 15. The default value is 0.

   Note: If you specify that only integer values can be included in the list, this option is disabled.

4 In the **Maximum number of decimal places allowed** box, enter the maximum number of decimal places that you want to allow. For example, if you enter a value of 4, then the following prompt values would be valid: 0, 1.1, 2.22, 3.333, 4.4444. However, 5.55555 would not be a valid prompt value. Negative values are not allowed. If you do not specify a value, then the default value is 2,147,483,647.

   Note: If you specify that only integer values can be included in the list, this option is disabled.

5 In the **Minimum value allowed** and **Maximum value allowed** boxes, enter the minimum and maximum values that can be entered by the user. For more information, see “Understanding Numeric Values” on page 424.

   - If you do not specify a minimum value, then the default value is –1.7976931348623157E308.
   - If you do not specify a maximum value, then the default value is 1.7976931348623157E308.

6 In the **From** and **To** boxes, specify default values for the lower and upper boundaries of the range.

   Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

Create a Date Prompt with a Single, User-Entered Value

A date prompt is a prompt that contains dates. You can create a date prompt where the user specifies a single value for the prompt. When specifying a date value, you can enter a specific date or select a relative date. For more information, see “Understanding Date Values” on page 424.

To create a date prompt with a single, user-entered value:

1. On the **Prompt Type and Values** tab, select **Date** from the **Prompt type** drop-down list.

2. From the **Method for populating prompt** drop-down list, select **User enters values**. For more information, see “Understanding the Methods for Populating Prompts” on page 423.

3. From the **Number of values** drop-down list, select **Single value**.

4. From the **Date type** drop-down list, select the type of date. You can choose from these options:

   - **Day**
   - **Week**
   - **Month**
   - **Quarter**
   - **Year**

5. In the **Minimum value allowed** and **Maximum value allowed** boxes, specify the earliest and latest dates that can be entered for the prompt. For more information, see “Specify a Date” on page 425.

   - If you do not specify a minimum value, the default value is January 1, 1600.
   - If you do not specify a maximum value, the default value is December 31, 2400.

6. Select any special values that you want to include in the list of prompt values. You can choose from these options:

   - **All possible values** - enables the user to select all of the prompt values at one time. The value **(all possible values)** is added to the list of prompt values.
   - **Missing values** - enables the user to select all of the missing values from the data source. The value (missing values) is added to the list of prompt values. For numeric prompts, the missing value is represented by a period (.). For more information, see “Understanding Missing Values” on page 424.

   Note: You can create a special numeric missing value for numeric prompts. The unformatted value that you specify must start with a period and be followed by an underscore or a letter of the alphabet (._ or .A through .Z). The letters can be uppercase or lowercase.

7 In the **Default value** box, specify the default value for the prompt. If you selected any of the options in the **Include Special Values** area, you can select those special values from the drop-down list. For more information, see “Specify a Date” on page 425.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

Create a Date Prompt with Multiple, User-Entered Values

A date prompt is a prompt that contains dates. You can create a date prompt where the user specifies multiple values for the prompt. When specifying a date value, you can enter a specific date or select a relative date. For more information, see “Understanding Date Values” on page 424.

1 On the **Prompt Type and Values** tab, select **Date** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User enters values**. For more information, see “Understanding the Methods for Populating Prompts” on page 423.

3 From the **Number of values** drop-down list, select **Multiple values** or **Multiple ordered values**. If you select **Multiple ordered values**, the user can reorder the selected values when the prompt is run.

4 From the **Date type** drop-down list, select the type of date. You can choose from these options:

- **Day**
- **Week**
- **Month**
- **Quarter**
- **Year**

5 In the **Minimum value count** and **Maximum value count** boxes, specify the minimum and maximum number of values that are required for the prompt.

- If you do not specify a minimum value count, the default is 0.
- If you do not specify a maximum value count, the default is 2,147,483,647.

Note: If you specify 0 as the minimum value, then the **Requires a non-blank value** check box on the **General** tab is automatically cleared. If you specify a value greater than 0, then the **Requires a non-blank value** check box is automatically selected.

6 In the **Minimum value allowed** and **Maximum value allowed** boxes, specify the earliest and latest dates that can be entered for the prompt. For more information, see "Specify a Date" on page 425.

- If you do not specify a minimum value, the default value is January 1, 1600.
- If you do not specify a maximum value, the default value is December 31, 2400.

7 Select any special values that you want to include in the list of prompt values. You can choose from these options:

- **All possible values** - enables the user to select all of the prompt values at one time. The value **(all possible values)** is added to the list of prompt values.
- **Missing values** - enables the user to select all of the missing values from the data source. The value (missing values) is added to the list of prompt values. For numeric prompts, the missing value is represented by a period (.). For more information, see "Understanding Missing Values" on page 424.

Note: You can create a special numeric missing value for numeric prompts. The unformatted value that you specify must start with a period and be followed by an underscore or a letter of the alphabet (._ or .A through .Z). The letters can be uppercase or lowercase.

8 To create the list of default values for the prompt, click **Add** and enter the value. If you selected any of the options in the **Include Special Values** area, you can select those special values from a drop-down list in the **Default Values** box. Repeat step 7 or press Enter to add additional values to the list. The order that the values appear in the **Default values** list is how they appear to the user when the prompt is run. The default values are used if the user does not specify a value when prompted.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

Create a Date Prompt with a Single Value from a Static List

A date prompt is a prompt that contains dates. You can create a date prompt where the user selects a single value from a static list. When creating a date prompt, you can enter a specific date or select a relative date. For more information, see "Specify a Date" on page 425.

1 On the **Prompt Type and Values** tab, select **Date** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User selects values from static list**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Single value**.

4 In the **Minimum value allowed** and **Maximum value allowed** boxes, specify the earliest and latest dates that can be entered for the prompt. For more information, see "Specify a Date" on page 425.

   - If you do not specify a minimum value, the default value is January 1, 1600.
   - If you do not specify a maximum value, the default value is December 31, 2400.

5 Select any special values that you want to include in the list of prompt values. You can choose from these options:

   - **All possible values** - enables the user to select all of the prompt values at one time. The value **(all possible values)** is added to the list of prompt values.
   - **Missing values** - enables the user to select all of the missing values from the data source. The value (missing values) is added to the list of prompt values. For numeric prompts, the missing value is represented by a period (.). For more information, see "Understanding Missing Values" on page 424.

Note: You can create a special numeric missing value for numeric prompts. The unformatted value that you specify must start with a period and be followed by an underscore or a letter of the alphabet (._ or .A through .Z). The letters can be uppercase or lowercase.

6 To display each unformatted value in square brackets ([]) after the formatted value when the prompt runs, select the **Append formatted values with unformatted values** check box. SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended formatted values also appear in the **List of values** box. For more information, see "Examples of Appended Formatted Values" on page 431.

7 To create a static list for the prompt:

   a Click **Add**, and enter the value that you want to use in the **Unformatted Value** column.

   b To specify a different formatted value that is displayed to the user in the list, enter the formatted value in the **Formatted (Displayed) Value** column.

   c To specify the value as the default value, select **Default**.

d Repeat steps 1–3 to add additional values to the list. Press Enter to add a new row. The order that the values appear in the **List of values** list is how they appear to the user when the prompt is run.

Note: If you selected any of the options in the **Include Special Values** area, those special values also appear in the **List of values** box.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

8 To allow the user to add additional unformatted values to the list of values in the prompt, select the **Allow user to specify additional (unformatted) values** check box.

Create a Date Prompt with Multiple Values from a Static List

A date prompt is a prompt that contains dates. You can create a date prompt where the user selects multiple values from a static list. When creating a date prompt, you can enter a specific date or select a relative date. For more information, see "Understanding Date Values" on page 424.

To create a date prompt with multiple values from a static list:

1 On the **Prompt Type and Values** tab, select **Date** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User selects values from static list**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Multiple values** or **Multiple ordered values**. If you select **Multiple ordered values**, the user can reorder the selected values when the prompt is run.

4 From the **Date type** drop-down list, select the type of date. You can choose from these options:

- **Day**
- **Week**
- **Month**
- **Quarter**
- **Year**

5 In the **Minimum value count** and **Maximum value count** boxes, specify the minimum and maximum number of values that are required for the prompt.

- If you do not specify a minimum value count, the default is 0.
- If you do not specify a maximum value count, the default is 2,147,483,647.

Note: If you specify 0 as the minimum value, then the **Requires a non-blank value** check box on the **General** tab is automatically cleared. If you specify a value greater than 0, then the **Requires a non-blank value** check box is automatically selected.

6 In the **Minimum value allowed** and **Maximum value allowed** boxes, specify the earliest and latest dates that can be entered for the prompt. For more information, see "Specify a Date" on page 425.

- If you do not specify a minimum value, the default value is January 1, 1600.
- If you do not specify a maximum value, the default value is December 31, 2400.

7 Select any special values that you want to include in the list of prompt values. You can choose from these options:

- **All possible values** - enables the user to select all of the prompt values at one time. The value **(all possible values)** is added to the list of prompt values.
- **Missing values** - enables the user to select all of the missing values from the data source. The value (missing values) is added to the list of prompt values. For numeric prompts, the missing value is represented by a period (.). For more information, see "Understanding Missing Values" on page 424.

Note: You can create a special numeric missing value for numeric prompts. The unformatted value that you specify must start with a period and be followed by an underscore or a letter of the alphabet (._ or .A through .Z). The letters can be uppercase or lowercase.

8 To display each unformatted value in square brackets ([]) after the formatted value when the prompt runs, select the **Append formatted values with unformatted values** check box. SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended formatted values also appear in the **List of values** box. For more information, see "Examples of Appended Formatted Values" on page 431.

9 To create a static list for the prompt:

a Click **Add**, and enter the value that you want to use in the **Unformatted Value** column.

b To specify a different formatted value that is displayed to the user in the list, enter the formatted value in the **Formatted (Displayed) Value** column.

c To specify the value as the default value, select **Default**.

d Repeat steps 1–3 to add additional values to the list. Press Enter to add a new row. The order that the values appear in the **List of values** list is how they appear to the user when the prompt is run.

Note: If you selected any of the options in the **Include Special Values** area, those special values also appear in the **List of values** box.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

10 To allow the user to add additional unformatted values to the list of values in the prompt, select the **Allow user to specify additional (unformatted) values** check box.

Create a Date Prompt with a Single Value from a Dynamic List

A date prompt is a prompt that contains dates. You can create a date prompt for which the user selects either single or multiple values from a list of unique values that is retrieved from the data source that you specify when you create the prompt. The list of values is populated when the prompt is run. When you create a date prompt, you can enter a specific date or select a relative date. For more information, see “Understanding Date Values” on page 424.

To create a date prompt with a single value from a dynamic list:

1 On the **Prompt Type and Values** tab, select **Date** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User selects values from a dynamic list**. For more information, see “Understanding the Methods for Populating Prompts” on page 423.

3 From the **Number of values** drop-down list, select **Single value**.

4 From the **Date type** drop-down list, select the type of date. You can choose from these options:

   - **Day**
   - **Week**
   - **Month**
   - **Quarter**
   - **Year**

5 In the **Minimum value allowed** and **Maximum value allowed** boxes, specify the earliest and latest dates that can be entered for the prompt. For more information, see “Specify a Date” on page 425.

   - If you do not specify a minimum value, the default value is January 1, 1600.

- If you do not specify a maximum value, the default value is December 31, 2400.

6 Select the data source from which the list of values is populated.

Note: Do not select a SAS Information Map table as the data source for a dynamic list. SAS Information Map tables are tables that are in SAS Information Map libraries. Prompts cannot retrieve values from this type of table.

7 In the **Unformatted Values** box, select the column that contains the unformatted values for the list. Only columns that contain valid values for the selected prompt type are available.

By default, only distinct values are displayed. To show all the unformatted values, clear the **Show only distinct values** check box.

8 In the **Formatted (Displayed) Values** box, select the column that contains the formatted values. To use the same column for the formatted and unformatted values, select the **Use 'Unformatted Values' column** option. If you select a different column for the formatted values, select the format of the column. By default, the format from the data source is used.

9 To display each unformatted value in square brackets ([]) after the formatted value when the prompt runs, select the **Append formatted values with unformatted values** check box. SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended formatted values also appear in the **List of values** box. For more information, see "Examples of Appended Formatted Values" on page 431.

10 To include a special value that enables the user to select all of the prompt values at one time, select the **All possible values** check box. The value **(all possible values)** is added to the list of prompt values.

11 From the **Sort order** drop-down list, select the sort order for the values in the list. By default, the values are sorted in the order which they appear in the data source. You can also sort the formatted or unformatted values in ascending or descending order.

12 From the **Default values** drop-down list, select the default values for the prompt. If you selected the **All possible values** check box, you can select those values from the drop-down list. If no values appear in the drop-down list, click **Select**.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

13 To allow the user to add additional unformatted values to the list of values in the prompt, select the **Allow user to specify additional (unformatted) values** check box.

Create a Date Prompt with Multiple Values from a Dynamic List

A date prompt is a prompt that contains dates. You can create a date prompt for which the user selects either single or multiple values from a list of unique values that is retrieved from the data source that you specify when you create the prompt. The list of values is populated when the prompt is run. When creating a date prompt, you can enter a specific date or select a relative date. For more information, see "Understanding Date Values" on page 424.

To create a date prompt with multiple values from a dynamic list:

1 On the **Prompt Type and Values** tab, select **Date** from the **Prompt type** drop-down list.

2 From the **Method for populating prompt** drop-down list, select **User selects values from a dynamic list**. For more information, see "Understanding the Methods for Populating Prompts" on page 423.

3 From the **Number of values** drop-down list, select **Multiple values** or **Multiple ordered values**. If you select **Multiple ordered values**, the user can reorder the selected values when the prompt is run.

4 From the **Date type** drop-down list, select the type of date. You can choose from these options:

- **Day**
- **Week**
- **Month**
- **Quarter**
- **Year**

5 In the **Minimum value count** and **Maximum value count** boxes, specify the minimum and maximum number of values that are required for the prompt.

- If you do not specify a minimum value count, the default is 0.
- If you do not specify a maximum value count, the default is 2,147,483,647.

Note: If you specify 0 as the minimum value, then the **Requires a non-blank value** check box on the **General** tab is automatically cleared. If you specify a value greater than 0, then the **Requires a non-blank value** check box is automatically selected.

6 In the **Minimum value allowed** and **Maximum value allowed** boxes, specify the earliest and latest dates that can be entered for the prompt. For more information, see "Specify a Date" on page 425.

- If you do not specify a minimum value, the default value is January 1, 1600.

- If you do not specify a maximum value, the default value is December 31, 2400.

7 Select the data source from which the list of values is populated.

Note: Do not select a SAS Information Map table as the data source for a dynamic list. SAS Information Map tables are tables that are in SAS Information Map libraries. Prompts cannot retrieve values from this type of table.

8 From the **Unformatted Values** drop-down list, select the column that contains the unformatted values for the list. Only columns that contain valid values for the selected prompt type are available.

9 From the **Formatted (Displayed) Values** drop-down list, select the column that contains the formatted values. To use the same column for the formatted and unformatted values, select the **Use 'Unformatted Values' column** option. To select a different column for the formatted values, click **Select**. By default, the format from the data source is used.

10 To display each unformatted value in square brackets ([]) after the formatted value when the prompt runs, select the **Append formatted values with unformatted values** check box. SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended formatted values also appear in the **List of values** box. For more information, see "Examples of Appended Formatted Values" on page 431.

11 To include a special value that enables the user to select all of the prompt values at one time, select the **All possible values** check box. The value **(all possible values)** is added to the list of prompt values.

12 From the **Sort order** drop-down list, select the sort order for the values in the list. By default, the values are sorted in the order which they appear in the data source. You can also sort the formatted or unformatted values in ascending or descending order.

13 From the **Default values** drop-down list, select the default values for the prompt. If you selected the **All possible values** check box, you can select those values from the drop-down list. If no values appear in the drop-down list, click **Select**.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

14 To allow the user to add additional unformatted values to the list of values in the prompt, select the **Allow user to specify additional (unformatted) values** check box.

Create a Date Range Prompt

A date range prompt contains two dates that are the upper and lower boundaries for a range of dates. When creating a date range prompt, you can enter a specific date or select a relative value. For more information, see "Understanding Date Values" on page 424.

To create a date range prompt:

1. On the **Prompt Type and Values** tab, select **Date range** from the **Prompt type** drop-down list.

2. From the **Date type** drop-down list, select the type of date. You can choose from these options:

   - **Day**
   - **Week**
   - **Month**
   - **Quarter**
   - **Year**

3. In the **Minimum value allowed** and **Maximum value allowed** boxes, specify the earliest and latest dates that can be entered for the prompt. For more information, see "Specify a Date" on page 425.

   - If you do not specify a minimum value, the default value is January 1, 1600.
   - If you do not specify a maximum value, the default value is December 31, 2400.

4. From the **Range type** drop-down list, specify the type for the default range. The available options depend on the selected date type.

5. In the **From** and **To** boxes, specify the default values for the range.

   The default values in the **From** and **To** boxes depend on the selected type. The following table lists the default ranges for each range type. For many values, you must specify the value of *N* to determine the range.

   Table 11.1 *Default Ranges for Each Range Type*

Range type	**From**	**To**
Year to date	Beginning of current year	Today
Month to date	Beginning of current month	Today
Previous N days	*N* days ago	Today
Next N days	Today	*N* days from now

Range type	From	To
Previous N weeks	*N* weeks ago	Current week
Next N weeks	Current week	*N* weeks from now
Previous N months	*N* months ago	Current month
Next N months	Current month	*N* months from now
Previous N quarters	*N* quarters ago	Current quarter
Next N quarters	Current quarter	*N* quarters from now
Previous N years	*N* years ago	Current year
Next N years	Current year	*N* years from now

If you select **Custom** as the range type, you can enter values in the **From** and **To** boxes. For more information, see “Specify a Date” on page 425.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

Create a Time Prompt with a Single Value

A time prompt contains time values. You can create a prompt with single or multiple values. When creating a time prompt, you can enter a specific time or select a relative value. For more information, see “Understanding Time Values” on page 428.

To create a time prompt with a single value:

1 On the **Prompt Type and Values** tab, select **Time** from the **Prompt type** drop-down list.

2 From the **Number of values** drop-down list, select **Single value**.

3 In the **Minimum value allowed** and **Maximum value allowed** boxes, specify the earliest and latest timestamp that can be entered for the prompt. For more information, see “Specify a Time” on page 429.

- If you do not specify a minimum value, the default value is 12:00:00 AM.
- If you do not specify a maximum value, the default value is 11:59:59 PM.

4 Select any special values that you want to include in the list of prompt values. You can choose from these options:

- **All possible values** - enables the user to select all of the prompt values at one time. The value **(all possible values)** is added to the list of prompt values.
- **Missing values** - enables the user to select all of the missing values from the data source. The value (missing values) is added to the list of prompt values. For numeric prompts, the missing value is represented by a period (.). For more information, see "Understanding Missing Values" on page 424.

Note: You can create a special numeric missing value for numeric prompts. The unformatted value that you specify must start with a period and be followed by an underscore or a letter of the alphabet (._ or .A through .Z). The letters can be uppercase or lowercase.

5 In the **Default value** box, specify the default value for the prompt. If you selected any of the options in the **Include Special Values** area, you can select those special values from a drop-down list in the **Default value** box. For more information, see "Specify a Time" on page 429. The default value is used if the user does not specify a value when prompted.

Note: If you selected either the **Hide at run time** or **Read-only values** option and the **Requires a non-blank value** option on the **General** tab, you must specify a default value.

Understanding Prompt Values

Understanding the Methods for Populating Prompts

When you create a prompt, you must specify how the values for the prompt are obtained using the **Method for populating prompt** drop-down list. There are three methods that you can use:

- **User enters values** - specifies that the user enters the value or values when the prompt is run.
- **User selects values from a static list** - specifies that the user selects from a fixed set of values that you define when you create the prompt.
- **User selects values from a dynamic list** - specifies that the user selects from a list of unique values. This list is populated by values from the selected data

source when the prompt is run. For example, suppose you create a prompt called Customer_prompt. The values for this prompt are retrieved from the Customer variable in the selected data source. New customer values are added frequently. When the prompt runs, the list is populated by the most current list of customers.

Understanding Missing Values

A missing value is a placeholder for a value that does not exist. Prompts allow for three types of missing values:

- **Character** - This type of missing value is represented by a blank.
- **Numeric** - This type of missing value is represented by a period (.).
- **Special numeric** - This type of missing value is represented by a period followed by an underscore or a letter of the alphabet (._ or .A through .Z). The letters can be uppercase or lowercase. Special numeric missing values can be used to represent different categories of missing data.

Note: Numeric and special numeric missing values are also used to represent missing date, time, and timestamp values.

Understanding Numeric Values

If you are working with an integer prompt, then do not enter values with decimal places. If you enter a value with zeros after the decimal point (for example, 1.00) for an integer prompt, then the zeros and the decimal point will be removed before the value is stored (for example, the 1.00 will be stored as 1).

For prompts that allow floating-point values, the unformatted prompt value can contain up to 15 significant digits. Values with more than 15 significant digits of precision will be truncated. Note that formatted values can have more than 15 significant digits.

Understanding Date Values

About Specific and Relative Dates

When creating a date prompt, you can select a specific date or a relative date.

- If you select a specific date, the prompt value is static. The specific date begins at midnight (12:00:00 a.m.) and ends at 11:59:59 p.m.
- If you select a relative date, the prompt value is dynamic. These values (such as yesterday, next month, and previous week) are set when the prompt is run and are based on the system date of the server that is processing the prompt.
  - All relative dates are based on the Gregorian calendar. Each year begins with the month of January and each week begins on Monday, regardless of locale.
  - A relative date begins at midnight (12:00:00 a.m.) on the first day of that value and ends at 11:59:59 p.m. on the last day of that value.

For example, if you select **Previous week**, you are specifying a period that ranges from 12:00:00 a.m on the Monday (first day) of the week before run time through 11:59:59 p.m. on the Sunday (last day) of the week before run time.

Specify a Date

You can specify a date in these ways:

- Select a specific date. The options that are available depend on the **Date type**.
- Select a relative date from the drop-down list. If you selected a date (such as *N* days ago) that requires a user-supplied value, specify this value in the **N** box.
- Enter a date in one of these Day, Week, Month, Quarter, or Year formats.

  Day

  You can specify these day formats. However, SAS Enterprise Guide automatically converts the date to the *month-name dd, yyyy* format.

Format	Examples
mm/dd/yy<yy>	12/02/08 04/20/1919
mm.dd.yy<yy>	11.03.08 07.20.1956
mm-dd-yy<yy>	10-05-06 05-13-1920
mmm/dd/yy<yy>	Oct/02/08 APR/20/2117
mmm.dd.yy<yy>	Nov.02.09 JUL.20.13
mmm-dd-yy<yy>	Oct-05-06

Format	Examples
	MAY-13-1924
ddmmmyyyy	14May2011 04APR2010
month-name dd, yyyy	February 02. 2008 SEPTEMBER 20., 2010
day of week, month-name dd, yyyy	Tuesday, January 15, 2008 FRIDAY, JANUARY 04, 2008

Here is an explanation of the syntax for the day formats:

dd
specifies a two-digit integer that represents the day of the month.

day-of-week
specifies the full name of the day of the week. This value is not case sensitive.

mmm
specifies the first three letters of the full name of the month. This value is not case sensitive.

month-name
specifies the full name of the month. This value is not case sensitive.

yy or *yyyy*
specifies a two-digit or four-digit integer that represents the year. Valid values range from 1600 to 2400.

Week
You can specify these week formats. However, SAS Enterprise Guide automatically converts the date to the *Week ww yyyy* format.

Format	Examples
Ww yy	W52 20 W1 76
Week ww yyyy	Week 20 2020 Week 5 1976

Here is an explanation of the syntax for the week formats:

w
specifies a one-digit or two-digit integer that represents the week of the year. Value values range from 1 to 52.

yy
specifies a two-digit integer that represents the year. To refer to a year that is more than 80 years in the past or 20 years in the future, use four digits.

yyyy
specifies a four-digit integer that represents the year. Valid values range from 1600 to 2400.

Month
When you specify a month, you can use these formats:

Format	Examples
mm/yy<yy>	8/20 10/1975
mm.yy<yy>	05.13 11.1918
mm-yy<yy>	03-21 12-2016
mmm/yy<yy>	Feb/20 Aug/1970
mmm.yy<yy>	SEP.18 NOV.2009
mmm-yy<yy>	JUN-22 MAR-1990
month-name yyyy	June 2010 JULY 2025

Here is an explanation of the syntax for the month format:

mm
specifies a two-digit that represents the month of the year.

mmm
specifies the first three letters of the full name of the month. This value is not case sensitive.

month-name
specifies the full name of the month. This value is not case sensitive.

yy or *yyyy*
specifies a two-digit or four-digit integer that represents the year. Valid values range from 1600 to 2400.

Quarter

When specifying a quarter, use this format:

Format	Examples
quarter-name quarter yy<yy>	1st quarter 12 2nd quarter 50 3rd quarter 2012 4th quarter 1950

Here is an explanation of the syntax for the quarter format:

quarter-name
specifies the quarter of the year. Valid values are 1st, 2nd, 3rd, and 4th.

yy or *yyyy*
specifies a two-digit or four-digit integer that represents the year. To refer to a year that is more than 80 years in the past or 20 years in the future, use four digits. Valid values for a four-digit year range from 1600 to 2400.

Year

When specifying a year, use this format:

Format	Examples
yyyy	2035 1995

Here is an explanation of the syntax for the year format:

yyyy
specifies a four-digit integer that represents the year. Valid values range from 1600 to 2400.

Understanding Time Values

About Specific and Relative Times

When creating a time prompt, you can select a specific time or a relative time.

- If you select a specific time, the prompt value is static.
- If you select a relative time, the prompt value is dynamic. These values (such as current time, beginning of next hour, and n minutes ago) are set when the

prompt is run and are based on the system time of the server that is processing the prompt.

Specify a Time

You can specify a time in these ways:

- Click 🕒 and select the time.
- Select a relative time from the drop-down list. If you selected a time (such as *N* minutes ago) that requires a user-supplied value, specify this value in the **N** box.
- Use this format to enter a time. However, SAS Enterprise Guide automatically converts the time to the *hh:mm:ss* AM | PM format.

Format	**Examples**
hh:mm<:ss><AM | PM>	13:01:01 1:01:01 PM 1:01 AM 22:05

Here is an explanation of the syntax for the time format:

hh

specifies a one or two-digit integer that represents the hour of the day. Valid values range from 0 to 24.

mm

specifies a two-digit integer that represents the minute of the hour. Valid values range from 0 to 59.

ss (optional)

specifies a two-digit integer that represents the second of the minute. Valid values range from 0 to 59. If this value is not specified, the value defaults to 00 seconds.

AM or PM (optional)

specifies either the time period 00:01–12:00 noon (AM) or the time period 12:01–12:00 midnight (PM). If this value is not specified and you are using the 12–hour system for specifying time, the value defaults to AM.

Note: Do not specify AM or PM if you are using the 24-hour system for specifying time.

Understanding Timestamp Values

About Specific and Relative Timestamps

When creating a timestamp prompt, you can select a specific timestamp or a relative timestamp.

- If you select a specific timestamp, the prompt value is static.
- If you select a relative timestamp, the prompt value is dynamic and is based on the Gregorian calendar. The value of a relative timestamp must be within 200 years of the current date. These relative values (such as current date and time, beginning of previous hour, and n minutes ago) are set when the prompt is run and are based on the system date and time of the server that is processing the prompt.

Specify a Timestamp

You can specify a timestamp in these ways:

- Click 📅 to select the date and time.
- Select a relative timestamp from the drop-down list. If you selected a time (such as *N* minutes ago) that requires a user-supplied value, specify this value in the **N** box.
- Use one of these formats to enter a timestamp. However, SAS Enterprise Guide automatically converts the timestamp to *month-name dd, yyyy hh:mm:ss* AM | PM format.

Format	Examples
mmm/dd/yy hh:mm:ss	07/03/08 12:40 PM 09/22/22 09:09 AM
*yyyy-mmm-dd*T*hh:mm:ss* AM | PM	2012–11–23T15:30:32 2008-09-09T01:01:01
<day-of-week,> month-name dd, yyyy hh:mm:ss AM | PM	Friday, July 27, 2007 04:45:45 PM December 05, 2020 12:00:00 AM

Here is an explanation of the syntax for the timestamp format:

dd
specifies a one-digit or two-digit integer that represents the day of the month.

day-of-week (optional)
specifies the full name of the day of the week. This value is not case sensitive.

mmm
specifies a one-digit or two-digit integer that represents the month of the year. This value is not case sensitive.

month-name
specifies the full name of the month. This value is not case sensitive.

yy or *yyyy*
specifies a two-digit or four-digit integer that represents the year. Valid values range from 1600 to 2400.

hh
specifies a two-digit integer that represents the hour of the day. Valid values range from 0 to 24.

mm
specifies a two-digit integer that represents the minute of the hour. Valid values range from 0 to 59.

ss
specifies a two-digit integer that represents the second of the minute. Valid values range from 0 to 59.

AM or PM
specifies either the time period 00:01–12:00 noon (AM) or the time period 12:01–12:00 midnight (PM). If this value is not specified and you are using the 12–hour system for specifying time, the value defaults to AM.

Note: Do not specify AM or PM if you are using the 24-hour system for specifying time.

Examples of Appended Formatted Values

SAS Enterprise Guide automatically appends the unformatted values to any duplicate formatted values. These appended values appear to the user when the prompt runs. The following table provides an example of an appended formatted value for each text, numeric, and date prompt.

Prompt Type	Unformatted Value	Formatted Value	Value Displayed When Prompt Runs
Text	United States	US	US[United States]

Prompt Type	Unformatted Value	Formatted Value	Value Displayed When Prompt Runs
	USA	US	US[USA]
Numeric	9	High	High[9]
	10.0	High	High[10.0]
Date	February 10, 2008	FEB08	FEB08[February 10, 2008]
	February 22, 2008	FEB08	FEB08[February 22, 2008]

Note: If you are running a stored process that was created in a SAS application other than SAS Enterprise Guide, the unformatted and formatted values for a multiple value prompt are displayed in separate columns.

12

Working with Results

Viewing Results

When you run a task or program in SAS Enterprise Guide, the results are displayed in the work area. By default results are generated as HTML5 output and displayed on the Results tab.

To view results:

- From the Project pane, double-click the task or code for which you want to view the results and click the **Results** tab.
- From the process flow, double-click the results that you want to view.

By default, HTML5, SAS Report, and Listing results open in the SAS Enterprise Guide work area. To change the default viewer for your results, you can use the Options window to specify a viewer for results. For more information, see "Setting Viewer Options" on page 488.

PDF, RTF, Microsoft Excel, and Microsoft PowerPoint results open in their default application.

Working with SGE Files

SAS ODS Graphics Editor (SGE) files are files in which all of the graphical elements (titles, footnotes, and so on) are available for individual editing. In addition, you can add annotations on top of the graph. You can create an SGE file in SAS Enterprise Guide by inserting the following code in a program:

```
ods listing sge=on;
```

SAS Enterprise Guide can launch the SAS/GRAPH ODS Graphics Editor to view or edit SGE files if these conditions are met:

- The SAS/GRAPH ODS Graphics Editor is installed on your local machine.
- SAS Enterprise Guide is connected to a SAS 9.4 server.

If the SAS/GRAPH ODS Graphics Editor is not installed, an error message is displayed.

If you use the SAS/GRAPH ODS Graphics Editor to make changes to the SGE file from within SAS Enterprise Guide, you cannot save the edited SGE file in your SAS Enterprise Guide project. You can save the edited file as a new SGE file, but you cannot access the saved file from within SAS Enterprise Guide. SAS Enterprise Guide can access only the SGE file that was originally created by the program.

Note: SGE files cannot be exported as a step in a program, sent to another user as a step in a program, or published to a repository.

For more information about SGE files, see *SAS/GRAPH ODS Graphics Editor User's Guide*.

Previewing Results

You can preview results within a SAS Enterprise Guide project before you print them.

To preview results:

1 Open the results item that you want to preview by clicking the appropriate tab in the work area or by double-clicking the item in the process flow.

2 Select **File** ⇨ **Print Preview for *results-name***.

Printing Results

You can print your results from within a SAS Enterprise Guide project. You can also change the print setup options that affect how the page looks when it is printed.

To change the print setup options:

1 Open the results item that you want to print by clicking the appropriate tab in the work area or by double-clicking the item in the process flow.

2 Select **File** ⇨ **Page Setup for *results-item***.

3 The Page Setup window varies depending on the content that you are printing. Specify any appropriate options and click **OK**.

To print results:

1 Open the results item that you want to print by clicking the appropriate tab in the work area or by double-clicking the item in the process flow.

2 Select **File** ⇨ **Print *results-item***.

3 Select any appropriate print options and click **Print** to print the results.

Working with Reports

About Reports

When you create results in the SAS Report format, you can use those results to create a customized report that you can print, export, and share with other SAS applications. You can select from a range of options to customize the printing of your report. You can add multiple results to the report along with text and images, and you can choose how to arrange them in the report. When the task or code that generated the results is rerun, the results in the report are automatically refreshed.

Note: You can also choose to create a snapshot of the results that does not get refreshed.

Creating and Opening Reports

Create a Report

You can create a report by using existing results in the SAS Report format and by inserting text and images. After you create a report, you can edit it.

Note: If the task or code that generated the results is rerun, the results in the report are automatically refreshed. If you do not want the results to be refreshed, you can create a report snapshot.

To create a report:

1. Select **File** ⇨ **New** ⇨ **Report**.
2. In the **Select SAS items** list, select the items that you want to add to the report and add them to the **Report layout** area.

   Note: If you have not created any task or code results in the SAS Report format, no results are listed in the **Select SAS items** list.

3. To add text, click **Insert Text**.

   Note: After the text is added to the report layout area, you can edit the text by double-clicking the text in the report, or by selecting the text and pressing the spacebar.

4. To add an image, click **Insert Image**.

   Note: After you create the report, you can resize the image.

5. To change how the items are arranged in the report:
   - To arrange items, use the drag-and-drop functionality in the **Report layout** area.
   - To increase or decrease the space that an item is allotted, select the item and drag the appropriate handle on its border.
6. To create the report, click **OK**. The report opens in the work area.

Duplicate a Report

In the Project pane, right-click the report name and select **Create** ⇨ **Duplicate report**. A copy of the report, named Report*n*, is created.

Create a Report Snapshot

You can create a report snapshot from another report, from an OLAP cube, or from results that are saved in the SAS Report format. A report snapshot is a report that contains a static copy of the data. The snapshot is not updated when the task or code that generated the results is rerun, or when the view of the OLAP cube is updated. You can add text and images to the report snapshot, change how the items are arranged in the snapshot, and modify the headers and footers.

To create a report snapshot, right-click the report, OLAP object, or results in the SAS Report format that you want to use, and then select **Create** ⇨ **Report snapshot**. The report snapshot opens in the work area.

To insert text into a report snapshot:

1. On the Report Snapshot toolbar, select **Design** ⇨ **Insert text**.
2. Select the font, size, font style, alignment, and color that you want to use.
3. Enter the text that you want to insert.
4. To link the text to a website or a file, insert a link.
5. Click **OK**. The text is added to the report snapshot.

Note: You can edit text that you have inserted by double-clicking it in the report snapshot, or by selecting the text and pressing the spacebar.

To insert an image into a report snapshot:

1. On the Report Snapshot toolbar, select **Design** ⇨ **Insert image**.
2. Select the file that you want to insert.
3. Click **Open**. The image is added to the report snapshot.

Note: You can resize the image by selecting it and dragging the handles until it is the size that you want.

To move items in the report snapshot, select the text or image that you want to move and drag it to the location that you want. If you drag the item to the top or bottom of the report snapshot, you are prompted for how you want to position the item in relation to any existing titles, footnotes, headers, or footers. You cannot toggle the titles and footnotes on or off in a report snapshot.

Editing Reports

Editing an Existing Report

You can edit an existing report and change the report contents and layout. If you only want to insert text or an image into the report or remove all of the page breaks from the SAS items, you can do that without opening the Edit Report Contents window.

To edit an existing report:

1. Open the report that you want to edit by double-clicking it in the Project pane or in a process flow. You can also open the report by right-clicking the report and selecting **Open** from the pop-up menu.
2. On the Report toolbar, click **Edit Contents**.

   Note: You cannot edit a report when you are viewing the page layout in the Page View.

3. When you edit a report, you have all the same options for adding, deleting, and rearranging items that you have when you create a report.
4. To save the report after you make changes, click **OK**. The report is updated in the work area.

Inserting Text into an Existing Report

You can edit an existing report and change the report contents and layout. You can also insert text or an image into the report without opening the Edit Report Contents window.

To insert text into a report:

1. On the Report toolbar, select **Design** ⇨ **Insert text**.
2. Select the font, size, font style, alignment, and color that you want to use.
3. Enter the text that you want to insert.
4. To link the text to a website or a file, insert a link.
5. Click **OK**. The text is added to the report.

Note: The text is inserted at the bottom of the report by default. You can move the text to a new location by dragging it. You can also edit text that you have inserted by double-clicking it in the report.

Viewing Text Properties

You can view the source of text in a report, and you can add a border or background color to the text. To view the properties for text in a report, right-click the text in the report and select **Properties**.

- **Source** – the source of the text. If the text was entered by a user, the source is listed as "Edited by user." If the text was not entered by a user, such as an automatically generated report title or footnote, the source is listed as "Generated by SAS."
- **Border** – displays a border around the text.
- **Background** – enables you to specify a background color for the text. The default background color is white.

Applying the Report Style to All Items

You can customize the style of the items in your report. For example, you can change the font type, the font color, and the font size of the report title or footnote.

To override any customizations that you made to the items in the report and return to the original report style, select **Design** ⇨ **Apply report style to all items** from the Report toolbar.

Insert Link into an Existing Report

To link to a web page or insert a file into a report:

1. Open the report by double-clicking it in the Project pane or in the process flow. You can also open the report by right-clicking the report and selecting **Open** from the pop-up menu.
2. On the Report toolbar, click **Edit Contents**.
3. In the Edit Report Contents window, click **Insert Text** or double-click the existing text.
4. Select the text that you want to use as an anchor and click 🌐.
5. In the Edit Hyperlink window, select **Link to web page/file**.
6. Click **Browse** to select the web page or file that you want to link to.

7. Enter the tooltip text in the **Tool tip** field.

8. Click **OK**. The text is linked to a web page or file that is displayed when you click the text in the report. You can add a new line at the end of linked text. You cannot add a new line within linked text.

Note: To delete linked text, position the cursor at the beginning of the linked text and press Delete. You cannot press Delete to delete a selection that contains both linked and unlinked text. You cannot use CTRL+Z or CTRL+Y to undo or redo the entering of linked text.

To remove all of the links in the report:

1. Open the report by double-clicking it in the Project pane or in the process flow. You can also open the report by right-clicking the report and selecting **Open** from the pop-up menu.

2. On the Report toolbar, click **Edit Contents**.

3. Click **Insert Text** or double-click the existing text.

4. Select the text that you want to use as an anchor and click 🌐.

5. In the Edit Hyperlink window, select **Remove hyperlinks**.

6. Click **OK**. All of the links are removed from the report.

Insert an Image into an Existing Report

You can edit an existing report and change the report contents and layout. You can also insert text or an image into the report without opening the Edit Report Contents window.

Note: When you add an image to a report, a reference is created in the report to that image file. If any changes are made to the image file, the image that is displayed in the report is also changed. If the image file is deleted, the image that is displayed in the report is replaced with a missing-file icon.

To insert an image into a report:

1. On the Report toolbar, select **Edit Contents**.

2. Select the file that you want to insert.

3. Click **Open**. The image is added to the report.

   Note: The image is inserted at the bottom of the report by default. You can move the image to a new location by dragging it.

Removing Page Breaks from an Existing Report

You can edit an existing report and change the report contents and layout. You can also remove all page breaks from the SAS items in the report.

On the Report toolbar, select **Design** ⇨ **Remove all page breaks**. The page breaks are removed from all of the SAS items in the report.

Working with Headers and Footers in a Report

You can customize a report by adding headers and footers to it. These headers and footers are printed at the top and bottom of each page of the report and are saved with the report. You can also view the headers and footers at the top and bottom of each page when you are viewing the page layout in the Page View.

You can choose whether to display the report title and footnote that are generated with the results, and you can change the appearance of the report title and footnote. These changes are reapplied when the task or code that generated the results is rerun.

To add headers and footers to a report:

1. Open the report to which you want to add headers and footers by double-clicking it in the Project pane or in the process flow.
2. On the Report toolbar, click **Design** ⇨ **Header & Footer**.
3. Click the **Header** tab or the **Footer** tab.
4. In the **Text** box, enter the text that you want to use as your header or footer.
5. Click **OK** to save the header or footer and apply it to your report.

To toggle the display of the report title and footnote:

1. Open the report to which you want to add headers and footers by double-clicking it in the Project pane or in the process flow.
2. On the Report toolbar, click **Design** ⇨ **Header & Footer**.
3. Click the **Titles & Footnotes** tab.
4. To toggle the title or footnote from a result item in the report, select or clear the appropriate **Title** or **Footnote** check box for the item.

Exporting Reports

You can export a report to a file in the SAS Report (*.srx) format. You can export the report directly, or you can add an export step to the current SAS Enterprise Guide project so that the report is exported each time the project is run.

Note: You cannot open an exported report in SAS Web Report Studio. If you want to use SAS Web Report Studio to view a report, you must publish the report.

To export a report to a file:

1. In the Project pane or process flow, right-click the report that you want to export and select **Share** ⇨ **Export**.
2. Select **My Computer** or **Servers** as the location in which to save the export file.
3. To change the folder or server in which the file is saved, select the correct folder or server from the **Save in** drop-down list.
4. Specify the file name.
5. To change the file type, select a new file type from the **Files of type** drop-down list.
6. Click **Save** to create the export file.

To export a report as a step in a project:

1. In the Project pane or process flow, right-click the report that you want to export and select **Share** ⇨ **Export as a step in project**.

   The report is selected in the **File To Export** list. Click **Next**.
2. Select the output type. Click **Next**.
3. Select **Local Computer** or **Servers** as the location for the output file. Click **Next**.
4. After confirming your selections, click **Finish** to export the report as a step in the process flow of the project.

13

Exporting and Sending Files

Export a File

You can export a data, code, or results file to a specified location. When you export a file to a specified location, SAS Enterprise Guide does not change the reference to the file in the project.

To export a file:

1 Right-click the item that you want export in the work area or in the process flow and select **Share** ⇨ **Export**.

2. Select the appropriate options for the file and click **Save** or **Export** to export the file.

Note: If you are exporting to a .sas7bdat file, you must be connected to your local SAS server. If the **Export** option is not available, verify that the **Automatically add local SAS server (if installed) to server list** option is selected in your server profile. For more information, see “Selecting Your Default Server” on page 10.

Export All Code

You can export all the code in a project or process flow into a single program file that you save to a specified location. When you export a program to a specified location, SAS Enterprise Guide does not change the reference to the program file in the project.

Note: Exporting all of the code in a project is not the same as saving the project itself. You cannot use the exported code to re-create a project and its tasks. To save all of the tasks in a project, you need to save the project.

To export all code in a project or process flow:

1. To export all of the code in the project, click ⋮ in the **Project** pane. Select **Share** ⇨ **Export all code in project**.

   To export all of the code in a process flow, select **Share** ⇨ **Export all code in process flow** on the **Process Flow** tab.

2. Specify the location where the code file will be saved.

3. From the **Encoding** drop-down list, select the type of encoding that you want to use for the exported data. By default, the exported data is encoded in the same way that the original data was encoded.

4. In the **Code available for export** list, clear the check boxes next to any code items that you do not want to export.

5. In the **Code preview** area, you can click **Preview All Selected** to preview all of the selected code items, or you can click **Preview of *code-item*** to view only the selected code item.

6. You can select these options:

   - **Include headers** to include headings that provide descriptive information about the code.

   - **Include generated wrapper code** to include wrapper code that SAS generates.

- **Include custom SAS code from the Tools --> Options dialog** to include SAS code that is specified in the Options dialog box.
- **Include library assignment code** to include the code that SAS generates to assign the library.

7 Click **Export** to export the selected code items.

Exporting Data or Code Files as a Step in a Project

Overview

You can export code, data files, and reports as a step in the process flow of a project. The export step is added to the process flow and the Project pane. The export step is executed when the project is run.

Note: When you export a data file to a different file type, the data values might not be formatted exactly the same.

Step 1: Select the Files to Export

1 In the work area or in the process flow, select the item that you want to export in the work area or in the process flow.

2 From the toolbar, select **Share** ⇨ **Export**.

3 If there is an error in creating the file that you are exporting, you can use the **Export even when errors occur in the file** option to choose how you want SAS Enterprise Guide to handle the error. If you select **Export even when errors occur in the file**, SAS Enterprise Guide attempts to export an earlier version of the file with the same name, if one exists. If you do not select this option, SAS Enterprise Guide displays an error during the export step and continues running the rest of the project.

4 From the **File To Export** list, select the file that you want to export.

5 If you have selected a data file, results in SAS Report format, or a log file, click **Next** to select the type of output file.

Otherwise, click **Next** to specify the name and location of the output file.

Step 2: (Optional) Select the Output File Type

1. Before you can select the output file type, you must select the file to be exported.
2. Select the file type from the **Output File Type** list.
3. If you selected a file type other than SAS, click **Next** to modify additional output file options.

   If you selected a SAS file type, click **Next** to specify the name and location of the output file.

Step 3: (Optional) Modify the Output File Options

Depending on the output file type, you can modify additional output file options.

To modify output file options:

1. Before you can select the output file options, you must select the output file type.
2. To use column labels instead of the column names in the exported file, select **Use labels for column names**.
3. Click **Next** to specify the name and location of the output file.

Step 4: Specify the Name and Location of the Output File

1. Select **Local Computer** or **SAS Servers** as the location of the output file.
2. To overwrite an existing file with the same name, select **Overwrite existing output**. If you do not select this option and a file with the same name already exists, a file name is created by appending the current date and time to the existing file name.
3. If you have selected a code file to export, click **Next** to preview the code to export.

   Otherwise, click **Next** to create the export file.

Step 5: (Optional) Preview the Code to Export

1. In the **Code available for export** list, clear the check boxes next to any code items that you do not want to export.
2. In the **Code preview** area, you can click **Preview All Selected** to preview all of the selected code items, or you can click **Preview of *code-item*** to view only the selected code item.
3. You can select these options:
   - **Include headers** to include headings that provide descriptive information about the code.
   - **Include generated wrapper code** to include wrapper code that SAS generates.
   - **Include custom SAS code from the Tools --> Options dialog** to include SAS code that is specified in the Options dialog box.
   - **Include library assignment code** to include the code that SAS generates to assign the library.
4. Click **Next** to create the export file.

Step 6: Create the Export File

1. View the selections that you have made. Click **Back** to make any changes.
2. When you have confirmed your export selections, click **Finish** to export the file as a step in the process flow of the project.

Sending a Single File

Send a File to Another User

You can send a copy of a file in a SAS Enterprise Guide project to another user through electronic mail. Files that you can send include data, programs, logs, results, and notes. To send files through email, you need an email application such as Microsoft Outlook that supports the Messaging Application Programming Interface (MAPI).

Note: You cannot send a report such as a SAS Web Report Studio report that you have opened from a report repository to another user or to a Microsoft Office application.

To send a file to another user:

1 In the Project pane, process flow, or work area, select the file that you want to send to another user.

2 Select **Share** ⇨ **Email**.

3 If you are sending a data file, the Select Format dialog box appears. Select the file format for the data from the list of formats and click **OK**.

4 An email message is created and the file is added as an attachment to the message. Address and send the message to whomever you want to receive the file.

Note: If the data is in Update mode, the data will be protected before it is added.

Send a File to a Microsoft Application

Depending on the file type, you can send a copy of the active file to Microsoft Word, Microsoft Excel, or Microsoft PowerPoint. In order to send results in the SAS Report format to any Microsoft Office application, you must have SAS Add-In for Microsoft Office 4.3 or later installed on your computer.

Note: You cannot send a report such as a SAS Web Report Studio report that you have opened from a report repository to another user or to a Microsoft Office application.

To send a file to a Microsoft application:

1 In the Project pane, process flow, or work area, select the file that you want to send.

2 Select **Share** ⇨ **Send To *name-of-Microsoft-application*** where *name-of-Microsoft-application* is Microsoft Word, Microsoft Excel, or Microsoft PowerPoint.

   Note: You cannot send a SAS data set to Microsoft PowerPoint.

   If you select Word, Excel, or PowerPoint and that Microsoft application is running, you must specify the location of the results. You can send the results to a new document or an active document. If you send the results to an active document and you have multiple documents open, you can select which document to send the results to.

When you send results to an active document, the results appear in these locations:

- In Word, the SAS add-in places the results at the location of the cursor in the active document.
- In Excel, the SAS add-in inserts the results at the location of the cursor in the active workbook.
- In PowerPoint, the SAS add-in inserts the results in a new slide after the active slide.

Send HTML Results to Your Default Browser

If your results are in an HTML format, you can send these results to your default browser. Right-click the HTML results in the process flow and select **Share** ⇨ **Send to system viewer**. The active file opens in a new tab or browser window, depending on your browser settings.

Send a Data File to JMP

If JMP 7, JMP 8, or a later release of JMP is installed on your computer, you can send a data file from SAS Enterprise Guide to JMP. When sending a file to JMP, SAS Enterprise Guide creates a temporary copy of your selected data file and opens it in a new JMP session, where you can then use the visualization features of JMP.

You can send these types of data files from SAS Enterprise Guide to JMP:

- SAS data sets from your local computer
- SAS data sets from the file system
- any data source from a SAS library

To send a file to JMP:

1. In the Project pane, process flow, or work area, select the data file that you want to send.
2. Select **Share** ⇨ **Send to JMP**. The active file is opened in JMP.

Note: You can also send a data file to JMP by clicking **Share** ⇨ **Send to JMP** on the work area toolbar.

Send a Project to Other Users

You can send a copy of your SAS Enterprise Guide project to other users through electronic mail. To send documents through email, you need an email application such as Microsoft Outlook or Lotus cc:Mail that supports either the Messaging Application Programming Interface (MAPI) or Vendor Independent Messaging (VIM).

To send a project to other users:

1 In the **Projects** pane, click ⋮. Select **Share** ⇨ **Email project**.

2 Address and send the message to whomever you want to receive the project.

Sending Email as a Step in a Project

About Sending Email as a Step in a Project

You can send email as a step in the process flow of a project. The email step is added to the process flow, and it is displayed in the Distribute container in the Project pane. The email step is executed when the project is run.

Note: SAS Enterprise Guide uses SMTP (Simple Mail Transport Protocol) to send email as an unattended step in a process flow. To configure the SMTP server and sender's address, select **Tools** ⇨ **Options** and click **Administration** in the selection pane.

Step 1: Select the Files to Send

1 Select an item in the work area or process flow.

2 Select **Share** ⇨ **Email as step in project**.

3 Click **Add** to add files to the **Files To Send** list. You can select files from the current project, your computer, or a SAS server.

Note: You must have SAS installed on your local computer in order to send a SAS data file to an email recipient.

4 If there is an error in creating the file that you are sending, you can use the **Send even when errors occur in the files** option to choose how you want SAS Enterprise Guide to handle the error. If you select **Send even when errors occur in the files**, SAS Enterprise Guide sends the file with the error. If you do not select this option, SAS Enterprise Guide does not send the file that contains the errors.

5 To create a ZIP file that includes all of the companion files to an HTML file, select **Compress all files; include banner image and style for results**. If you do not select this option, only the main HTML file is sent.

6 To rename the compressed file that is attached to the email, select **Rename compressed file to** and enter the name of the file in the box.

7 Click **Next** to specify the recipients of the email.

Step 2: Specify the Recipients

1 Enter the addresses of the email recipients in the **Recipient** box. Separate addresses with a semicolon.

2 Enter a subject in the **Subject** box.

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

4 To include a message in the body of the email, enter the message in the **Message** box.

5 Click **Next** to send the email.

Send the Email

To send the email now, as well as when the project is run, select **Send email immediately**. When you have confirmed the selections that you have made, click **Finish** to send email as a step in the project.

14

Publishing Data and Results

About Publishing

Publishing is a way of delivering electronic information to interested users in a timely manner. SAS Enterprise Guide enables you to publish data and task results to predefined channels, which function as repositories to which users subscribe. Any information that is published to a channel is delivered to all of that channel's subscribers through email, the SAS Portal, or some other method.

Before you can publish items and reports in SAS Enterprise Guide, you must set up publishing.

Note: You must be connected to a server to publish data and results. The publication channels are defined on the metadata server that you connect to using your profile. You might need to change your profile in order to publish content. For more information, see "Managing Profiles" on page 9.

See Also

- “Publishing Items to the Enterprise” on page 454
- “Publishing SAS Reports” on page 457

Setting Up Publishing

Before you can publish information, you must set up publishing by completing these tasks:

1. Define a publication channel.
2. Identify subscribers.
3. Specify the delivery transport and archive path.

For more information about each of these tasks, see *SAS Publishing Framework: Developer’s Guide*.

Publishing Items to the Enterprise

About Publishing Items to a Publishing Channel

In SAS Enterprise Guide, you can publish these items:

- analysis results
- data tables
- external files
- tasks
- reports

Note: If you are publishing a report that is in the SAS Report format, you can choose to publish the report to a report repository or a publishing channel. For more information, see “Publishing SAS Reports” on page 457.

You publish items by using the Publish to the Enterprise Wizard. The items are combined into a publishing package. In the wizard, you must complete these steps:

1. Name the package.
2. Choose a server.
3. Choose a publishing channel.
4. Specify properties and keywords for items and the package.

Step 1: Name the Package

You must specify a name for the package that you are publishing. You can also specify a description and an expiration date and time.

To name the package:

1. To publish an item, right-click the item that you want to publish in the **Project** pane or in the process flow, and select **Share** ⇨ **Publish**. The Publish to the Enterprise wizard opens.
2. Specify the title of the package that you are publishing.
3. Provide a brief description of the package. This description is optional.
4. Specify an expiration date and time for the package. This information is stored in the metadata of the package and can be read by other applications, such as the SAS Portal, that display the package.
5. Click **Next** to choose a server.

Step 2: Choose a Server

1. Before you can select the server, you must name the package.
2. Select a server that can write to the location that is specified by the channel to be used for publishing. The list shows the servers that are defined in the currently selected repository. If the transport method is email, the server that you select must be configured to send email.

   Note: If you publish a task, you must choose the same server to publish the task as the server that is used to run the task.
3. Click **Next** to choose a channel.

Step 3: Choose a Publishing Channel

1 Before you can choose a publishing channel, you must choose a server.

2 Select the channel that you want to use for publishing. The list shows the channels that are defined in the currently selected repository. The channel must be compatible with the server that you selected in the previous step.

3 Click **Next** to specify item properties and keywords.

Step 4: Specifying Properties and Keywords for Items and the Package

You can publish more than one item at a time. You can view the properties of each item and include keywords in the metadata of an item or package. These keywords can be searched for in the SAS Portal or other application that displays the published package. To further refine the search, you can also associate a value with a keyword. For example, a keyword could be "Southeast Region," and a value could be the product ID.

To specify properties and keywords for items and the package

1 Before you can specify properties and keywords for an item, you must choose a publishing channel.

2 To add an analysis result, data table, external file, or task to the list, click **Add Item**. To remove an item from the list, select the item and click **Remove Item**.

Clear the check box next to an item to exclude the item from the publish process. For example, if you have added a task to the list, you might not want to publish all of the components of the task.

3 To change the properties of an item, select the item and click **Item Properties**. From this dialog box, you can complete these tasks.

- Enter a new description in the **Description** box.
- Select a new output type from the **Publish as type** drop-down list.
- Specify the keywords for an item by clicking **Item Keywords**.

Note: You can also specify the keywords for a package by clicking **Keywords** in the wizard.

4 To specify keywords for the package, click **Keywords**.

5 Click **Finish** to publish items to the specified channel.

Publishing SAS Reports

About Publishing SAS Reports

SAS Reports are results that are created in the SAS Report format. You can specify the format of your results in the Options dialog box. For SAS tasks, you select the format of the results in the Results General pane, and for stored processes, you specify the format of the results in the Stored Process pane.

Using SAS Enterprise Guide, you can publish SAS Reports to a report repository for SAS Web Report Studio. After a report is published to this repository, you can open the report in SAS Web Report Studio. The setup of this repository is done by your SAS administrator.

As with other items in SAS Enterprise Guide, you publish a SAS Report by using the Publish to the Enterprise Wizard. However, the steps in this wizard differ depending on whether you are publishing to a repository or a publishing channel.

Publishing a Report to a Repository

Step 1: Select the Type of Publishing

You can publish a SAS Report to a report repository or to a publishing channel.

To select the type of publishing:

1 To publish a SAS Report, right-click the SAS Report that you want to publish in the **Project** pane or in the process flow, and select **Share** ⇨ **Publish**. The Publish to the Enterprise Wizard opens.

2 Select one of these options:

- **Report Repository** publishes the report to a report repository that is accessed by SAS Web Report Studio. After the report is published to a repository, you can open the report in SAS Web Report Studio. If you have access to more than one SAS Web Report Studio session, then you can select the session that you want to use from the drop-down list. If you have access to only one SAS Web Report Studio session, then the drop-down list is not displayed.

> **Note:** If you do not have access to any SAS Web Report Studiosessions, the **Report Repository** option is disabled.

- **Publishing Channels** creates a publishing package and publishes it to a publishing channel.

3 Click **Next**.

If you are publishing to a report repository, clicking **Next** enables you to specify the report location.

If you are publishing to a publishing channel, clicking **Next** enables you to name the package. For more information, see "About Publishing Items to a Publishing Channel" on page 454.

Step 2: Specify the Report Location

1 Before you can specify a name and location, you must select **Report Repository** as the type of publishing.

Provide a brief description of the report. This description is optional. Specify the keywords that can be used to describe the report. Separate multiple keywords with a comma. Keywords help search utilities identify the stored process. Keywords are optional. Click **Next** to select the type of report.

2 Specify the name and location of the report that you are publishing.

3 (Optional) Provide a brief description of the report.

4 (Optional) Specify the keywords that can be used to describe the report. Separate multiple keywords with a comma. Keywords help search utilities identify the stored process.

5 Click **Next** to select the type of report.

Step 3: Select the Report Type

When you publish a report, you can specify whether the data in the report can be refreshed after the report is published. You can also make the report read-only.

To select the type of report:

1 Specify whether the data in the published report can be refreshed from SAS Web Report Studio. You can choose from the following options:

- **Data cannot be refreshed after the report is published, similar to a snapshot report** – This option is available for all reports. If you select this option, then you cannot refresh the report from SAS Web Report Studio.
- **Data can be refreshed after the report is published** – This option is available only if the SAS Report is the output from a stored process. Because

stored processes are stored on a central server, they can be accessed by multiple clients, including SAS Web Report Studio. If you select this option, reports that are created from a stored process can be refreshed from SAS Web Report Studio.

- **Data can be refreshed for only portions of the published report** – This option is available for SAS Reports that contain output from both stored processes and tasks. You can refresh the content that is generated by the stored process, but you cannot refresh the content that is generated by the task. Click **View Details** to view the sections of the report that you will be able to refresh from SAS Web Report Studio. In the Hybrid Report Details dialog box, sections that you can refresh appear in yellow, and sections that you cannot refresh appear in gray.

2 (Optional) Select the **Make report read only** check box.

3 Click **Finish** to publish the report to the repository. You can now open the report in SAS Web Report Studio.

15

Understanding Git Integration in SAS Enterprise Guide

About Git Integration in SAS Enterprise Guide

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

In order to use the Git integration features in SAS Enterprise Guide, you must install Git on the local Windows computer where SAS Enterprise Guide is installed. For more information, see "Configure SAS Enterprise Guide for Git Integration" in *SAS Enterprise Guide and SAS Add-In for Microsoft Office: Administrator's Guide*.

You can use the Git Repositories (⎇) pane in the navigation area to access the basic Git features from within SAS Enterprise Guide. These features include cloning repositories, committing file changes, pulling and pushing files, viewing your repository history, creating and merging branches, and performing a basic differentiation between files in your local repository. SAS Enterprise Guide uses HTTPS with a user name and password to authenticate with your repository hosting service.

Note: If the Git Repositories pane is not available in the navigation area, select **View** ⇨ **Git Repositories**. The Git Repositories pane is displayed by default.

Note: If you want to track changes in programs that are embedded in a SAS Enterprise Guide project, you can use the Program History feature. For more information, see "Understanding Program History" on page 297.

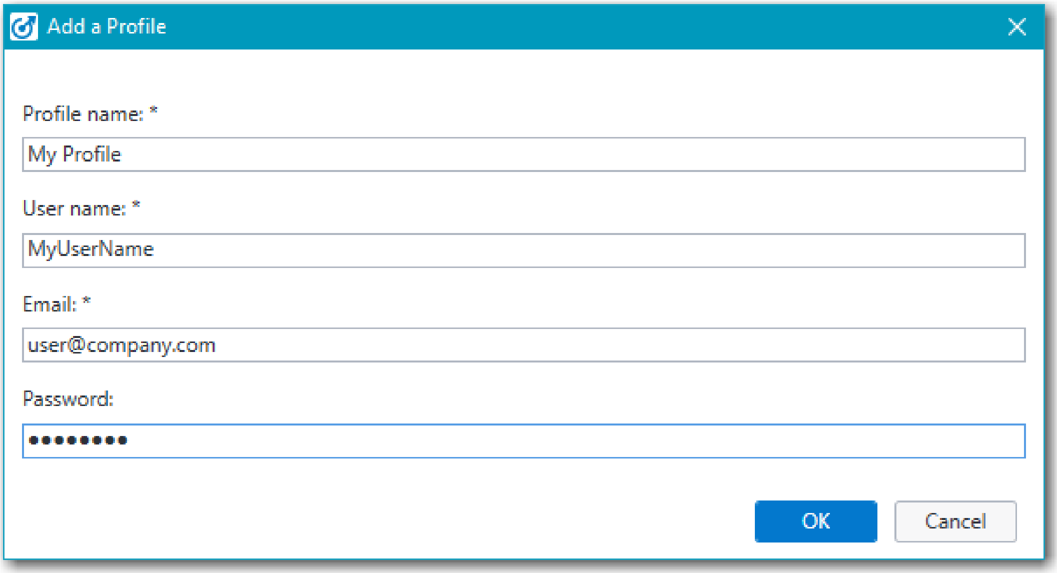
Learn more about connecting SAS Enterprise Guide to Git here:

- SAS Communities Library: How to connect SAS to your Git profile
- SAS Tutorial: How to use Git with SAS

Working with Git Profiles

Creating a Git Profile

Before you can get started using any of the Git features in SAS Enterprise Guide, you must create a Git profile. SAS Enterprise Guide uses HTTPS for authentication. To create a Git profile, select **Tools** ⇨ **Manage Git connections** and make sure that **Profiles** is selected. Click +. The Add a Profile window appears. Enter your profile name, user name, and a valid email address and password in the appropriate boxes. The user name and password are used to authenticate updates to the remote repository where your changes can be shared with other users. In many cases, the user name and password are the same ones that are associated with your GitHub account. Click **OK** to create your profile.



Profile name: * My Profile

User name: * MyUserName

Email: * user@company.com

Password: ●●●●●●●●

Editing a Git Profile

To edit an existing profile, select **Tools** ⇨ **Manage Git connections** and click **Profiles**. Select the profile that you want to edit and click ✎. The **Edit Profile** window appears so that you can make your changes. Click **OK** to save your changes.

Deleting a Git Profile

To delete a profile, select **Tools** ⇨ **Manage Git connections** and click **Profiles**. Select the profile that you want to delete and click 🗑. You cannot delete a profile that is associated with a Git repository.

Cloning and Adding Git Repositories

About Cloning and Adding Git Repositories

In order to access the files from a Git repository in SAS Enterprise Guide, you must clone the repository to a local repository that is accessible via your local file system. When you clone the repository, a copy of all the files in the remote repository are copied to a working directory that is accessible via your local file system.

If there is an existing Git repository on your file system, you can add the repository to your repositories in SAS Enterprise Guide. For example, you might have already cloned a repository using a tool such as Git Extensions. However, if the Git repository does not already exist on your file system, you must first clone the repository before you can access it.

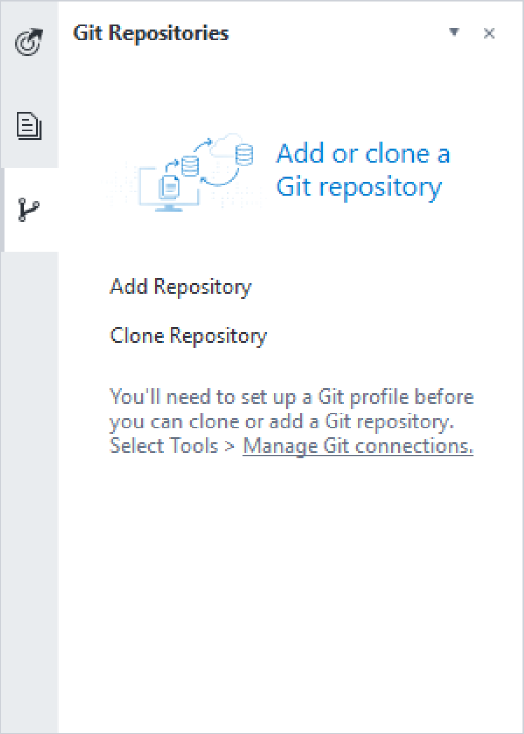
You can access the files in a repository by selecting **File** ⇨ **Open** and navigating to the directory to which you cloned or added the repository. You can make changes to the files in the working directory, and then commit those changes to your local repository.

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

Cloning a Git Repository

To clone a Git repository:

1 In the navigation area, click ⎇, and then click **Clone Repository**.



Git Repositories

Add or clone a Git repository

Add Repository

Clone Repository

You'll need to set up a Git profile before you can clone or add a Git repository. Select Tools > Manage Git connections.

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

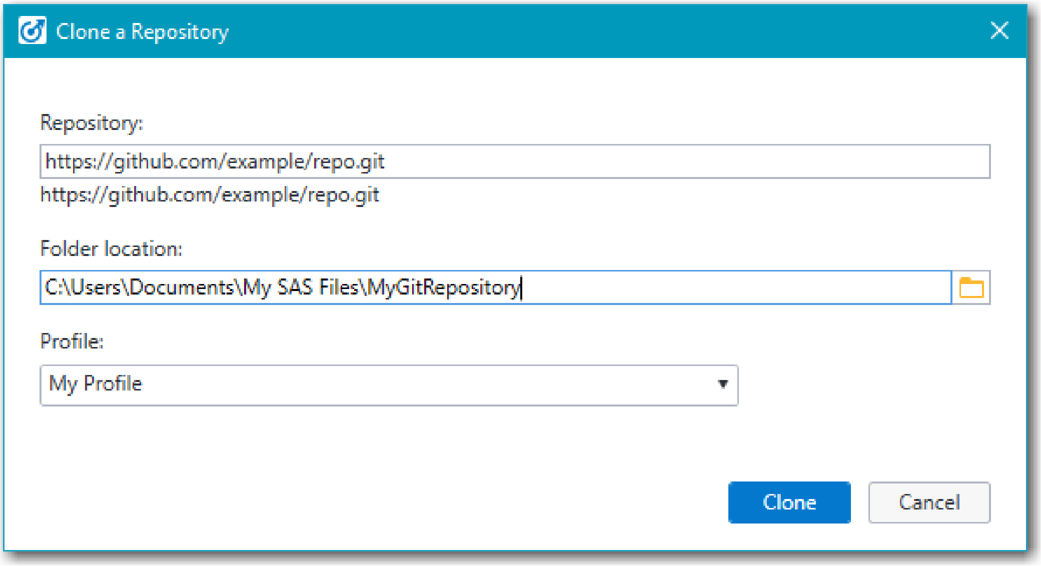
- In the Git Repositories pane, click **Clone**.
- Select **Tools** ⇨ **Manage Git connections** and click **Repositories**. Click + and select **Clone a new repository**.

2 In the **Repository** box, enter the remote repository that you want to clone. For example, your repository name might look like

`https://github.com/example/repo.git.`

3 In the **Folder location** box, enter the path name of a folder on your local file system for the cloned repository.

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



Repository:
https://github.com/example/repo.git
https://github.com/example/repo.git

Folder location:
C:\Users\Documents\My SAS Files\MyGitRepository

Profile:
My Profile

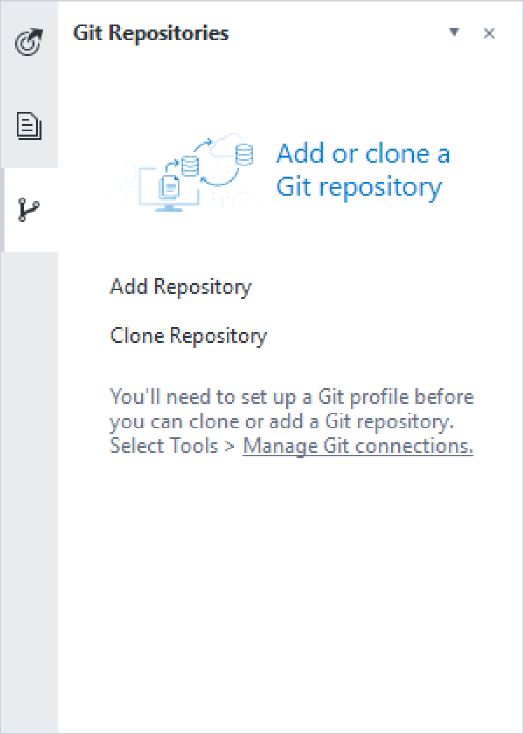
5 Click **Clone** to clone the repository. A Git repository tab opens in the work area. You can also view the files in the cloned repository by selecting **File** ⇨ **Open** and navigating to the directory in which you cloned the repository.

Adding an Existing Git Repository

You can add a local repository to SAS Enterprise Guide from a folder in your file system. After you add the repository, you can access the added repository from the Git Repositories pane and track and commit your changes. You can also associate the local repository with a remote repository and push your changes to the remote repository.

To add a local repository:

1 In the navigation area, click ⑂, and then click **Add Repository**.



Git Repositories

Add or clone a Git repository

Add Repository

Clone Repository

You'll need to set up a Git profile before you can clone or add a Git repository. Select Tools > Manage Git connections.

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

- In the Git Repositories pane, click **Add**.
- Select **Tools** ⇨ **Manage Git connections** and click **Repositories**. Click + and select **Add existing repository**.

2 In the **Folder location** box, enter the path name of a folder on your file system for the cloned repository.

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

4 Click **Add** to add the repository to SAS Enterprise Guide. A Git repository tab opens in the work area. You can also view the files in the added repository by selecting **File** ⇨ **Open** and navigating to the directory to which you cloned the repository.

Understanding the Git Repository Tab in SAS Enterprise Guide

When you clone or add a Git repository to SAS Enterprise Guide, a Git repository tab opens in the work area.

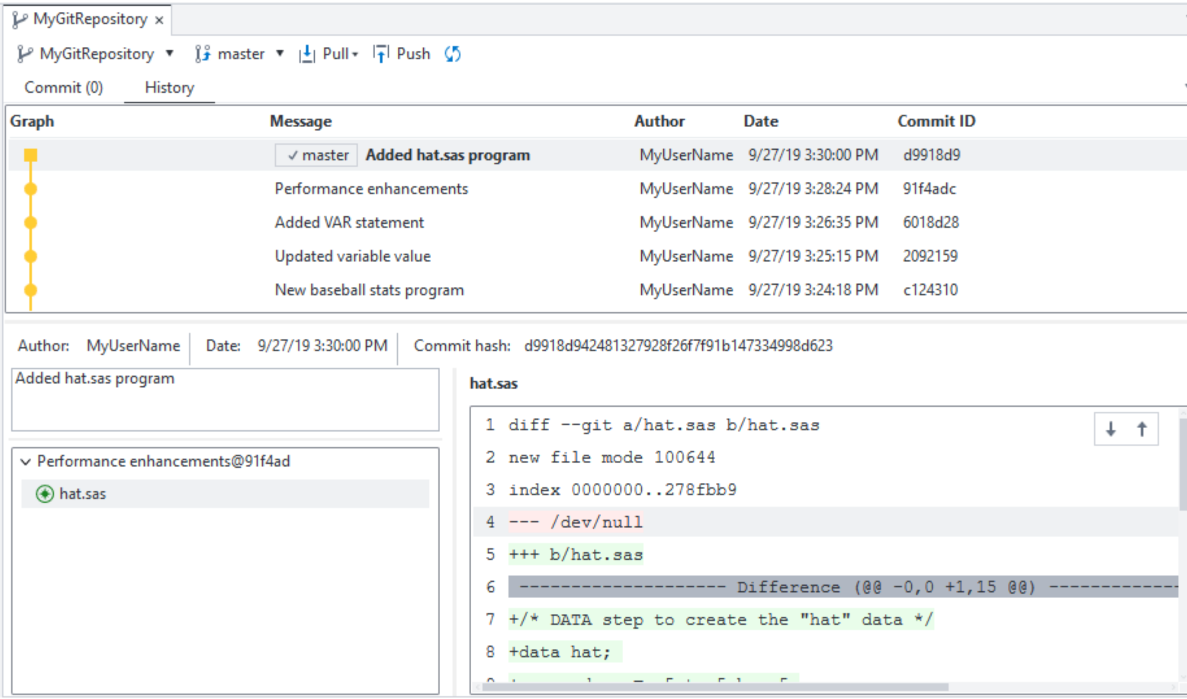
The Git repository tab contains two subtabs: History and Commit. The History subtab displays the history of all the changes, or commits, in a repository and enables you to view the differences between the selected and parent commits. The Commit subtab enables you to stage, unstage, reset, and commit changes to files in a repository. You can also view the differences between versions of a file.

To open your local Git repository tab, click ⑂ in the navigation pane and double-click the repository that you want to open. The Git repository tab opens in the work area.

Note: You can view only one Git repository in SAS Enterprise Guide at a time.

From the main Git repository toolbar, you can perform these tasks:

- Specify the local repository that you want to open.
- Check out a local branch.
- Pull or fetch data from the remote repository. For more information, see "Pulling and Fetching Files" on page 473.
- Push committed changes to the remote repository. For more information, see "Pushing Files" on page 473.
- Refresh the local repository.



Graph	Message	Author	Date	Commit ID
	✓ master **Added hat.sas program**	MyUserName	9/27/19 3:30:00 PM	d9918d9
	Performance enhancements	MyUserName	9/27/19 3:28:24 PM	91f4adc
	Added VAR statement	MyUserName	9/27/19 3:26:35 PM	6018d28
	Updated variable value	MyUserName	9/27/19 3:25:15 PM	2092159
	New baseball stats program	MyUserName	9/27/19 3:24:18 PM	c124310

Author: MyUserName | Date: 9/27/19 3:30:00 PM | Commit hash: d9918d942481327928f26f7f91b147334998d623

Added hat.sas program

Performance enhancements@91f4ad

hat.sas

hat.sas

```
diff --git a/hat.sas b/hat.sas
new file mode 100644
index 0000000..278fbb9
--- /dev/null
+++ b/hat.sas
 -------------------- Difference (@@ -0,0 +1,15 @@) ------------
+/* DATA step to create the "hat" data */
+data hat;
```

Viewing the Commit History

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

After you have saved your changes to the files in your working directory, make sure the tab for your local Git repository is open in the work area. Click the **History** subtab. A list of all the commits made in that repository is displayed in the work area in reverse chronological order. For each commit, the commit message, the author, the date and time of the commit, and the commit ID are displayed by default.

Note: By default, up to 1,000 rows are loaded in the commit history. You can specify a different limit by selecting **Tools** ⇨ **Options** and clicking **Version Control**. Enter a new value in the **Commits to load in log history** option. Increasing the number of rows that are loaded might affect your performance.

To view the details of a specific commit, select the commit from the commit history. The details of the commit are displayed in the lower section of the work area.

Committing Changes to Your Local Repository

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

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

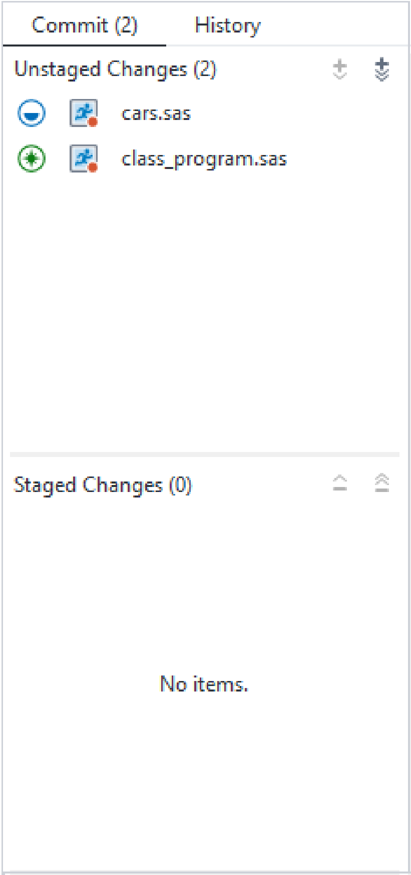
To commit changes to your local repository:

1 After you have saved your changes to the files in your working directory, make sure your local Git repository tab is open in the work area. Click the **Commit** subtab.

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

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

- ⊛ – a new file to be added to the repository
- ◎ – a renamed file
- ◒ – an existing file with changes
- ⊘ – a deleted file
- ↶ – a file with content that is very similar or identical to another file in the local repository
- ⊗ – a file with a merge conflict



Commit (2) History

Unstaged Changes (2)

cars.sas

class_program.sas

Staged Changes (0)

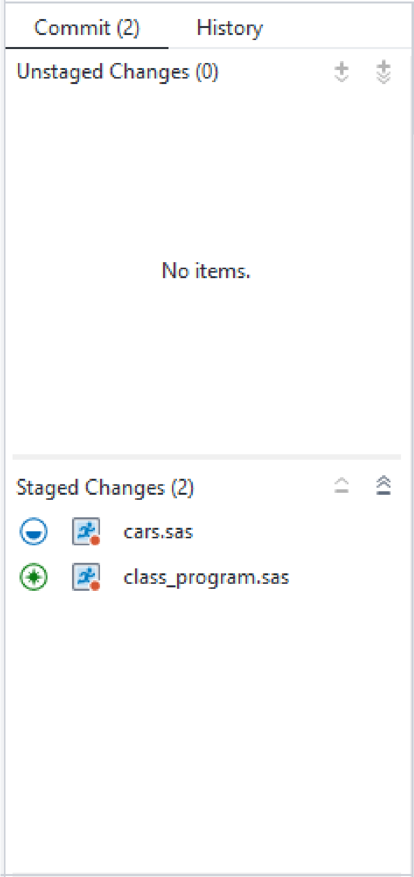
No items.

Note: Autosaved files that include ~ in the file name are not included in the list of unstaged files.

3 Select one or more changes that you want to commit and click ⇣. When you select files to commit, it is a good practice to group changes that are related and belong together in a single commit.

TIP You can also stage and unstage files by right-clicking one or more files and selecting **Stage selected** or **Stage all**.

Note: Changes that you do not stage are not included in the commit. Your changes remain in your working directory and in the Unstaged Changes area until you commit them.



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

TIP You can view the differences between the versions of a file in the staging area and your local repository by clicking the file on the **Commit** tab. To discard the changes to a file in the staging area and reset the file to the version in your local repository, right-click the file and select **Reset**. If the file that you are resetting is currently open in the work area, you are prompted to confirm whether you want to keep the current version of the file in the Program Editor or discard the changes and reload the file.

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

Pulling and Fetching Files

You can download data from the remote repository to your local repository in two ways: pulling and fetching. The pull feature enables you to download the files from the remote repository and update your local repository to make sure you have the latest content. If you have made changes to some of the files in your local repository, Git merges the content from the remote repository with your local files. To pull files from the remote repository, make sure the tab for your local Git repository is open in the work area, and then select **Pull** ⇨ **Pull**.

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

The fetch feature downloads the files from the remote repository, but fetching does not merge any changes into your current working branch. To fetch files from the remote repository, make sure the tab for your local Git repository is open in the work area, and then select **Pull** ⇨ **Fetch**.

Pushing Files

After you have staged and committed your changed files from your working directory to your local repository, you can push them to the remote repository. When you push your committed changes to the remote repository, you are synchronizing files in your local repository with the remote repository.

To push your files to the remote repository, make sure the tab for your local Git repository tab is open in the work area. Click **Push** on the toolbar. The commit history shows your committed changes merged into the remote repository.

Resetting Your Local Repository

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

To reset your local repository:

1 Make sure the commit history is open for your local repository. Right-click the commit that you want to reset and select **Reset**. The Reset Local Repository window appears.

2 Select the type of reset that you want to perform:

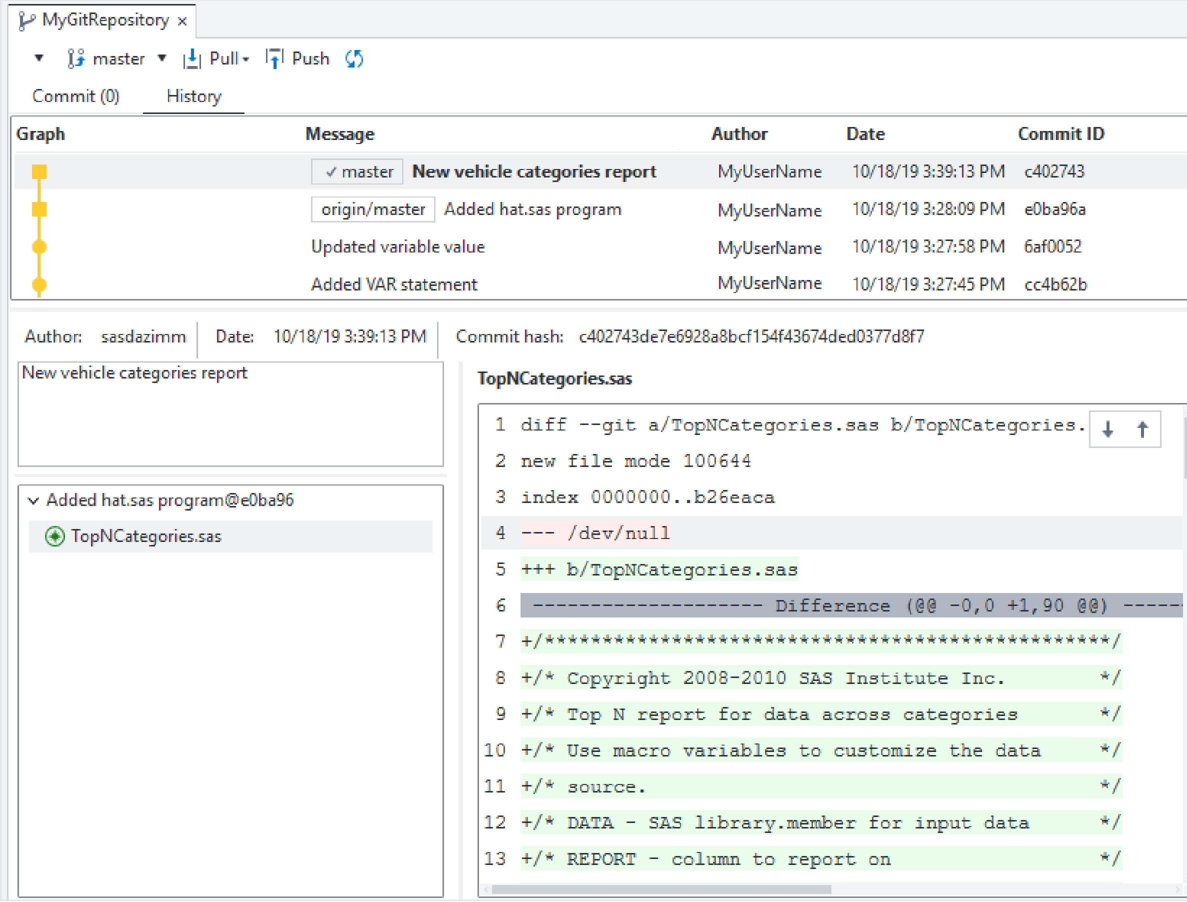
   - **Soft** – resets the current branch to the prior commit. No changes are made to the staged changes (the index) or your working directory.
   - **Mixed** – resets the current branch to the prior commit and resets your staged changes (the index). No changes are made to your working directory.
   - **Hard**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

3 Click **OK** to reset your repository.

Working with Branches in Git

Your work in Git is managed in branches. By default, the main branch in your remote repository is identified as the origin/master branch. By default, the main branch in your local repository is identified as the master branch. You can create additional branches and merge branches. When you push your committed changes to the remote repository, you are synchronizing files in a branch in your local repository with the branch in the remote repository. You can use the History subtab to see the commits that are associated with a branch and whether branches are synchronized. For more information, see “Viewing the Commit History” on page 469.

In this example, the master branch is ahead of the origin/master commit by one commit. This indicates that you have made changes to the local master branch that are not included in the remote origin/master branch because the "New vehicle categories report" commit has not yet been pushed to the remote repository. For more information, see “Pushing Files” on page 473.



Graph	Message	Author	Date	Commit ID
	✓ master **New vehicle categories report**	MyUserName	10/18/19 3:39:13 PM	c402743
	origin/master Added hat.sas program	MyUserName	10/18/19 3:28:09 PM	e0ba96a
	Updated variable value	MyUserName	10/18/19 3:27:58 PM	6af0052
	Added VAR statement	MyUserName	10/18/19 3:27:45 PM	cc4b62b

Author: sasdazimm | Date: 10/18/19 3:39:13 PM | Commit hash: c402743de7e6928a8bcf154f43674ded0377d8f7

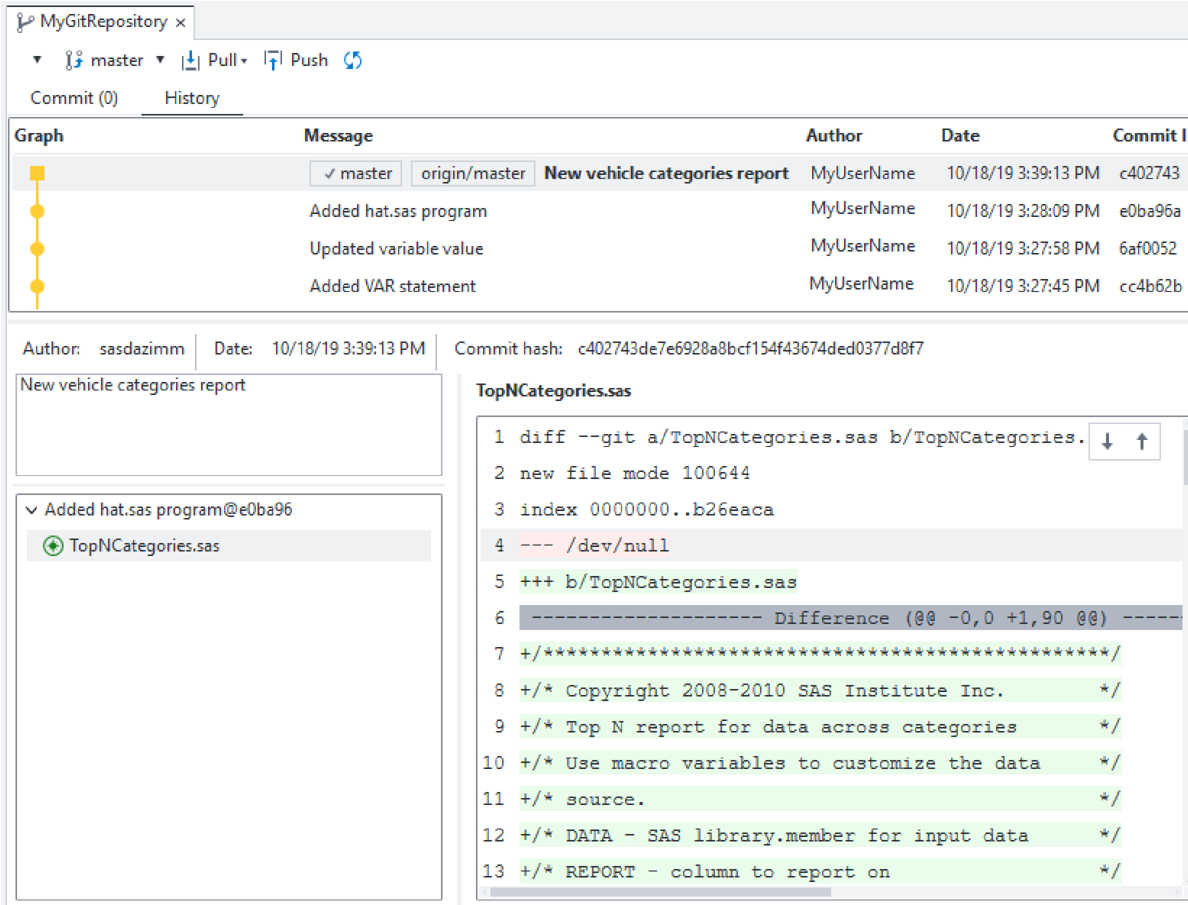
New vehicle categories report

Added hat.sas program@e0ba96
TopNCategories.sas

TopNCategories.sas

```
diff --git a/TopNCategories.sas b/TopNCategories.
new file mode 100644
index 0000000..b26eaca
--- /dev/null
+++ b/TopNCategories.sas
 -------------------- Difference (@@ -0,0 +1,90 @@) -----
+/*****************************************************/
+/* Copyright 2008-2010 SAS Institute Inc.            */
+/* Top N report for data across categories           */
+/* Use macro variables to customize the data         */
+/* source.                                           */
+/* DATA - SAS library.member for input data          */
+/* REPORT - column to report on                      */
```

After the “New vehicle categories report” commit has been pushed to the remote repository, the origin/master and master branches are synchronized and are listed on the same line in the commit history.



Graph	Message	Author	Date	Commit I
	✓ master origin/master **New vehicle categories report**	MyUserName	10/18/19 3:39:13 PM	c402743
	Added hat.sas program	MyUserName	10/18/19 3:28:09 PM	e0ba96a
	Updated variable value	MyUserName	10/18/19 3:27:58 PM	6af0052
	Added VAR statement	MyUserName	10/18/19 3:27:45 PM	cc4b62b

Author: sasdazimm Date: 10/18/19 3:39:13 PM Commit hash: c402743de7e6928a8bcf154f43674ded0377d8f7

New vehicle categories report

Added hat.sas program@e0ba96
- TopNCategories.sas

TopNCategories.sas

```
diff --git a/TopNCategories.sas b/TopNCategories.
new file mode 100644
index 0000000..b26eaca
--- /dev/null
+++ b/TopNCategories.sas
 -------------------- Difference (@@ -0,0 +1,90 @@) -----
+/*************************************************/
+/* Copyright 2008-2010 SAS Institute Inc.        */
+/* Top N report for data across categories       */
+/* Use macro variables to customize the data     */
+/* source.                                       */
+/* DATA - SAS library.member for input data      */
+/* REPORT - column to report on                  */
```

For more information about managing branches in Git, see the documentation for your repository hosting service.

Creating a Branch

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

Note: When you have multiple branches, you can check out the branch that you want to work in by using the current branch drop-down list at the top left of your Git repository tab. You can also right-click the branch in the commit history and select **Checkout branch** ⇨ ***branch-name***. For more information, see "Checking Out Branches" on page 477.

To create a branch:

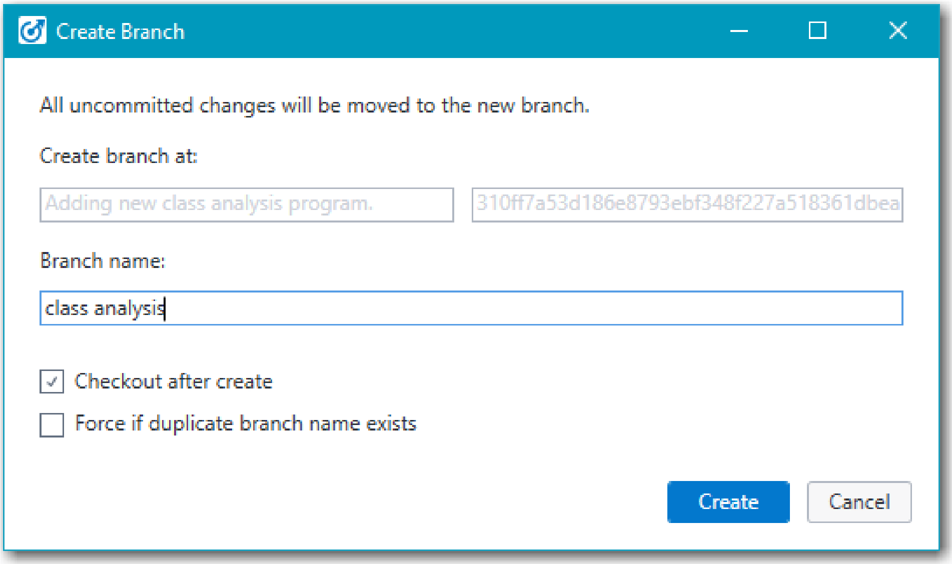
1 Make sure the commit history is open for your local repository. Right-click the commit with which you want to start a new branch and select **Create new branch**. The Create Branch window appears.

2 In the **Branch name** box, enter a name for your new branch. The branch name can contain only letters, numbers, and special characters such as hyphens or underscores.

3 If you want to create a branch but remain in the current branch, clear the **Checkout after create** check box. This option is selected by default.

4 If you want to discard any changes in your working directory that you have not committed when the branch is created, select the **Force if duplicate branch name exists** check box. This option is not selected by default.

5 Click **Create** to create the branch. Your local repository now contains the files from the commit that you selected for creating the branch.



All uncommitted changes will be moved to the new branch.

Create branch at:

Adding new class analysis program.

310ff7a53d186e8793ebf348f227a518361dbea

Branch name:

class analysis

☑ Checkout after create

□ Force if duplicate branch name exists

Checking Out Branches

To open a branch in your working directory, you must check out the branch. You can check out local branches by using the current branch drop-down list at the top of the Git tab or by right-clicking the local branch in the commit history and selecting **Checkout branch** ⇨ ***branch-name***.

If you want to work with a remote branch that does not have a corresponding local branch, you can create a local copy of that branch in your working directory in one of the following ways:

- Create a local branch with the same name as the remote branch. For more information, see "Creating a Branch" on page 476.

- Right-click the remote branch in the commit history and select **Checkout branch ⇨ *branch-name***. A local version of the branch is created and checked out.

  Note: If you check out a remote branch that already has a corresponding local branch, the remote branch is merged into the local branch.

Merging Branches

When you merge one branch into another, Git combines the changes from the commit sequences in each branch. A merge conflict occurs when Git encounters something in a file that has been changed in both branches. You must resolve any conflicts manually before the merge can be completed. After the merge is complete, you can still view the history of both branches. The existing branches are not changed in any way.

To merge a branch:

1. Make sure the commit history is open for your local repository. In the current branch drop-down list, select the branch into which you want to merge the new branch. For example, to merge *newbranch* with the master branch, you would need to check out the master branch.
2. Right-click the branch that you are merging, and select **Merge into current branch ⇨ *branch-name***. For example, to merge *newbranch* with the master branch, you would right-click *newbranch* and select **Merge into current branch ⇨ *newbranch***. The current branch in the commit history list is updated with the latest commit from the new branch.

Note: SAS Enterprise Guide displays a message if there is a conflict between files when you merge the branches. If a conflict occurs, the conflicted files are moved to the Unstaged changes area on the Commit subtab. After you resolve the conflicts, you can complete the merge by staging and committing the files.

Removing a Repository

You can remove the local repository from your Git Repositories list. When you remove a repository only the Git repository definition is deleted. The files in your working directory are not deleted.

To remove the local repository:

1 Select **Tools** ⇨ **Manage Git Connections**. Click **Repositories** and select the repository that you want to remove. Click 🗑. The Confirm Remove window appears.

2 Click **Remove** to remove the repository.

16

Customizing SAS Enterprise Guide

About Setting Options

The Options window enables you to change options that affect the entire SAS Enterprise Guide application. To set options, select the option that you want and click **OK**.

Select **Tools** ⇨ **Options** to open the Options window.

Note: The **Reset All** button resets all of the options in the Options window to their default settings.

Setting Start-Up Options

To access the Start-Up options, select **Tools** ⇨ **Options**. In the selection pane, click **Start Up**.

Show Start page	Displays the **Start Page** tab each time you start SAS Enterprise Guide. The Start Page includes options to create a program or project, select an existing file or recently viewed file to open, and access learning materials for SAS Enterprise Guide. To dismiss the **Start Page** tab as soon as you open another project or file, select **Automatically dismiss**. If you do not select this option, SAS Enterprise Guide opens with options to create a program or project or open an existing file.
Other items to display on start up	Specifies whether to open a new project or program when you start SAS Enterprise Guide.

Automatically check for updates

Specifies that SAS Enterprise Guide should prompt you when updates are available. This option is not selected by default. For more information, see “Check for Software Updates” on page 6.

Display language

Displays the current language of the SAS Enterprise Guide interface, online Help, and online tutorial. To change the language, select a new language from the drop-down list. Your choice of languages depends on the language components that were installed with SAS Enterprise Guide. You must restart SAS Enterprise Guide before the new language is applied. SAS Enterprise Guide uses this locale rather than the locale settings for the operating system.

Server session locale

Sets the locale of the workspace server session. You can use the locale for the display language for SAS Enterprise Guide, the locale on the client machine, or the locale on the server.

Note: This option does not apply to stored processes.

Setting General Options

To access the General options, select **Tools** ⇨ **Options**. In the selection pane, click **General**.

Show full details when viewing SAS Libraries in File Dialog details view

Displays additional details, such as Date Created and Date Modified, when you use the Details view in the Open Data window. This option is not selected by default. Selecting this option can slow the performance of this window.

Note: To use the Details view in the Open Data window, expand the **Views** button and select **Details**.

Enable hardware rendering

Specifies whether your graphics hardware, such as a graphics card, is used to render, or display, graphics and text in SAS Enterprise Guide. If you do not select this option, then software rendering is used, which means that the rendering takes place in the CPU. You might want to consider using software rendering if screen updates in SAS Enterprise Guide are slow. By default, this option is selected for the Microsoft Windows Vista, Windows 7, and Windows Server 2008 operating systems.

Note: If you change this option, you must restart SAS Enterprise Guide before the change takes effect.

Color theme	Specifies whether to use the light (**Illuminate**) or dark (**Ignite**) color theme for the user interface.
Number of items to display	Sets the number of recently opened projects and programs that are displayed on the Start page, in the Open window, or when you select **File** ⇨ **Open recent items**.
Show a notification when a server request exceeds threshold (minutes)	Specifies the maximum number of minutes for a server request. If the number of minutes exceeds this threshold, a message appears in the Notification bar.
Reset All Warnings	Resets SAS Enterprise Guide to display all warning messages, including those that you have selected not to be shown. This option affects warning messages in the Notification bar and any windows that open to display a warning message.

Setting Project and Process Flow Options

To access the Project and Process Flow options, select **Tools** ⇨ **Options**. In the selection pane, click **Project and Process Flows**.

When an item is opened, automatically add to the project	Automatically adds any files that you open to your current project.
Automatically open a process flow when opening a project	Automatically opens a process flow tab when you open a project.
Open a new program when creating a project	Automatically opens a new program tab when you create a project.
Automatically run "Autoexec" process flow when project opens	In a project that contains a process flow named "autoexec," SAS Enterprise Guide automatically runs that process flow when you open the project. If you do not select this option, SAS Enterprise Guide prompts you before running the

	autoexec process flow. For more information, see "Running a Process Flow When You Open a Project" on page 59.
Include project name in exported file names	Automatically includes the name of the SAS Enterprise Guide project in the name of any exported file.
Enable project log for new projects	Automatically creates an aggregated log of all activity for any projects that you create. You can view the project log from the Project Log window.
Group programs and stored process items in a separate folder	Automatically groups all stored process items together in a Stored Processes folder and all program items together in a Programs folder in the Project pane.
Display full path of data and referenced files	Displays the full pathname of all of the data objects, SAS programs, text files, and other external files such as Microsoft Word files in the Project pane. If the data is stored in a library, the pathname is displayed as *library.member*. **Note:** When you select this option, you cannot rename the objects.
Maximum number of output data sets to add to the project	Specifies the maximum number of output data sets from tasks, queries, and programs to add to your project. The default value is `50`.
Display output data as a node in the project	Specifies whether to display output data from tasks, queries, and programs as nodes in the Project pane.
Show generated code items	Specifies whether to display an object in the process flow for code that was submitted during the last run of the process flow. The Last Submitted Code object is not displayed by default. **Note:** If you do not select this option, you can still view the code that was submitted during the last run by right-clicking the task or code object and selecting **Open** ⇨ **Last Submitted Code**. The code is opened in a separate tab in the workspace. A code object is not added to the process flow.
Show SAS output log	Specifies whether to display an object for the log file that was generated when the task was run. The log object is not displayed by default. **Note:** If you do not select this option, you can still open the log object by right-clicking the task object and selecting **Open** ⇨ **Log**. The log is opened in a separate tab in the workspace. A log object is not added to the process flow.

Show generated results	Displays an object for the results that were generated when the task was run. The results are displayed by default. **Note:** If you do not select this option, you can still open the results by right-clicking the task object and selecting **Open** ⇨ **Results**.
Display full path of data and referenced files	Displays the full pathname of all of the data items, SAS programs, text files, and other external files (such as Microsoft Word files) in the process flow. If the data is stored in a library, the pathname is displayed as *library.member*. **Note:** When you select this option, you cannot rename the items.

Setting Auto Recovery Options

To access the Auto Recovery options, select **Tools** ⇨ **Options**. In the selection pane, click **Auto Recovery**.

Enable auto recovery	Enables SAS Enterprise Guide to automatically save a copy of the active project to the specified backup location at the specified interval. If SAS Enterprise Guide closes unexpectedly, you can recover the project as it was at the last time it was automatically saved. This option is selected by default.
Backup interval	Specifies the interval at which the active project is automatically saved. The default interval is five minutes.
Backup location	Specifies the location in which to automatically save backup copies of the active project. You can enter the path in the box or click **Browse** to find the location. You cannot save the backup files on a remote server; the files must be saved on your local computer.
Update external items	Saves backup copies of any linked items, such as an external SAS program file, with the backup file in the location that you have specified. Linked items are referred to in the project, but the contents of those items are not included in the project file. When you restart SAS Enterprise Guide and choose to open or save the recovered project file, you are also prompted to save each linked file. This option is selected by default.

Setting Results Options

Setting General Results Options

To access the general results options, select **Tools** ⇨ **Options**. In the selection pane, click **General** under the Results heading.

HTML	Generates results in HTML format. This is the default results format.
SAS Report	Generates the results as a SAS Report. SAS Report results can be used to create a report.
Listing	Generates results in text format.
Excel	Generates results in Microsoft Excel format. This format can be generated by any SAS server that is running SAS 9.4M3 or later.
PDF	Generates results in PDF format.
PowerPoint	Generates results in Microsoft PowerPoint format. This format can be generated by any SAS server that is running SAS 9.4M3 or later.
RTF	Generates results in RTF format.
Default format	Specifies how the results from a task or program are displayed in the workspace. Results are generated in all of the selected formats. The default format is displayed as the active tab in the workspace.
Default tab to display	Specifies which tab to display after you run a task, query, or program. Results are displayed by default. Use the **If default doesn’t exist, display** option to specify another tab to display if the selected tab does not exist.
Move focus to the tab that is displayed	Automatically moves the focus to the tab that is displayed, such as when you open a file or generate results and output data.

Display SAS log when errors occur	Displays the SAS log (instead of the default tab) if the SAS program generates errors while running.
Embed handcoded ODS results in EG project	Embeds results that are created by your customized ODS statements in your SAS Enterprise Guide project. This option is not selected by default. If you select this option, these additional results are downloaded by SAS Enterprise Guide from the server on which they are created and stored in your project. This action can affect your performance if the results files are large.
Change task icon when warnings occur	Displays a WARNING overlay on the task icon in the Project pane or process flow if there is a warning in the SAS log.
Show generated wrapper code in 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.
Stream log updates while a procedure is running	Displays log updates as a procedure is processed. If you do not select this option, the log is displayed when the procedure has finished running. This option is selected by default and might have a slight effect on performance.
Prompt before opening results larger than	Displays a confirmation window when you attempt to open a results file on a remote server that is larger than n megabytes (MB). The default value is 5 MB.
Specify the page size for log and text output	Uses the OPTIONS statement to specify the page width, or length of a line in the output, and the page height, or the number of lines per page in the output, for any log and text output.

Setting Viewer Options

To access the viewer options, select **Tools** ⇨ **Options**. In the selection pane, click **Viewer** under the Results heading.

HTML Viewer	■ **SAS Enterprise Guide** – Uses SAS Enterprise Guide to display HTML task results. ■ **Windows default** – Uses your HTML default viewer to display HTML task results. ■ **Preferred viewer** – Uses the viewer of your choice to display HTML task results. Click **Browse** to select your viewer.

RTF Viewer	■ **Windows default** – Uses your RTF default viewer to display RTF task results. ■ **Preferred viewer** – Uses the viewer of your choice to display RTF task results. Click **Browse** to select your viewer.
PDF Viewer	■ **Windows default** – Uses your PDF default viewer to display PDF task results. ■ **Preferred viewer** – Uses the viewer of your choice to display PDF task results. Click **Browse** to select your viewer.
File extensions to open as text by default	Displays the file types that are opened by default in the SAS Enterprise Guide text editor instead of the external associated Windows application. Note: Any changes that you make to the list of file types in this option affect only files that are added to your project after you make the changes. Files that have already been added to the project are not affected.

Setting HTML Results Options

To access the HTML results options, select **Tools** ⇨ **Options**. In the selection pane, click **HTML** under the Results heading.

Use HTML5	Generates the results in HTML5 format. To embed the style sheet in the HTML5 results so that your results are formatted correctly when you are not connected to the server, select **Embed style sheet from server**.
Use HTML	Generates the results in HTML4 format. To embed the style sheet in the HTML results so that your results are formatted correctly when you are not connected to the server, select **Embed style sheet from server**.
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.
Encode character set	Includes the ENCODING= option in the ODS statement that is in the code that SAS Enterprise Guide automatically generates. The ENCODING= option is used to specify the character-set encoding that is used when the results are created. The default value is UTF8 for HTML results. If you disable this option, SAS Enterprise Guide uses the default character-set encoding for the server on which the results are created.

Additional options for ODS HTML statement	Enables you to specify options to be added to the ODS HTML statement. For more information, see "ODS HTML" in *SAS Output Delivery System: User's Guide*.

Setting SAS Report Results Options

To access the SAS Report results options, select **Tools** ⇨ **Options**. In the selection pane, click **SAS Report** under the Results heading.

Embed style sheet from server	Embeds the style sheet in your results so that your results are formatted correctly when you are not connected to the server.
Style	Displays the style that is applied to results in the SAS Report format. To change the style that is applied to the results, select another style from the drop-down list.
Manage Styles	Opens the Style Manager window so that you can create a style or edit an existing style.
Additional options for ODS SASREPORT statement	Enables you to specify options to be added to the ODS SASReport statement. For more information, see ODS Tagset Statement.

Setting RTF Results Options

To access the RTF results options, select **Tools** ⇨ **Options**. In the selection pane, click **RTF** under the Results heading.

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.
Columns	Specifies the number of page columns for multiple column outputs.
Minimize page breaks within a table	Prevents a table from being split by a page break unless the entire table cannot fit on one page. If you do not specify this option, then a table can be split by a page break.

Author	Enables you to specify an author of the file.
Title	Enables you to specify a title for the file.
Encode character set	Includes the ENCODING= option in the ODS statement that is in the code that SAS Enterprise Guide automatically generates. The ENCODING= option is used to specify the character-set encoding that is used when the results are created. If you disable this option, SAS Enterprise Guide uses the default character-set encoding for the server on which the results are created.
Additional options for ODS RTF statement	Enables you to specify options to be added to the ODS RTF statement. For more information, see "ODS RTF" in *SAS Output Delivery System: User's Guide*.

Setting PDF Results Options

To access the PDF results options, select **Tools** ⇨ **Options**. In the selection pane, click **PDF** under the Results heading.

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.
Columns	Specifies the number of page columns for multiple column outputs.
Color	Uses all of the color information that the style template provides.
Add Bookmarks	Automatically creates bookmarks, or a table of contents, in the PDF file.
Author	Enables you to specify an author of the file.
Keywords	Enables you to associate keywords with the file that can be used in a search.
Subject	Enables you to specify a subject for the file.
Title	Enables you to specify a title for the file.
Create accessible PDF	Adds accessibility metadata to the PDF file that enables the file to be accessed by assistive technology such as a screen reader.

	When metadata is added, the file is often called a "tagged PDF" and follows the PDF/Universal Accessibility (PDF/UA) format.
Additional options for ODS PDF statement	Enables you to specify options to be added to the ODS PDF statement. For more information, see "ODS PDF" in *SAS Output Delivery System: User's Guide*.

Setting Excel Results Options

To access the Excel results options, select **Tools** ⇨ **Options**. In the selection pane, click **Excel** under the Results heading.

For more information about these options, see "ODS Excel Statement" in *SAS Output Delivery System: User's Guide*

Style	Displays the style that is applied to results in Excel. To change the style that is applied to the results, select another style from the drop-down list.
Author	Specifies the author of the Excel document.
Category	Specifies the category of the Excel document.
Comments	Enables you to add comments to the properties of the Excel document.
Keywords	Enables you to add keywords to the Excel document properties that can be used in a search.
Status	Specifies the status of the Excel document.
Title	Specifies a title for the Excel document.
Frozen Headers	Specifies whether the headers can scroll with the scroll bar. You can enter **YES**, **NO**, or the number of the header row that does not scroll.
Sheet Interval	Specifies the criteria for when a worksheet is created.
Frozen Rowheaders	Specifies whether the row headers on the left scroll when the table data scrolls. You can enter **YES**, **NO**, or the number of columns to freeze.
Rowbreak Interval	Controls the placement of page breaks. This option can place a page break after each output object or after each procedure.

Autofilter	Turns on filtering for specified columns in the worksheet.
Sheet Name	Specifies the criteria for when a worksheet is created.
Embedded Titles	Specifies whether titles should appear in the worksheet.
Start At	Specifies a starting cell for the report. The default is to start at column 1 and row 1.
Embedded Footnotes	Specifies whether embedded footnotes should appear only at the bottom of the worksheet.
Print Header	Specifies the text that is placed in the header when printing. If no title is specified, this text is used by Excel on the printed page. If a title is specified with the TITLE statement, that title is used.
Print Footer	Specifies the text that is placed in the footer when printing.
Additional options for ODS EXCEL statement	Enables you to specify options to be added to the ODS EXCEL statement. For more information, see "ODS EXCEL" in *SAS Output Delivery System: User's Guide*.
Additional tagset options for ODS EXCEL statement	Enables you to specify tagset options to be added to the ODS EXCEL statement. For more information about the ExcelXP tagset, see "ODS Tagset" in *SAS Output Delivery System: User's Guide*.

Setting PowerPoint Results Options

To access the PowerPoint results options, select **Tools** ⇨ **Options**. In the selection pane, click **PowerPoint** under the Results heading.

Style	Displays the style that is applied to results in PowerPoint. To change the style that is applied to the results, select another style from the drop-down list.
Author	Specifies the author of the PowerPoint document.
Category	Specifies the category of the PowerPoint document.
Comments	Enables you to add comments to the properties of the PowerPoint document.

Keywords	Enables you to add keywords to the PowerPoint document properties that can be used in a search.
Status	Specifies the status of the PowerPoint document.
Title	Specifies a title for the PowerPoint document.
Additional options for ODS POWERPOINT statement	Enables you to specify options to be added to the ODS POWERPOINT statement. For more information, see "ODS POWERPOINT" in *SAS Output Delivery System: User's Guide*.

Selecting Graphics Output Formats

In SAS Enterprise Guide, you can also choose from these formats: ActiveX, ActiveX image, GIF, JPEG, PNG, Java, Java image, SAS EMF, and SVG.

The ActiveX and Java output formats are interactive. You can right-click any graphics that are generated in these output formats and change certain options. The options that you can change are different for the ActiveX and the Java output formats. Any options that you change are reflected in the output only, not in the task window selections that you made to create the graph.

Note: ActiveX and ActiveX image output formats are not available if you are connected to a SAS Viya environment or to a server that is running SAS 9.4M8 or later.

All other output formats are noninteractive. You cannot change the appearance of the graph after you generate it.

Setting Graph Results Options

To access the results graph options, select **Tools** ⇨ **Options**. In the selection pane, click **Graph** under the Results heading.

Note: ActiveX and ActiveX image output formats are not available if you are connected to a SAS Viya environment or to a server that is running SAS 9.4M8 or later.

Graph Format	■ **Default** – Uses the default graph format that is associated with the format of the results.

- **PNG** – Generates the graph as a PNG file. The PNG output format is noninteractive, so you cannot change the appearance of the chart after you generate it.
- **ActiveX** – Generates the graph as an ActiveX control. This format can be viewed only on computers with Microsoft Windows installed. The ActiveX control output format is interactive. You can right-click any graphic output that is generated in this format and change the chart type and many other options. Any options that you change are reflected in the output only, and not in the task window selections that you made to set up the chart.

  You can specify the URL for the ActiveX control for the version of SAS that you are running. You can also specify the location of the text maps (GIF or JPEG files) to use for bar, pie and donut charts.
- **Java** – Generates the graph as a Java applet. The Java output format is interactive. You must specify the URL for the Java applet JAR files.
- **GIF** – Generates the graph as a GIF file. The GIF output format is noninteractive; you cannot change the appearance of the chart after you generate it.
- **JPEG** – Generates the graph as a JPEG file. The JPEG output format is noninteractive; you cannot change the appearance of the chart after you generate it.
- **ActiveX image** – Generates the graph as a PNG file by using ActiveX technology. This format can be generated only by SAS on Windows. The PNG output format is noninteractive; you cannot change the appearance of the chart after you generate it.
- **Java image** – Generates the graph as a PNG file by using Java technology. The PNG output format is noninteractive; you cannot change the appearance of the chart after you generate it.
- **SAS EMF** – Generates the graph as a Microsoft Enhanced Metafile.
- **SVG** – Creates SVG graphs instead of PNG graphs in HTML output. SVG graphs maintain clarity when you zoom in and out.

For many graph formats, you can specify these options:

- **Include titles inside the image** – Indicates whether to include the titles as an embedded part of the image. By default, the title is not included as a part of the image and the NOGTITLE option is included in the SAS code. When the title is not a part of the image, you can copy and paste the title text in the graph output. However, you have less control over the font settings that are set by the server. If you select this option, the GTITLE option is included in the SAS code. How the title

	is rendered depends on the graphics output format that you are using. - **Include footnotes inside the image** – Indicates whether to include the footnotes as an embedded part of the image. By default, the footnote is not included as a part of the image and the NOGFOOTNOTE option is included in the SAS code. When the footnote is not a part of the image, you can copy and paste this text in the graph output. However, you have less control over the font settings that are set by the server. If you select this option, the GFOOTNOTE option is included in the SAS code. How the footnote is rendered depends on the graphics output format that you are using.
Best fit (let SAS decide)	Allows SAS to determine the best dimensions for the graph.
Use the following (in pixels)	Enables you to specify the width and height in pixels of the generated graph. The minimum value is 2.
Create accessible graphs when possible	Produces graphs that can be read by screen readers. A hyperlink and data table that describes the chart and its values are included with the graph. This functionality is not supported by all graph types such as PDF or SAS Report.
Suppress graph option statements in generated code	Enables you to suppress the GOPTIONS statement that is automatically added to the SAS code that SAS Enterprise Guide generates. You might want to select this option if you do not have SAS/GRAPH installed on your SAS server. If you select this option, the values for the **Default Graph Dimensions** option and the **Create accessible graphs when possible** option are overridden. If you do not have SAS/GRAPH installed on your SAS server, the GOPTIONS statement generates an error in the log file.

Setting Stored Process Options

To access the stored process options, select **Tools** ⇨ **Options**. In the selection pane, click **Stored Process** under the Results heading.

HTML	Generates results in HTML format. This is the default format.
SAS Report	Generates results in a SAS report.
PDF	Generates results in PDF format.

RTF	Generates results in RTF format.

Setting Data Options

Setting General Data Options

To access the general data options, select **Tools** ⇨ **Options**. In the selection pane, click **General** under the Data heading.

Display columns in alphabetical order within task windows and the Query Builder	Within the task windows and the Query Builder, displays the columns of a data table in alphabetical order instead of the order in which they appear in the table. This order does not affect how your tasks are processed or how the columns appear in your results or the data grid.
Use data in unprotected (read/write) mode	Turns off protected mode when you insert data in your project. If you do not select this option, data is inserted in Read-Only mode by default. Note: Data tables that SAS Enterprise Guide cannot edit cannot be unprotected and are therefore always inserted in Read-Only mode. For example, SAS Enterprise Guide cannot edit data sources that it reads by using the ADO (Active Data Objects) interface. Examples of these types of data sources include Microsoft Excel files, Microsoft Access database files, text files, and Paradox database files.
Always prompt when closing a project with temporary data references	In a project that refers to a temporary data source, such as a data set in the Work library, SAS Enterprise Guide informs you that your project includes temporary data and that this data is lost when you close the project. When you save a project, all references to temporary data sources are saved, but the temporary data is deleted from the SAS server. If you do not select this check box, SAS Enterprise Guide does not display this informational prompt. Note: If you save your project with a data item that refers to a temporary data source, that data source might not be available the next time you access your project. However, if your project includes tasks that create these data sources as part of the task output, these files are re-created when you rerun the task or project.

Always obtain the total record count for DBMS tables

Determines the total record count for a DBMS (database management system) table when you open it. If you are accessing a very large DBMS table, selecting this option could increase the time that it takes you to open the table.

Note: This option affects tables from databases including Oracle, DB2, Informix, CA-OpenIngres, Sybase, Teradata, ADABAS, CA-IDMS, CA-Datacom/DB, IMS-DL/I, Oracle Rdb, SAP R/3, and SQL/DS.

Auto-size column widths

Automatically adjusts the width of the columns in the data grid to fit the size of the column content.

Default number of columns

Specifies the number of columns that you want in the data grid when you are creating data. The default value is `6`.

Default number of rows

Specifies the number of rows that you want in the data grid when you are creating data. The default value is `12`.

Clear cache

Clears any cached SAS format and informat information from your computer. When you clear the cached formats and informats, they are refreshed from the appropriate SAS server the next time you access them.

Use labels for column names

Displays the column label, rather than the column name, in the data grid, in the Query Builder, in the Data Explorer, and on the Data page of the tasks. A similar option to use labels in task results exists for some tasks.

Note: To add labels to columns, right-click the column heading and select **Properties**.

Valid variable names

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

- **Extended variable names (ANY)** – Specifies that the variable names can begin with or contain any characters, including blanks, must contain at least one character, and cannot contain any null bytes. Variable names can contain mixed-case letters as well as special and multi-byte characters. Names can be up to 32 bytes in length. This option is selected by default.
- **Basic variable names (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. Leading blanks are preserved, but trailing blanks are ignored.
- **Basic, upper-cased variable names (UPCASE)** – Specifies that the variable name follows the same rules as Basic variable names, except that the variable name is uppercase.

- **Use setting from server** – Uses the VALIDVARNAME= system option that is specified on the SAS server.

Valid member names

Enables you to specify one of the following sets of rules to apply to SAS data set names, SAS data view names, and item store names:

- **Extended member names (EXTEND)** – Specifies that names must contain at least one character (letters, numbers, valid special characters, and national characters), cannot begin with a blank or a period, and cannot include a null byte. Names can contain mixed-case letters and can be up to 32 bytes in length. Leading blanks are preserved, but trailing blanks are ignored. This option is selected by default.
- **Basic member names (COMPATIBLE)** – Specifies that the names must begin with a letter of the Latin alphabet (A–Z, a–z) or the underscore character and cannot contain blanks or special characters except for the underscore. Names can contain mixed-case letters and can be up to 32 characters in length.
- **Use setting from server** – Uses the VALIDMEMNAME= system option that is specified on the SAS server.

Setting Data Performance Options

To access the general data options, select **Tools** ⇨ **Options**. In the selection pane, click **Performance** under the Data heading.

Format data locally if applicable

Uses a static format table on the client computer to format the data values. Formatting the data locally can improve performance. However, it might not include the newest formats or user-defined formats. If this option is not selected, then SAS Enterprise Guide uses the SAS server to format the values. By default, this option is not selected.

When importing data, the width for character based columns should be

- **Longest value found within the data** – Sets the column width to the actual maximum column width of the longest data value. If you do not select this option, the default column width is set to 255 bytes. This option affects how Microsoft Excel, HTML, and text files are saved to a SAS data set. This option does not affect how data is displayed in the data grid.
- **Set to 255 bytes** – Sets the column width to 255 bytes.

Maximum number of rows to scan to

Specifies the number of rows that you want to display as a preview, or sample, in the Import Data task. This option also

determine data attributes when importing data

limits the number of rows that the Import Data task reads from a text file in order to determine the format of the data. If the format of the rows that occur after the rows that are scanned as a preview differs from the format of the rows that are scanned, the Import Data task might not import your data in the format that you are expecting. If you have trouble importing your file, try increasing the limit.

Note: If you select **No limit** or set the limit to a large number, this decreases the performance of the Import Data task when you are importing large files.

Maximum number of rows to display in the data grid with SAS/ACCESS data

Specifies the number of rows that you want to display in the data grid when you are viewing data by using a licensed SAS/ACCESS interface to a database. The default value is 10,000.

Note: This option is intended to improve performance when you are opening very large files. If you select **No limit** or set the limit to a large number, this decreases the performance of the data grid when you are opening very large files.

When output tables are generated, automatically open

Specifies whether to open the last table, the first table, or no table after you run a program that generates multiple tables. This option applies only the first time that the output tables are generated.

Unlock open data after inactivity (minutes)

Specifies how long SAS Enterprise Guide waits before releasing the lock on an idle open data set. A data set becomes idle if you have had no recent interaction with it, including mouse moves, mouse clicks, edits, refreshes, or scrolling. The default value is `3 minutes`.

Unlock all open data before running code

Releases the locks on all open data sets when you run a SAS program, regardless of whether the data sets are being modified by the program. If you do not select this option, only the data sets that are modified by the program are unlocked.

Automatically refresh open data after running code

Refreshes the open data after the code runs.

Setting Query Options

To access the Query options, select **Tools** ⇨ **Options**. In the selection pane, click **Query**.

Validate query whenever changes are made

Automatically validates your query each time you apply or save changes to it. This option is selected by default. You might want to turn this option off if you are working with a database that has a limited number of available licenses. Turning this option off reduces the number of simultaneous connections that SAS Enterprise Guide makes to the database. Contact your system administrator for specific information about your site.

Note: If you turn this option off, you run the risk of submitting queries that are invalid.

Attempt to join tables in a query

Enables the Query Builder to automatically look for columns in the two tables that have the same name and type. The Query Builder also assumes that you are joining on equal values of the joining columns. If no column name and type matches are found, you receive a warning that you should specify the join manually. If you do not select this option, you must specify every join manually.

Add columns from input tables to result set of query

Includes all columns from all input tables in the output. To include all of the columns from an input table, this option must be selected before you add the table to the query. If you add a table to a query and then select this option, none of the columns are added. By default, the output table includes only those columns that you used in the query.

Apply formats to summarized columns when appropriate

Automatically applies the format of the column that is being summarized to the summarized column if the summary function is one of the following: **Sum**, **Sum Distinct**, **Avg**, **Avg Distinct**, **Min**, **Max**, or **Range**. If the format of the original column is not wide enough to display the summarized column, SAS Enterprise Guide attempts to fit the value in the format by omitting commas, decimal points, dollar signs, or by displaying the value in scientific notation.

Convert alias and label definitions to a standard SAS identifier

Automatically converts all spaces and special characters in alias names and labels to underscores.

Surround query expressions with parentheses

Automatically encloses each filter expression in parentheses so that each expression is evaluated as a single condition. You might want to select this option if you are creating an advanced filter expression, such as (t1.Age > 14 OR t1.Sex = 'F'), that contains more than one condition. If you do not select this option, then this expression is evaluated as two conditions. For example, suppose that you have created a filter with the following filter expressions:

```
t1.Age > 12 AND
t1.Sex = 'F' OR t1.Age = 14 AND
t1.Age < 15
```

If you do not select this option, each condition is evaluated individually as follows:

```
t1.Age > 12 AND
t1.Sex = 'F' OR
t1.Age = 14 AND
t1.Age < 15
```

If you select this option and each expression is enclosed in parentheses, the conditions are evaluated as follows:

```
(t1.Age > 12) AND
(t1.Sex = 'F' OR t1.Age = 14) AND
(t1.Age < 15)
```

Maximum number of rows to process for distinct value requests

Specifies the number of rows that are processed when distinct values are retrieved from a data set. By default, SAS Enterprise Guide processes 100,000 rows when you request distinct values. If you set the limit to a very large number, it allows more rows to be processed. However, this decreases the performance of the Query Builder when you are requesting distinct values from a large table. If you do not select this option, all rows are processed.

Note: This option is not supported by views.

Number of distinct column values to get for each request

Specifies the number of distinct values to get when you display a list of column values in the Query Builder or the Filter and Sort window. You display column values in the Query Builder when you are creating a basic filter that is based on a distinct value of a column or are recoding values in the query results. You display column values in the Filter and Sort window when you are creating a simple filter. The default value is `250`.

Number of rows to process in preview results window

Specifies the number of rows to process as input and the number of rows to display as output when you preview the results of your query. This option affects both the Query Builder and the Filter and Sort window. The default values are 1,000 input rows and 25 output rows.

Do not generate column aliases longer than

Limits the length of any automatically generated aliases in the Query Builder. For example, an alias is automatically generated when you add a column to a table that already has a column of the same name. Some database engines, such as Informix, work more efficiently when the length of alias names is limited. The default value is `32`.

Save query result set as

Specifies in what format query results are saved. You can choose to save query results in any one of three formats:

- **Data table** – Saves query results as a static data table against which you can run SAS Enterprise Guide tasks. Query results in this format are not updated until you rerun the query. If you do not specify the library and file name, the results are saved to the Work library.

	■ **Data view** – Saves query results as a dynamic data view against which you can run SAS Enterprise Guide 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. If you do not specify the library and file name, the results are saved to the Work library. ■ **Report** – Saves the query results as a report in the format that you have specified as your default: SAS Report, HTML, PDF, RTF, or text. Query results in this format are not updated until you rerun the query. You cannot run SAS Enterprise Guide tasks against query results in this format.
Reset Query Builder window settings	Resets all of the Query Builder windows to their default size and location.

Setting OLAP Data Options

To access the OLAP data options, select **Tools** ⇨ **Options**, and then click **OLAP Data** in the selection pane.

Note: The OLAP data options do not take effect until you open a new cube or re-open the current cube by using the Open OLAP Cube window or the Servers window.

Default Actions for Opening OLAP Cubes	Specifies whether you can specify the members to display in the rows and columns when the OLAP cube opens. ■ **Use default view** – Opens the OLAP cube and displays its default view. ■ **Always open with options** – Automatically opens the Options window when you open an OLAP cube so that you can specify the members that you want to display in the rows and columns.
View Synchronization	Specifies whether the cube views are synchronized. ■ **Synchronized** – Automatically synchronizes all of the cube views in the OLAP Analyzer so that if you make a change to one view, such as expanding, drilling, or filtering, all of the other views are updated as well. The cube views are synchronized by default when you open the OLAP Analyzer.

	- **Not synchronized** – Desynchronizes all the cube views in the OLAP Analyzer so that you can make modifications to one view without affecting any of the other views. For more information, see “Synchronizing the Cube Views” on page 133.
View Details Area	Specifies whether to display the View Details area when you open an OLAP cube. - **Fully collapsed** – Displays the View Details area as collapsed when you open an OLAP cube. - **Expanded** – Displays the View Details area as expanded when you open an OLAP cube. By default, the View Details area is displayed as expanded when you open the OLAP Analyzer.
Other View Options	For more information about other view options, see “Setting the OLAP Viewer Options” on page 188.

Setting Task Options

Setting General Task Options

To access the general task options, select **Tools** ⇨ **Options**. In the selection pane, click **General** under the Tasks heading.

Default title text for task output	Displays the title that is automatically generated with task output. The title can contain text and macro variables that are evaluated when the task is run. You can enter up to 10 lines of text.
Default footnote text for task output	Displays the footnote that is automatically generated with task output. The footnote can contain text and macro variables that are evaluated when the task is run. You can enter up to 10 lines of text.
Display all generated SAS code in task output	Displays code for additional parameters such as the current OPTIONS statements, ODS specifications, and LIBNAME statements. By default, only the task-specific code is displayed in the Code window.
Use PROC SQL instead of	Uses PROC SQL to sort the data in tasks instead of using PROC SORT. PROC SORT is used by default. In some cases when the data in the task includes large DBMS (database management

PROC SORT for sorting in tasks

system) tables, such as ORACLE or DB2 tables, it is more efficient to use PROC SQL.

Include SAS procedure titles in results

Includes the name of the SAS procedure that is associated with the task in the heading of the task results.

Note: This option is not available for all SAS procedures.

Filter unused variables

Is a performance option that enables you to control whether the unused variables are included in the output data set. Filtering the unused variables can help improve performance. The following options are available:

- **Auto** – SAS Enterprise Guide determines whether unused variables are included as part of the input data. This is the default and the recommended setting.
- **Always** – Always filters the unused variables from the input data. If an output data set is created as a result of running a task, the unused variables are also filtered from the output data set.
- **Never** – Never filters the unused variables from the input data.

Insert custom SAS code before task and query code

Inserts custom code that you have written before the automatically generated task and query code. You can create or edit your custom code by clicking **Edit**.

Note: Your custom code can contain macro variables that are evaluated when the task is run. If your custom code includes a global SAS statement, then the first time the code runs, SAS Enterprise Guide does not use this global statement. This is because the custom task code is not intended to be run at the beginning of the process flow or project execution. This code is intended to be run immediately before the task code. On subsequent runs of the code, SAS Enterprise Guide uses the global statement.

Insert custom SAS code after task and query code

Inserts custom code that you have written after the automatically generated task and query code. You can create or edit your custom code by clicking **Edit**.

Note: Your custom code can contain macro variables that are evaluated when the task is run. If your custom code includes a global SAS statement, then the first time the code runs, SAS Enterprise Guide does not use this global statement. This is because the custom task code is not intended to be run at the beginning of the process flow or project execution. This code is intended to be run immediately after the task code. On subsequent runs of the code, SAS Enterprise Guide uses the global statement.

For more information about the order in which SAS Enterprise Guide processes task code, see How SAS Enterprise Guide Runs Task Code.

Setting SAS Studio Integration Options

To access the SAS Studio Integration task options, select **Tools** ⇨ **Options**. In the selection pane, click **SAS Studio Integration** under the Tasks heading.

Display SAS Studio tasks from your version of SAS Studio	Displays in the **Tasks** pane any tasks that were shipped with SAS Studio. If you are running SAS Studio 3.7 or later, you can access tasks from either the SAS Studio Enterprise Edition or the SAS Studio Single User Edition. SAS Enterprise Guide and SAS Studio Single User Edition must be installed on the same machine. If you have more than one instance of SAS Studio Single User Edition installed, you must select one instance to be active. Microsoft Edge 42.1 (or later) is required to run the SAS Studio tasks.
SAS Studio URL and credentials	Specifies the URL and credentials for SAS Studio 5.2. You must select this option to access tasks in SAS Studio 5.2 or later.

Setting Output Library Options

To access the output library task options, select **Tools** ⇨ **Options**. In the selection pane, click **Output Library** under the Tasks heading.

Default Library for Output Data	Displays the list of default libraries that SAS Enterprise Guide searches for when output data is created. The libraries are listed in the order in which SAS Enterprise Guide searches for them. By default, SAS Enterprise Guide searches for the first writable library from among Egtask, Gridwork, Work, and then Sasuser. **Note:** The Gridwork library is assigned only when a SAS grid is initialized for a task or a project. For information about running a task or project on a SAS grid, see "Using SAS Enterprise Guide with a SAS Grid" on page 12. You can add a new library to the search list by clicking **Add** and entering the name of the library. You can remove a library by selecting the library that you want to remove and clicking **Remove**. You can change the order of the search list by selecting the library that you want to move and clicking **Up** or **Down**.

Setting SAS Program and Editor Options

Setting SAS Program Options

To access the SAS Program options, select **Tools** ⇨ **Options**. In the selection pane, click **SAS Programs**.

Editor Options

For more information, see "Setting General Editor Options" on page 508.

Insert custom SAS code before submitted code

Automatically inserts any custom SAS code that you have written before the code in the Program Editor runs. You can create or edit your custom code by clicking **Edit**.

Note: Your custom code can contain macro variables that are evaluated when the task is run.

Insert custom SAS code after submitted code

Automatically inserts any custom SAS code that you have written after the code in the Program Editor runs. You can create or edit your custom code by clicking **Edit**.

Note: Your custom code can contain macro variables that are evaluated when the task is run.

Submit SAS code when server is connected

Automatically submits custom SAS code when a connection to a workspace server is made. For example, you could use this option to create some temporary data sets in the Work library each time you connect to a new server. You can create or edit your custom code by clicking **Edit**.

Insert SAS code before %STPBEGIN

Automatically inserts custom SAS code that you have written before the %STPBEGIN macro in your stored processes. You can create or edit your custom code by clicking **Edit**.

Insert SAS code after %STPEND

Automatically inserts custom SAS code that you have written after the %STPEND macro in your stored processes. You can create or edit your custom code by clicking **Edit**.

Initialize grid (if available) when connecting to a workspace

Specifies whether to automatically initialize a SAS grid connection when a workspace server connection is created. If a SAS grid has been configured, this option automatically initializes the SAS grid connection and assigns the Gridwork library when a workspace server connection is created. If this option is not selected, the SAS grid connection is not initialized and the Gridwork library is not assigned until the first task is run on the grid. This option is not selected by default. For an overview of SAS grid support in SAS Enterprise Guide, see "Using SAS Enterprise Guide with a SAS Grid" on page 12.

Note: You might want to select this option if you need to have grid-specific libraries such as Gridwork assigned before you run your first task on the grid. However, selecting this option can increase the amount of time it takes to connect to the workspace server.

Setting General Editor Options

To access the general editor options, select **Tools** ⇨ **Options**. In the selection pane, click **SAS Programs**. Click **Editor Options**, and then click the **General** tab.

Allow cursor movement past end of line	Enables you to move the cursor past the end of the line in the program window. If you do not select this option, the cursor jumps down to the next line when you move it after a carriage return, and only the text on each line is highlighted during selection.
Drag-and-drop text editing	Enables drag-and-drop editing to cut, copy, or paste text within SAS Enterprise Guide or between SAS Enterprise Guide and another Windows application.
Show line numbers	Displays line numbers in the leftmost column of the program and log windows.
Strip Ctrl+Z characters	Removes any Ctrl+Z characters from the file. Ctrl+Z characters can be an end-of-file marker in some files.
Enable DATA Step Debugger	Enables you to use the DATA Step Debugger with DATA step code. For more information, see "About the DATA Step Debugger" on page 330.
Enable smart highlighting	Automatically highlights all occurrences of a selected word in the Program Editor. ■ **For word selection** – Highlights all occurrences of a word that you select in the Program Editor. ■ **For find/replace search** – Highlights all occurrences of a word that you search for in the Program Editor.
Case-sensitive smart highlighting	Highlights only the occurrences of a selected word in the Program Editor that match the case of the selected word.

File type	Displays the type of file whose general settings you are editing.
Tab size	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.
Insert spaces for tabs	Inserts the number of spaces listed in the Tab size box instead of a single tab character.
Replace tabs with spaces on file open	Automatically replaces all tab characters with the number of spaces listed in the Tab size box when you open the file.
Indentation	■ **None** – Turns off all automatic indention. ■ **Automatic** – Automatically indents a new line by the amount of space that the previous line is indented.
Show section lines in text	Inserts lines between step boundaries within code.
Show section brackets in margin	Displays brackets in the margin to indicate step boundaries within code.
User-Defined Keywords	Opens a window that enables you to create your own list of keywords whose appearance you can customize. For more information, see "Creating User-Defined Keywords" on page 345.

Setting Editor Appearance Options

To access the general editor options, select **Tools** ⇨ **Options**. In the selection pane, click **SAS Programs**. Click **Editor Options**, and then click the **Appearance** tab.

File type	Specifies the type of file whose appearance you want to customize.
Scheme	Specifies the appearance scheme. For information about creating your own scheme, see "Customizing the Program Editor" on page 343.
Font	Enables you to select the font name and size. The font settings apply to all text for the specified file type. Note: A fixed-width font such as Courier New or Tahoma is recommended for the text.
File elements	Enables you to select the syntax element whose appearance you want to customize. If you are not sure what the name of the element is, click

a sample of the element in the sample program that is displayed in the **Preview** box.

Note: To customize the appearance of user-defined keywords, select a file type of **SAS Program** and a file element of **User-defined keywords**.

Setting Autocomplete Options

To access the general editor options, select **Tools** ⇨ **Options**. In the selection pane, click **SAS Programs**. Click **Editor Options**, and then click the **Autocomplete** tab.

Enable Autocomplete	Turns on the autocomplete feature of the Program Editor. This feature can predict the next keyword that you want to enter before you completely enter it. For more information, see "Using the Autocomplete Feature" on page 305.
Autocomplete on	Specifies on which character the autocomplete selection should be offered. You can choose to autocomplete your text on the first, second, or third character.
Commit a selection when typing any of these symbols	Specifies the symbols that can be used to commit, or insert, the selected keyword from the autocomplete list and the symbol into your program. For example, if you select RUN from the autocomplete list and enter a semicolon (;), then `run;` is added to your program. **Note:** The equal sign (=) is not included in the list of symbols, but it can be used to autocomplete a value-dependent option or statement such as DATA= or PARENT= to your program. However, if the currently selected keyword is not a value-dependent option or statement, then entering the equal sign does not automatically insert the keyword in your program. For example, if you enter PROC PRINT DA and then you enter =, DATA= is selected from the autocomplete list and inserted into your program. However, if you type DATA _NULL_; IN and then you enter =, IN= is inserted into your program. The autocomplete list includes the INFILE statement, but INFILE= is not inserted into the program because it is not a value-dependent statement. The behavior of the equal sign does not change even if you add it to the list of symbols for this option.
Add spaces when committing with these symbols	Specifies the symbols that can be used to commit, or insert, the selected keyword from the autocomplete list, followed by a space, the symbol, and then another space into your program. For example, if you select a macro variable

	myConstant from the autocomplete list and type +, then myConstant + is added to your program.
Add spaces when committing value-type options	Adds a space, "=", and another space when you commit, or insert, an option that ends in "=". For example, if you select this option and then select the YPIXELS= option from the autocomplete list, ypixels = is added to your program. If this option is not selected, ypixels= is added to your program.
Commit selection when pressing the SPACEBAR key	Commits, or inserts, the selected keyword into your program when you press the Spacebar key. This option is selected by default.
Commit selection when pressing the ENTER or TAB key	Commits, or inserts, the selected keyword into your program when you press the Enter or Tab key. This option is selected by default.
Autocomplete these keywords	Specifies which types of keywords you want SAS Enterprise Guide to autocomplete for you.
SAS color format	Specifies the color format that is used to specify colors. If you select SAS color names, then the predefined SAS/GRAPH color names are used. If you select RGB Color Codes, then the RGB color-naming scheme is used.
Enable integrated syntax help	Displays brief SAS syntax documentation as you write and edit your programs. This option is selected by default. For more information, see "Using the Syntax Help" on page 309.
Search Engine	Specifies the search engine that is used to search support.sas.com when you click the keyword link in the syntax help window. The default value is SAS Support. Select **SAS Support** to search for the keyword in the SAS Product Documentation.

Setting Indenter Options

To access the general editor options, select **Tools** ⇨ **Options**. In the selection pane, click **SAS Programs**. Click **Editor Options**, and then click the **Indenter** tab.

Indent size	Displays how many spaces are inserted for each tab character. This option can be set only by using the **Tab size** option on the General tab. You can change the default indent unit from a tab character to spaces. For more information, see "Setting General Editor Options" on page 508.

Put "THEN/ELSE DO" or "%THEN/%ELSE %DO" keywords on a single line

Puts all of the text for the THEN/ELSE DO or %THEN/%ELSE%DO keywords on a single line in the program. If you do not select this option, each keyword is displayed on a separate line.

Capitalize the first letter of every procedure name (if lowercase)

Capitalizes the first letter of all of the procedure names in your program. For example, proc means would be displayed as proc Means.

Add a blank line before these statements/definitions

Specifies the statements and definitions before which you want to insert a blank line.

Exclusions

Specifies the exceptions to the rules for inserting blank lines.

Setting Diff and Merge Tools Options

To access the Diff and Merge Tools options, select **Tools** ⇨ **Options**. In the selection pane, click **Diff and Merge Tools**.

Use custom file comparison tool

Uses an external file comparison tool to compare different versions of programs that are saved in your program history. For more information, see "Understanding Program History" on page 297.

- **File comparison tool** – Specifies the external file comparison tool to use when you are comparing different versions of a program in your program history. By default, SAS Enterprise Guide includes several of the most common file comparison tools, such as KDiff3, if they are installed in their default locations. If you want to specify another file comparison tool, select `Other`, and enter the pathname in the **Location** box. This option is available only if the **Use custom file comparison tool** option is selected.
- **Location** – Specifies the pathname of the external file comparison tool that you are using. Click **Browse** to browse for a file. This option is available only if the **Use custom file comparison tool** option is selected.

Use custom file merge tool

Uses an external file merging tool to compare different versions of programs and merge changes between versions. For more information, see "Understanding Program History" on page 297.

- **File merge tool** – Specifies the external file merge tool to use when you are comparing and merging different versions of a program in your program history. By default, SAS Enterprise Guide includes several of the most common file comparison tools, such as KDiff3, if they are installed in their default locations. This option is available only if the **Use custom file merge tool** option is selected.

- **Location** – Specifies the pathname of the external file merge tool that you are using. Click **Browse** to browse for a file. This option is available only if the **Use custom file merge tool** option is selected.

Setting Version Control Options

To access the Program History options, select **Tools** ⇨ **Options**. In the selection pane, click **Version Control**.

Commits to load in log history	Specifies the maximum number of rows to load in the commit history. The default value is 1000. Increasing the number of rows that are loaded might affect your performance. For more information, see “Viewing the Commit History” on page 469.
Enable program history for embedded files	Enables you to use the program history features with embedded programs and programs that are stored in Git repositories. If you clear this option, then the program history features are disabled. For more information, see “About Program History” on page 297.
Add a new version to the history	Specifies when to commit your program changes. ■ **Only when I choose to commit my changes** – Commits your changes only when you click **Commit** on the program toolbar. ■ **Every time I save my project** – Commits the changes to all programs in your project each time you save your project.
Clear the program history log for embedded programs	Deletes all previous versions of all programs in your project. You cannot delete individually selected versions of a program. When you delete your program history, it cannot be restored.
User name	Specifies a user name to use when you commit your program changes. This option is required if you are committing changes to a program in a Git repository.

Email	Specifies an email address to use when you commit your program changes. This option is required if you are committing changes to a program in a Git repository.

Setting Security Options

To access the Security options, select **Tools** ⇨ **Options**. In the selection pane, click **Security**.

Persist credentials	Determines the duration of security credentials. This option applies to password-protected data and projects. ■ **No persistence** – Credentials need to be re-entered each time the protected data or project is accessed. ■ **Persist during EG session** – Credentials do not need to be re-entered for the duration of the current SAS Enterprise Guide session. ■ **Persist with project** – Credentials need to be entered only once for a project. ■ **Persist for user** – Credentials need to be entered only once on a machine for the user that is logged in.
Do not put plain text passwords in generated code	Removes passwords for password-protected data from the automatically generated task code.
Mask credentials in SAS log when possible	Ensures that any security credentials that SAS Enterprise Guide inserts in the code before sending it to the server are removed from the log when the code comes back from the server.
Clear	Clears any credentials, such as data set passwords, that were cached.

Setting Administration Options

To access the Administration options, select **Tools** ⇨ **Options**. In the selection pane, click **Administration**.

Transfer Mode

You can choose to transfer files between servers in binary or text mode. The value that you select is the default for all files.

For example, you might want to use binary mode if you were copying a file with Japanese characters from a Japanese server to an English server so that the transfer would not fail with transcoding errors. However, you might want to use text mode to copy a file from a z/OS server to a Windows server.

- **Binary** – Transfers the files as is. There are no encoding changes even if the file is text based.
- **Text** – Automatically transcodes the files.

Note: You can also change the transfer mode for all files by right-clicking the background of the Server List window when no servers are selected and selecting **Transfer Mode** ⇨ **Binary** or **Transfer Mode** ⇨ **Text**.

Outgoing email server and port (SMTP)

Specifies the SMTP server and the port from which the email is sent. This is the server against which any authentication criteria are validated. For information about your mail server name and port number, contact the administrator of your SMTP server.

Sender email address

Specifies the email address of the sender. This address is displayed in the From field of the email message. The SMTP server might require that this address have a specific value that is associated with the authentication that is used to connect to the server. For more information, contact the administrator of your SMTP server.

Authentication

Select the type of authentication that you want to use:

- **None** – Uses an anonymous connection without a user ID and password.
- **Basic** – Requires a user ID and password that are sent without encryption.
- **SSL/TLS** – Requires a user ID and password that are sent with SSL encryption.
- **Windows Integrated** – Uses the Windows user ID and password with which you are currently logged in.

- **Windows Alternate** – Enables you to specify another valid Windows user ID and password.

Setting Application Logging Options

To access the Application Logging options, select **Tools** ⇨ **Options**. In the selection pane, click **Application Logging**.

Enable logging	Turns on the application logging feature in SAS Enterprise Guide. When this feature is enabled, a log configuration file and a log folder are created, if they do not already exist. Diagnostic log files are created in the log folder as you use SAS Enterprise Guide. - To edit the logging.config configuration file, click **Edit File**. The file opens in the default editor. - To view the diagnostic log files that have been created in the log directory, click **Browse Log Files**.
Enable metadata tracing	Captures metadata queries and results in order to help diagnose problems related to objects that are defined in the metadata.
Reset to Defaults	Disables the application logging feature and resets the logging.config file to its default contents and logging levels.

17

Using SAS Tasks

Using SAS Tasks in SAS Enterprise Guide

Accessing Tasks

SAS Enterprise Guide provides SAS tasks that you can run on your data. This list of tasks can include any tasks that are shipped with SAS Enterprise Guide, any custom tasks, and any tasks that are shipped with SAS Studio. (For more information, see the **Display SAS Studio tasks** option in "Setting Task Options" on page 504.) Any task templates that you have created also appear in the task pane.

To enable the **Tasks** pane, select **View** ⇨ **Tasks**. To open the **Tasks** pane, click ⚙

Note: If there is actively selected data that is opened in Update mode when you run a task, you are prompted to change it to Read-Only mode before you can run the task.

To learn more about a task, place your mouse pointer over the task name. From the flyout window, you can view the following information:

- a brief description of the task
- the category for the task
- whether the task is a built-in task (in other words, a task that shipped with SAS Enterprise Guide), a custom task, or a SAS Studio task
- the SAS procedures used in the task

About Temporary Libraries

As it performs various tasks, SAS Enterprise Guide creates temporary libraries. You can identify these libraries by their "EC" prefix (such as in EC1nnnnn, ECLIB0000, ECSQL000).

Occasionally you might see these libraries and their contents when you are browsing for files on a server. In general, these libraries should be reserved for software use only. Because they are temporary, it is best not to perform any action on (or select) the files in these libraries.

Changing the Destination of Output Data

By default, SAS Enterprise Guide tasks that create output data sets, such as the Standardize Data task, store the data in your Work library on the SAS server.

Note: Most tasks include an option to save your output data. To designate a different library for your output, use the Save As window that is accessible within the specific task. For more information, see the Help for the task.

There are several reasons why you might want to designate a different location for your output data:

- You might want the data to reside in a location other than that defined by Work—for example, in a central library that many users can access.
- The Work library is configured as a temporary location. Therefore, if you want your output data to be saved permanently, then you need to designate a different location for it.

SAS Enterprise Guide can use a special library, which is named Egtask, to store output data. When you run tasks, SAS Enterprise Guide detects whether this library exists. If it does, then SAS Enterprise Guide saves output data to that location instead of to Work.

Note: You can modify the list of libraries that SAS Enterprise Guide searches for by using the output library options.

To ensure that the Egtask library is available when you run your task, you can use an autoexec process to define the Egtask library and associate it with the server that you run tasks on. After you create this definition, the Egtask library is defined automatically when you connect to your SAS server.

Viewing the Results of a Task

Right-click the task item in the Project pane or process flow and select **Open**. The results from the previous run of the task appear in the SAS Enterprise Guide window.

Note: You can also open the results by double-clicking on the task node in the process flow.

Using Prompts in a Task

In a SAS task window, any variable prompts that you created appear in the **Variables to assign** list. If the task is a wizard, then the variable prompt is available in the Assign variables to role step. You can assign this variable prompt to any role in the task. When the task runs, the user must select a variable from the input data source to use in the analysis.

For example, suppose you need to create bar-line charts that show the predicted and actual sales by Country, Region, or Product Type. For a task, you can create a prompt by using the Prompt Manager.

Note: If you assign a prompt to more than one role, you are prompted only once. If you want to be prompted for each role to which you have assigned a prompt, you must create separate prompts for each role.

1 First, you create a character variable prompt called Categories. (You can create a prompt by using the Prompt Manager.) You can populate the values for this prompt from your input data source so that when the prompt runs, the user can select only from the character variables in the input data source.

2 Then, you open the Bar-Line Chart task. In the task, you assign the Categories prompt to the **Group charts by** role.

3 When the task runs, the user selects whether to group the charts by the Country, Region, or Product Type variable.

Modify a Task

1 Right-click the task item in the Project pane or process flow and select **Modify**.

2 Edit the options in the task.

3 Click **Run** or **Save**.

Adding User Code to a Task

When you preview task code, you have the option to insert your own code into the code that SAS generates for the task. You can insert SAS code procedure options or statements into the generated code with all the features of the Program Editor. You cannot edit or delete the automatically generated code within the Code Preview for Task window; you can insert only new code.

To insert your own code:

1 In the task window, click **Preview code**.

2 In the **Code Preview for Task** window, select the **Show custom code insertion points** check box. These lines are added to the task code in each location where you can insert custom code:

```
/* Start of custom user code (Framework_BeforeTaskCode) */
<insert custom code here>
/* End of custom user code (Framework_BeforeTaskCode) */
```

3 Click the location where you want to insert the code and enter your code.

4 To save your changes, close the Code Preview for Task window. To clear all of the custom code that you have entered, click **Clear All**.

Setting Task Properties

How to View the Task Properties

You can view and change many of the properties of each task. By default, tasks use settings from the Options dialog box, but you can choose to customize those settings for individual tasks.

You can view the properties for a task in these ways:

- Right-click the item in the Project pane or in a process flow and select **Properties**.
- Click **Edit** in the **Properties** panel of the task window.

General Properties for Tasks

- **Label** specifies the name of the task.
- **Last Execution Time** displays how long it took for the task to run the last time it was executed. This time is rounded to the nearest second and does not include the time it takes to connect to the server.
- **Server** specifies the name of the server on which SAS Enterprise Guide executes the task. You cannot change the server for a task. SAS Enterprise Guide executes the task on the server where the data for the task resides.
- **Input data** specifies the name of the data set on which the task is based. You cannot change the input data for a task.
- **Data limits**

  - **Start at row** specifies the number of the row that you want to use as the first input row for the task.
  - **Limit the number of rows used** limits the number of input rows that can be used to generate results from a task. If a filter has been specified, then the limit is applied to the filtered data. Specify the upper limit of rows that you want to use for the task.

Note: These options are not available for the Append Table, Create Format, Import SAS Information Map, Import Data, High-Performance Linear Regression, and High-Performance Logistic Regression tasks.

Results Properties for Tasks

- **Use preferences from Tools->Options** - specifies that the results for the task should use the settings that are specified in the Results section of the Options dialog box. By default, the SAS Report format is used with the **HTMLBlue** style. The other results options are disabled if you select this option.
- Customize result formats, styles, and behavior - specifies that you want to customize the results for this task. You can choose from these results formats and styles:
  - **SAS Report** - generates results in SAS Report format. You can also select a style from the drop-down list. The style will be applied to the SAS Report results for this task. SAS Report results can be used to create a report.
  - **HTML** - generates results in HTML format. This is the default. You can also select a style from the drop-down list. The style will be applied to the HTML results for this task.
  - **PDF** - generates results in PDF format. You can also select a style from the drop-down list. The style will be applied to the PDF results for this task.
  - **RTF** - generates results in RTF format. You can also select a style from the drop-down list. The style will be applied to the RTF results for this task.
  - **Text** - generates results in text format.
  - **Excel** - generates results in Microsoft Excel format. This format can be generated by any SAS server that is running the third maintenance release for SAS 9.4 or later.
  - **PowerPoint** - generates results in Microsoft PowerPoint format. This format can be generated by any SAS server that is running the third maintenance release for SAS 9.4 or later.

Note: For more information about creating and editing styles, see “About the Style Manager” on page 531.

- **Graph format** - determines the image format for graphs in results for this task. To change the format, select another format from the drop-down list. For information about the different image formats, refer to Selecting graphics output format.

Code Submission Properties for Tasks

- **Use preferences from project properties** specifies that the code submission settings for the task should use the settings that are specified in the **Code Submission** section of the Properties dialog box. For more information, see "Viewing Project Properties" on page 37.
- **Customize code submission options** specifies that you want to customize the code submission options for this task. You can choose from these options:
  - **Use grid if available** enables SAS Enterprise Guide to run tasks in a SAS grid computing environment. For an overview of SAS grid support in SAS Enterprise Guide, see "Using SAS Enterprise Guide with a SAS Grid" on page 12.
  - **Allow parallel execution on the same server** enables SAS Enterprise Guide to create one or more additional workspace server connections so that parallel process flow paths can be run in parallel on the same server. For more information, see "Using SAS Enterprise Guide with Parallel Execution" on page 13.

  Note: Because there might be more than one workspace server connection, any changes that you make in one workspace session, such as library assignments or changes to macro variables or options, might not be available to tasks that are running in another parallel workspace session.

- **Action to take on error in the execution of this task** specifies how to handle an error in the execution of this task.
  - **Stop all items** stops the execution of all subsequent items in the project or process flow.
  - **Stop current branch** stops the execution of all subsequent items in the current branch of the process flow. This is the default value.
  - **Continue execution** continues the execution of all subsequent items in the project or process flow.

Prompt Properties

Project prompts used displays the SAS name, display name, and data type of each prompt that you have associated with the task. You can add and remove prompts from the task and create prompts by using the Prompt Manager. For more information, see "Using Prompts in a Task" on page 520.

To associate existing project prompts with the task, click **Add** and select the prompt that you want to add.

Working with the Input Data Source

Selecting an Input Data Source

When you open a task, the task uses the active data source as the input data source by default. However, in SAS Enterprise Guide, you can change the data source from the **Data** panel for the task.

1. In the **Data** panel of the task or the Verify Data step of the wizard, click **Edit**.
2. Select a data source.
3. (Optional) Create a filter for the data.
4. Click **OK**. The name of the input data source now appears in the **Data** panel of the task.

Sorting the Input Data Source for a Task

Before running some tasks, SAS automatically sorts the input data source. Sorting is performed only for SAS tasks where sorted data improves performance.

By default, SAS uses SORT procedure to sort the data. However, if the input data source is from a DBMS engine, then SAS uses the functionality on the database to perform this sort. Using the database functionality optimizes the performance of the SAS task.

Working with Task Templates

About Task Templates

When you run a task, such as the Bar Chart task, you might want to frequently use the same settings. Instead of opening the task multiple times to specify these settings for different input data sources, task templates enable you to save your settings for a specific task to a template. You can then run that template with any input data source.

When working with task templates, remember these limitations:

- Values or settings that depend on the data source are not saved in the task template. For example, when you assign a variable to a specific role, the variable assignment is not saved in the task template. If an option is dependent on the values of the data source, then the option setting is not saved in the task template. For example, in the Line Plot task, symbol definitions that depend on the data values are not saved in the task template.
- You cannot create a task template from these tasks: Compare Data, Forecast Studio Create Project, Forecast Studio Open Project, Forecast Studio Override Project, List Report Wizard, Model Scoring, and Summary Tables.

In SAS Enterprise Guide, task templates are available from the **Tasks** pane. These templates are listed with all other tasks.

Task templates are also available from the Task Templates Manager. To open the Tasks Template Manager, open the **Tasks** pane. Click ⋮ and select **Manage task templates**.

Create a Task Template

1. Open the task and specify the options for the template.
2. Click the down arrow next to **Run** and select **Create Template** from the pop-up menu. The Create Task Template window appears.

   Note: If the task is a wizard, click the down arrow next to **Finish** and select **Create Template** from the pop-up menu.

3. Specify a name for the template.
4. (Optional) Provide a brief description for the template.
5. Select where to add the new template. You can select from these options:
   - **Create a new task template in group**, and then select the group from the drop-down list.
   - **Replace an existing task template**, and then select the task to replace from the tree view.

     Note: If you choose to replace an existing task template, then the name that you previously specified is used for the new template.

6. Click **Create**. The new task template is now available from the list of tasks.

Note: You can create a template from an existing task by right-clicking the task and selecting **Create task template** from the pop-up menu.

Export a Template

You might want to export a task template to a central location, so it can be imported by other users at your site. To export a task template:

1 In the **Tasks** pane, click ⋮ and select **Export Template**. The Export Task Templates window appears.

2 Select the templates that you want to export.

3 To select the location for these templates, click **Export to**. Select the location in the Browse for Folder dialog box and click **OK**. This location now appears in the Export Task Templates window.

4 Click **Export**. A copy of the templates is saved to the location that you selected.

Import a Template

1 In the **Tasks** pane, click ⋮ and select **Import Template**. The Import Task Templates window appears.

2 To select one or more templates to import, click **Add**. Select the templates that you want to add and click **Open**. The templates now appear in the **Templates to import** pane.

3 From the **Import to group** drop-down list, select the group where you want to place the imported templates.

4 Click **Import**. The templates now appear in the list of tasks.

Run a Task Template

1 Double-click the task name in the **Tasks** pane.

2 (Optional) Select the input data source.

3 In the **Data** panel of the task window, assign variables from the input data source to task roles.

4 (Optional) Modify any additional options in the task.

5 Click **Run**. The results from the task now appear in SAS Enterprise Guide.

Working with Custom Tasks

About Custom Tasks

You can create custom add-in tasks specifically for SAS Enterprise Guide. You can add any custom add-in tasks that you created for use with SAS Enterprise Guide 3.0 or later. When you add a custom task to SAS Enterprise Guide, you are able to access that task from within SAS Enterprise Guide by selecting it in the **Tasks** pane.

You can add custom tasks to SAS Enterprise Guide in two different ways:

- Use the Add-In Manager in SAS Enterprise Guide
- Create a directory for the custom tasks

Add a New Task to the List of Available Add-Ins

1. To open the Add-In Manager, select **Tools** ⇨ **Add-In manager**
2. Select the registry where you want the task DLL file to be stored. You can select one of the following registries:
   - **All** displays all of the task files that are currently registered with SAS Enterprise Guide.
   - **Common** displays all of the task files that are associated with your personal computer. Anyone who logs on to your personal computer can access these tasks.
   - **User** displays all of the task files that are stored in your personal registry. Only you are able to access these tasks.
3. Click **Browse**. Navigate to the directory where the DLL file for the task is stored, and select the file.
4. Click **Open**. The task is added to the list of available add-ins.
5. To update the metadata for a task that you have changed, select the name of the task and then click **Update**.

   Note: If you click **Update** without selecting a specific task, all of the metadata for the registry location is updated.

6 Click **OK**. You can access the task by selecting **Tools** ⇨ **Add-In** ⇨ ***task-name*** , or by selecting it in the **Tasks** pane.

Creating a Directory for Custom Tasks

You can add a custom task to SAS Enterprise Guide by copying the appropriate DLL (dynamic link library) files to a special directory. From this special directory, SAS Enterprise Guide can automatically load the files.

There are two custom directories into which you can copy the files:

- If you have administrator privileges, you can create a directory named "custom" in the SAS Enterprise Guide application installation directory. If you copy custom task files to this location, then the custom tasks in this folder are available to all users who run this application. For more information, contact your system administrator.
- You can create a directory named "custom" in the folder for application-specific SAS Enterprise Guide application data. If you copy custom task files to this location, then the custom tasks in this folder are available only to the user of this application data and only in SAS Enterprise Guide.

You can access the task by selecting it in the **Tasks** pane.

Using a Grid Computing Environment

This section is for traditional workspace server setups. If you have launched your workspace servers using the SAS grid (which is the recommended setup), this section does not apply.

Running a Task with Local Data on a SAS Grid

If you want to run a task on a SAS grid and use local data, you must first upload the local data file to the Gridwork library on the grid node. By default, when you run a task that uses local data, SAS Enterprise Guide makes a temporary copy of the data in the Work library on the workspace server where the task is run. However, when a task is run on a SAS grid, SAS Enterprise Guide cannot access the Work library on the workspace server.

In order to upload local data to the Gridwork library, you must first ensure that the Gridwork library has been assigned, and then use the Upload Data Files to Server task.

To upload local data to the Gridwork library:

1 Confirm that the Gridwork library is available by expanding the appropriate server in the Server pane, and then expanding the list of libraries. The Gridwork library should appear in the list.

  If the Gridwork library is not included in the list of libraries, right-click the server and select **Connect To Grid**. When you select this option, SAS Enterprise Guide establishes the connection to the grid, and the Gridwork library is assigned so that it is available for you to use.

2 To upload the data to the Gridwork library, in the **Tasks** pane, select **Upload Data Files to Server**. For more information, see “Upload Data Files to Server” in *SAS Tasks in SAS Enterprise Guide and SAS Add-In for Microsoft Office*. When you specify the upload options, select **GRIDWORK** as the destination library.

See Also

“Upload Data Files to Server” in *SAS Tasks in SAS Enterprise Guide and SAS Add-In for Microsoft Office*

Enabling Tasks to Run on a SAS Grid

If your site has configured a SAS grid, you can specify that a SAS Enterprise Guide task run on the grid. Running a task on a SAS grid provides workload balancing among the grid machines based on job priority and resource allocation.

Note: If you want to run a task on a SAS grid and use local data, you must first upload the local data file to the Gridwork library on the grid node. For more information, see “Running a Task with Local Data on a SAS Grid” on page 528.

To run a task on a SAS grid:

1 Right-click the task and select **Properties**, and then click **Code Submission** in the selection pane.

2 Select **Customize code submission** options, and then select **Use grid if available**. If a SAS grid has been configured, the task code is automatically sent to the grid for processing. If a SAS grid has not been configured, then this option has no effect.

Note: If you specify that a task should run on the grid, you can also use the **Allow parallel execution on the same server** option to specify that each task use a separate grid node. This option is also available in the **Customize code submission options** section of the task properties window. Running tasks on multiple grid nodes enables you to take advantage of the parallel process capabilities of the SAS grid. However, you must plan for access to shared data and libraries between the tasks. For more information, see “Using SAS Enterprise Guide with Parallel Execution” on page 13.

Note: You can also configure projects to run on the SAS grid. For more information, see "Using SAS Enterprise Guide with a SAS Grid" on page 12.

For more information about the SAS grid computing environment, see the Scalability and Performance Community and the documentation page for SAS Grid Manager.

Appendix 1

Customizing Styles

About the Style Manager

You can change the look of your HTML or SAS Report output by modifying and creating styles. Styles are based on cascading style sheets (CSS) and are a set of specifications that control what your HTML or SAS Report output looks like. There are several predefined styles that you can use. You can select an existing style for your output, modify an existing style, or create a new style. You can also use external styles that reside on a web server or elsewhere on your network.

To open the Style Manager:

- In SAS Enterprise Guide, select **Tools** ⇨ **Style manager**.
- In SAS Add-In for Microsoft Office, click the **SAS** tab in the Ribbon. In the **Tools** group, click **Tools**, and then select **Style Manager**.

Set a Default Style

1 Open the Style Manager.

- In SAS Enterprise Guide, select **Tools** ⇨ **Style manager**.
- In SAS Add-In for Microsoft Office, click the **SAS** tab in the Ribbon. In the **Tools** group, click **Tools**, and then select **Style Manager**.

Note: In the Style Manager window, the default style is displayed in bold in the **Style List** box.

2 From the list of available styles, select the style that you want to use as your default style, and click **Set as Default**.

3 Click **OK**.

Create a Copy of an Existing Style

You can create a copy of an existing style so that you can edit it.

1 Open the Style Manager.

- In SAS Enterprise Guide, select **Tools** ⇨ **Style manager**.
- In SAS Add-In for Microsoft Office, click the **SAS** tab in the Ribbon. In the **Tools** group, click **Tools**, and then select **Style Manager**.

Note: In the Style Manager window, the default style is displayed in bold in the **Style List** box.

2 From the list of available styles, select the style that you want to copy and click **Create a Copy**.

3 Enter the name of the new style.

4 Click **Save**. The copied style, which has a location of My Style, is added to the list of styles.

Note: Styles that you copy and save are stored in your user profile area on your machine in this directory: `%APPDATA%\SAS\SharedSettings\Styles`. For example, this might resolve to `C:\Users\`*`user profile name`*`\AppData\Roaming\SAS\SharedSettings\Styles`.

Edit an Existing Style

There are several predefined, or built-in, styles that you can use as is or that you can make a copy of and then modify and save. You can also edit the name and URL of external styles.

Note: You cannot edit a predefined style without first making a copy of it.

1 Open the Style Manager.

- In SAS Enterprise Guide, select **Tools** ⇨ **Style manager**.
- In SAS Add-In for Microsoft Office, click the **SAS** tab in the Ribbon. In the **Tools** group, click **Tools**, and then select **Style Manager**.

Note: In the Style Manager window, the default style is displayed in bold in the **Style List** box.

2 From the list of available styles, select the style that you want to edit and click **Edit**. If you are editing a copied style with a location of My Style, the Style Editor window opens so that you can make changes to the style.

If you are editing an external style, then the Add New Style window appears so that you can change the style name and URL.

Note: You can edit only styles with a location of My Style or External Style.

Create a New Style Based on an Existing Style

You can use the predefined styles as the basis for creating a new style.

1 Open the Style Manager.

- In SAS Enterprise Guide, select **Tools** ⇨ **Style manager**.

- In SAS Add-In for Microsoft Office, click the **SAS** tab in the Ribbon. In the **Tools** group, click **Tools**, and then select **Style Manager**.

Note: In the Style Manager window, the default style is displayed in bold in the **Style List** box.

2 Click **Add**. The Add New Style window appears.

3 Select **Add new based on existing style**.

4 In the **Style name** box, enter the name of the new style.

5 From the **Based on** drop-down list, select the style that you want to use as the basis for your new style, and click **OK**.

Note: Styles that you copy and save are stored in your user profile area on your machine in this directory: `%APPDATA%\SAS\SharedSettings\Styles`. For example, this might resolve to `C:\Users\`*`user profile name`*`\AppData\Roaming\SAS\SharedSettings\Styles`.

Add an External Style

You can add external styles that you can apply to any results that you generate. There are two types of external styles that you can add:

- external styles that reside on a Web server or elsewhere on your network
- SAS styles that are stored on the SAS server

Note: After you add an external style, you can edit the style name and URL.

1 Open the Style Manager.

- In SAS Enterprise Guide, select **Tools** ⇨ **Style manager**.
- In SAS Add-In for Microsoft Office, click the **SAS** tab in the Ribbon. In the **Tools** group, click **Tools**, and then select **Style Manager**.

Note: In the Style Manager window, the default style is displayed in bold in the **Style List** box.

2 Click **Add**. The Add New Style window appears.

3 Select **Add new external style**.

4 In the **Style name** box, enter the name of the style.

5 In the **Style URL** box, enter the URL location of the style (for example, http://www.sas.com/tjf/Reports/SansPrinterTop.css).

If the style is a SAS server style, select **This is a SAS server style only**. The **Style URL** box is disabled, because you do not need to specify a URL for SAS server styles.

6 Click **OK**.