

Find Out What You're Missing: Programming with SAS® Enterprise Guide®

Chris Hemedinger, SAS Institute Inc., Cary, NC

ABSTRACT

In this paper, you can read about the productivity gains that you can enjoy when you add SAS® Enterprise Guide® to your SAS programming toolbox. You will see how to perform old tasks in a new way as well as how to accomplish some tasks that would have been very difficult—if not impossible—without the benefit of an integrated tool like SAS Enterprise Guide. Topics in this paper include:

- Working with projects and the process flow
- Getting the most from SAS tasks
- Flexing your power with project prompts
- Conditionally running portions of your project
- Noting a few things that you cannot do

INTRODUCTION

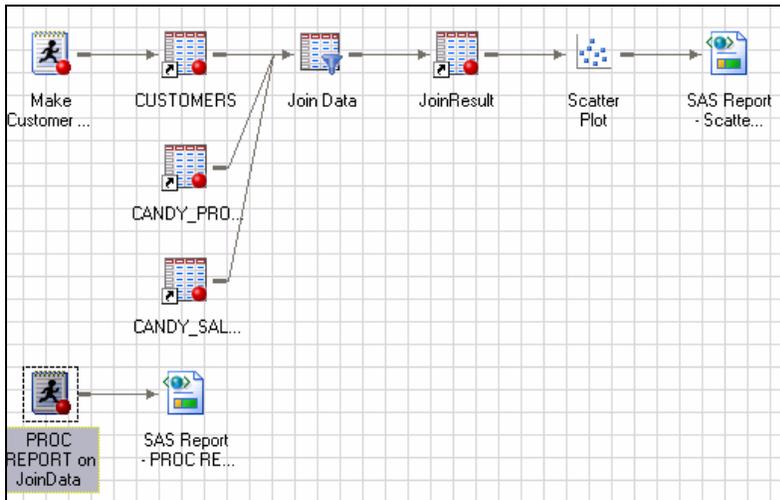
More and more SAS programmers are adopting SAS® Enterprise Guide® as their programming environment. Many are “willing converts” because they see the productivity benefits. Others are reluctant, but are forced down this path because their organization has centralized SAS on a server, and SAS® Enterprise Guide® is their new gateway to SAS. Whatever your motivation, there are many techniques and features that can help you to get the most out of your new SAS environment.

GET ORGANIZED WITH PROJECTS

One of the biggest advantages that SAS® Enterprise Guide® offers is the capability to organize your work in project files. A SAS Enterprise Guide project is a great place to store related work together, including SAS programs, references to data, Output Delivery System (ODS) results such as HTML, and SAS logs.

Project files do more than store all your work items, though. Project files also store the relationships among those work items. The process flow view of your project serves as a form of documentation for your work.

The following image shows a sample process flow:



You can easily read this project from left to right to see how it's put together. It begins with a SAS program, Make Customer Data, which builds a customer table. That table, along with two other tables, is used as input into a query task that joins the three tables together to create an output table named JoinResult. JoinResult is then used as input to a scatter-plot task.

The only item that appears to be hanging out there is the SAS program at the bottom, the one that is labeled PROC REPORT. Although the label might be informative, it is difficult to see how it relates to the other items.

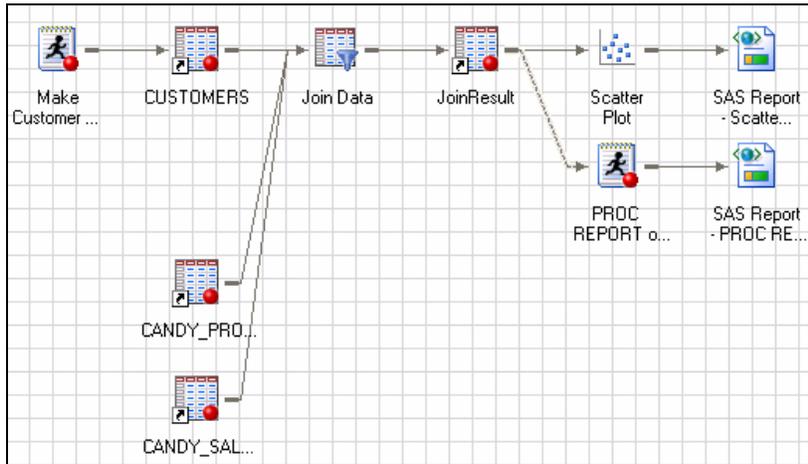
CONNECT THE DOTS WITH LINKS

The relationships that are described in this sample project so far, as shown by the arrow links in the process flow, are implicit. That is, SAS Enterprise Guide detects these relationships and illustrates them in the process-flow view, with no intervention needed from you. SAS Enterprise Guide also lets you define your own links among items. This adds even more readability to your project and enforces the sequence in which items are run.

For example, suppose that the lone SAS program with the PROC REPORT label is meant to report on the JoinResult table. You could follow the steps below to build an explicit link from the data table to the SAS program:

1. Right-click the JoinResult item and choose Link JoinResult To. The Link window appears, showing a list of candidate items in the project to which you can link.

- Choose the PROC REPORT item from the list and then click **OK**. The process-flow view updates to show the new relationship, as is shown here:



Tip: When you link a data item to a SAS program item as input, SAS Enterprise Guide automatically assigns the data reference to the &SYSLAST macro variable before running the SAS program. Most SAS procedures will use the &SYSLAST value as the DATA= value, if set. You can use this technique to associate data tables with generic SAS programs without having to refer to the data by name within the program.

AVOID ENTROPY WITH ORDERED LISTS

The process flow ties related tasks together. It is then easier to run them all as a group and ensure that the tasks that produce output that is needed by other tasks are run first. But what if you want to run just a subset of the tasks in your project, but still keep them in a certain sequence? In the manual method, you would have to select each task one at a time, run it, wait while it completed, and repeat this procedure for each task in order.

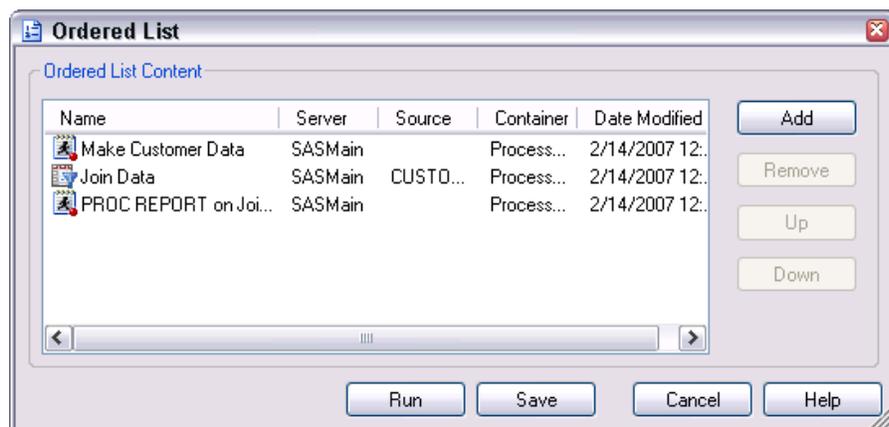
SAS Enterprise Guide has a hidden gem of a feature, ordered lists, which let you build simple lists of tasks from your project that you want to run in a prescribed sequence. You can select these tasks à la carte from anywhere in your project, including across multiple process flows, and run them in whatever order you need.

To create an ordered list, follow these steps:

- Select Tools** ► **Create Ordered List** . The Ordered List window appears.
- Click **Add**. The Add from Project window appears, presenting you a list of all the tasks within your project.
- Choose the tasks you want to include by clicking them; then click **OK** to add them to your list.

Tip: Press and hold the CTRL key to select multiple items at once.

This picture shows an example of the Ordered List window with a few tasks added. At this point, the tasks might not be in the correct order for your needs.



4. To change the sequence for a task, select it in the list and click the **Up** or the **Down** button to move it within the list.
5. When the list of tasks reflects the order that you want, click **Save**.
6. If you want to run the tasks immediately, click **Run**.

The ordered lists that you create appear in a special Ordered Lists section of your project view. To run an ordered list after you create it, right-click the list item in the **Ordered List** section and choose **Run Ordered List**. SAS Enterprise Guide runs each task in the list in the correct order.

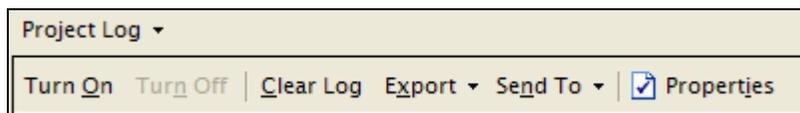
THE PROJECT LOG: YOUR WORK ON RECORD

Every task and SAS program that you run in SAS Enterprise Guide generates a log file as part of its output. SAS programmers rely on log files to show what work was done, how long it took to complete the work, and whether any errors or warnings occurred.

The project log (supported in SAS Enterprise Guide® 4.1 and enhanced in 4.2) is an aggregated view of all of the log files for all the tasks in your project. Every time you run your task or even the entire process flow, SAS Enterprise Guide adds the logs to the project-log view. The logs accumulate across iterations, which means that the project log offers a history of every task you have run in your project. When you save your project, SAS Enterprise Guide saves your project log along with it.

The project-log feature is not enabled by default, so you must turn it on when you create your project if you want to build up this project history. To enable the project log in SAS Enterprise Guide 4.2, follow these steps:

1. **Select View ► Project Log.** The Project Log window appears, the top of which is shown here:



2. Choose **Turn On** at the top of the Project Log window.
3. SAS Enterprise Guide 4.1 requires these additional steps to enable the project log: open the project properties, find the Project Log section, and choose **Maintain Project Log**.

LET SAS TASKS DO THE HEAVY LIFTING

SAS Enterprise Guide supplies nearly 90 tasks that generate SAS program code for you, and all you have to do is point and click. The tasks cover basic data reporting, plots and charts, and advanced statistics. You can use these tasks as a starting point for writing SAS programs. Let SAS Enterprise Guide generate as much of the code as possible.

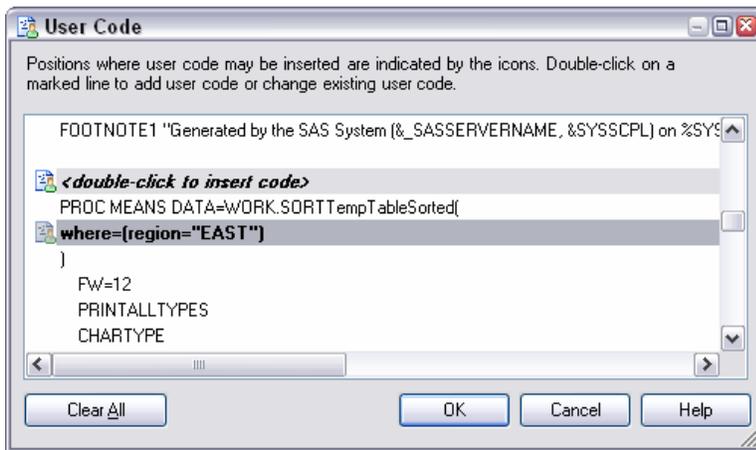
SAS tasks cover the most popular options for the SAS procedures. However, it doesn't take long for an experienced SAS programmer to discover that something is missing—some option or statement that hasn't surfaced in the point-and-click task interface.

There is a simple and obvious remedy: use the SAS task to generate as many of the statements and options as possible; then take a copy of the generated code and use it as the basis for your own SAS program. (SAS Enterprise Guide even offers a CREATE CODE TEMPLATE option to make this easier.) The disadvantage of this approach is that after you create your own SAS program from the task-generated version, you can no longer use the task-user interface to maintain the program. You are on your own with the SAS program editor.

Here's a better way: Many SAS tasks allow you to insert your own statements and options at predefined points within the task-user interface. By using this feature, you can have it both ways: point-and-click for the mainstream options, with the capability to customize the generated SAS program with some extra statements.

Here are the steps to insert your own statements within a task, using the Summary Statistics task as an example:

1. Select **Describe ► Summary Statistics**, and use the data of your choice. The task window appears.
2. Use the controls on the page to select which variables to analyze, and any other options that you want.
3. Click the **Preview Code** button on the bottom left of the task window. The Code Preview window appears with the SAS program code that reflects your selections in the task thus far.
4. On the top of the Code Preview window, click the **Insert Code** button. The User Code window appears, as shown here:



As you scroll through this user code view, notice several lines labeled “<double-click to insert code>”. These are the locations within the SAS program that the task defines for you. They allow you to insert your own statements and options.

5. To insert your own code, double-click one of the lines indicated. The Enter User Code window appears with a text field.
6. You can type your own SAS code segment into the text field.
7. After you add the options you want, click **OK**.

8. You can then close the User Code window, and also close the Code Preview window by clicking the button again on the task window.

Note: Whatever user code you enter is merged into the task-generated code as is. This means that you need to take special care that the code you enter is syntactically correct and makes sense at the insertion point that you selected. If you make a mistake, you will see errors in the SAS log when you run the task.

GET FLEXIBLE WITH PROJECT PROMPTS

Like most software development, SAS programs tend to evolve over time. The first stage of any given SAS program usually consists of DATA step code and procedure statements written to perform a task against a specific source of data. Perhaps the program is required to meet a short-term goal or simply serve as a prototype or proof of concept.

If you get good results with that first version of the program, chances are high that you or someone else will want to use your program to analyze a different data source, or perhaps even a variety of data sources. It is at this point in the SAS program lifecycle that you might consider restructuring the program code to be more generic and reusable. You might use macro variable substitution with %LET statements at the top of your program to assign values as needed, or you might devise a more sophisticated version that contains a SAS macro program with parameters in the macro call.

SAS Enterprise Guide can integrate with your SAS programs using project prompts (called project parameters in previous releases). You can think of prompts as SAS macro variables that SAS Enterprise Guide keeps track of, so that when you run your project, the application knows to prompt you for values. After gathering responses to the interactive prompts, SAS Enterprise Guide generates the %LET statements for you and includes them as a prologue to your program.

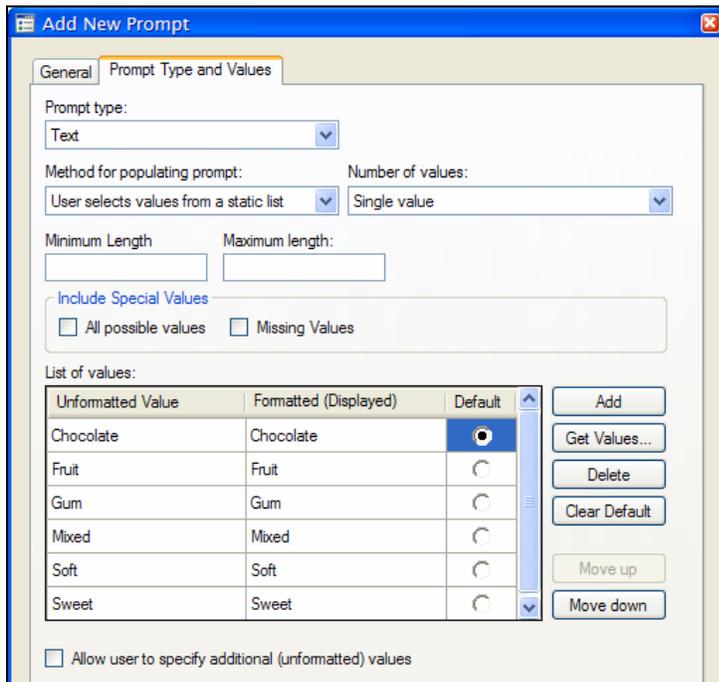
Macro variables are usually simple constructs in SAS, but prompts can be much more sophisticated to provide prompts to an end user. You can create prompts to accept text strings, numbers (with range validation), single or multiple values from a predefined list of values, date or date-time values, and even variable names for use within SAS task roles.

To get started with project prompts in SAS Enterprise Guide 4.2, perform the following steps:

1. Select **View ► Prompts Manager**. The Tools-Prompt Manager window appears.
2. Click **add**. The Add New Prompt dialog box appears.
3. Type a name for your prompt.

SAS Enterprise Guide automatically forms a valid SAS code name or macro variable name from the descriptive name that you entered. You can change this code name if you want. You can also add a description to help document the prompt.

4. Click the **Prompt Type and Values** tab near the top of the window. Here is an example of this dialog box:



In this example, the prompt type is **Text**, and the data value type is **Single value**. With this list type, you have the opportunity to specify the contents of the list to present during a prompt. The **Get Values** button (on the right) offers an easy way to populate the list based on data values within a SAS data set or other data source.

Notice that the Add New Prompt dialog box also contains many options for how to treat this prompt. You can specify a default value, make it required, or even hide the prompt. There are too many options in this window to describe here; the feature is very flexible, and you should be able to find a combination of options to fit your needs.

5. When you finally settle on all your options, click **OK** to add the prompt to the project. It now appears in the Prompt Manager window.

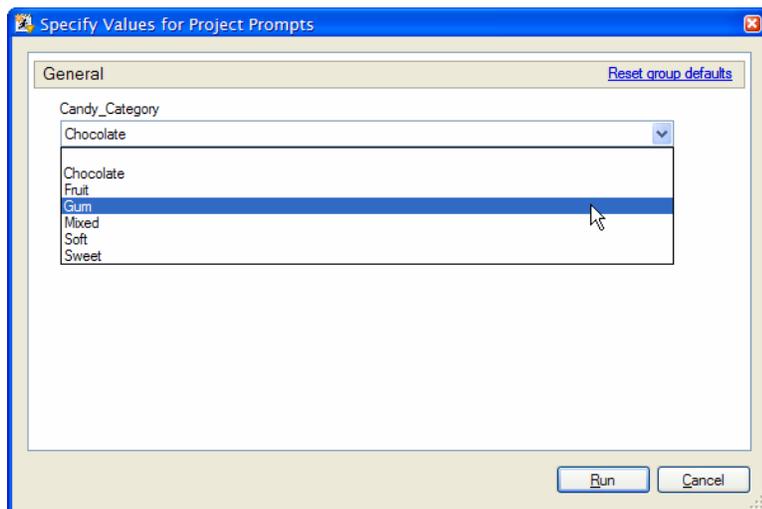
The most natural place to use prompts in SAS Enterprise Guide is within the Query Builder. You can use prompts within filters to make a query definition that is more flexible.. For example, instead of creating a filter that equates to WHERE REGION="EAST", you can substitute a prompt value for the literal value "EAST" and prompt for the valid regions.

You can also use prompts within SAS programs. As you write your program, use the macro variables within the program as you normally would. But instead of having a list of %LET statements at the top of your program to set values for your macro variables, you can have SAS Enterprise Guide prompt you for values.

To associate the prompt with a SAS program, follow these steps:

1. In the SAS program window, select **Properties** (located near the top). The Properties window appears.
2. In the Properties window, select the **Prompts** section, and then click **Add**. The Select Prompts window appears.
3. Select the prompts that you want to associate with this SAS program.

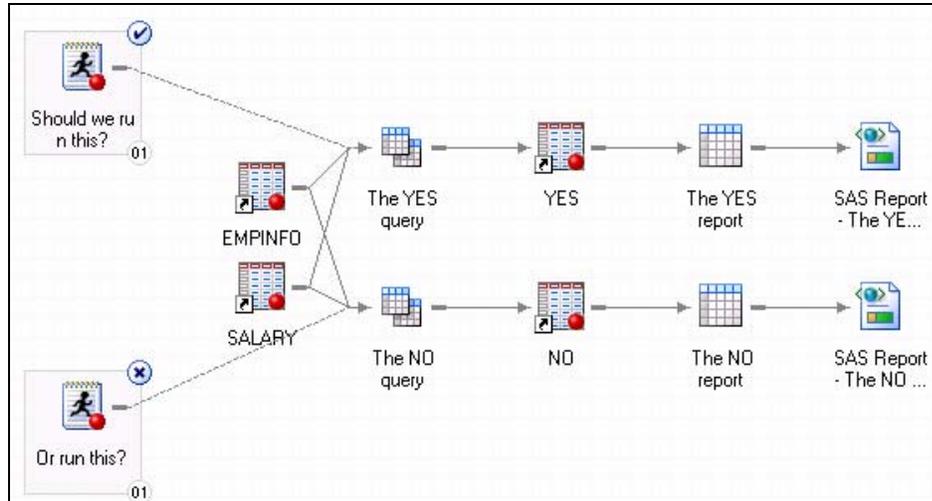
Now when you run your program in SAS Enterprise Guide, the prompts window gathers the values for your macro variables. It looks like this:



TO RUN OR NOT TO RUN: APPLY CONDITIONS TO YOUR TASKS

SAS Enterprise Guide 4.2 provides a new way to control how your project is run. Every runnable object in the project (for example, SAS programs, tasks, and queries) can examine one or more conditions before they are actually run. By defining these conditions, you can implement an if-then-else processing structure within your projects.

Here is an example process flow with some conditions applied:



In this example, the decision as to which branch to run depends on the response to a project prompt. If you select **Yes** when the process runs, SAS Enterprise Guide will run the top **Yes** branch. Otherwise, the bottom **No** branch is run instead. The checkmark and **x** icons that you see at the decision points indicate which path was followed. The checkmark indicates that the connected path was run (the condition was met), and the **x** indicates that the path was not run (condition was not met).

In addition to project prompt values, you can design conditions that are based on date or time values or on values that are read from a SAS data set. You can also combine conditions to create sophisticated decision points within your project. For example, you can design parts of your process flow that run only once a week on a specific day, or only when a certain data value falls out of a specified range.

OFF LIMITS: STUFF THAT DOES NOT WORK

Unfortunately, the world of SAS Enterprise Guide isn't completely utopian. There are a handful of SAS programming practices that do not work, at least not without a struggle.

X STATEMENT AND SYSTASK

Many SAS programs use the X statement and the SYSTASK function to escape out of the SAS program and perform some work in the shell of the operating system where the program is running. For example, these techniques allow you to copy files among folders, query the contents of directories, and run batch files or shell scripts.

The default centralized SAS environment disables use of the X statement and SYSTASK function. The reason is that in a centralized environment accessed by dozens or hundreds of people, these types of shell-level commands can represent a security risk and they can introduce instability. SAS Enterprise Guide makes it very easy for less experienced users to have access to your SAS environment. It might not be a good idea for novice users to have unfettered access to your system shell environment.

You can work around this limitation with the cooperation of your system administrator. It is possible to configure your SAS environment to allow these statements again, using the **ALLOWXMD** system option in the SAS startup command. However, use this approach with extreme caution and ensure that all involved understand the potential risks of *rogue* SAS programs.

DDE IS DOA

DDE, or Dynamic Data Exchange, is a 20-year-old protocol that Microsoft Windows applications can use to send messages and commands to each other. The SAS programming language includes a FILENAME statement access method for DDE to facilitate conversations between SAS for Microsoft Windows and other applications. For years, SAS programmers have used DDE to read and write data programmatically in Microsoft Excel worksheets. When the SAS program runs, it issues commands to start a Microsoft Excel process and establish a communication link, open workbook files, and access data in particular worksheet cells. It's interesting to watch such programs in action because Microsoft Excel windows pop up and values appear in cells as if they were typed in by an invisible hand.

DDE technology works only under certain conditions, and these conditions are often not met when using SAS Enterprise Guide. One condition is two processes that are communicating via DDE must be running on the same machine. In a distributed environment where SAS is running on a remote server, the version of Microsoft Excel on your local PC is inaccessible to your SAS program. Remember, the DDE link is between Microsoft Excel and SAS, not SAS Enterprise Guide. The remote SAS session might even be running on a system other than Windows, such as UNIX, where DDE isn't supported at all.

The second requirement is that the SAS session must run in a windowing environment. Even if your SAS session is running on a PC that has Microsoft Excel installed, the SAS session is running *headless*, which means that it has no visible windows. Without this window environment in place, DDE (which relies on Windows messages) is not effective.

SAS Enterprise Guide has built-in features to import and export data to and from Microsoft Excel, and you can use those features to gain back some of the ground lost without DDE. However, SAS Enterprise Guide doesn't offer the same level of control at the cell level as DDE.

NOWHERE TO SHOW: SAS/AF[®] AND %WINDOW

SAS/AF[®] is a legacy application development environment that is built right into SAS. Using SAS/AF components such as Frames and Screen Control Language (SCL), it is possible to build applications that drive SAS processes. The user interface appears dated compared to most modern desktop applications and web-based applications, but some companies continue to rely on their investment in these early full-screen applications.

Because of the client/server nature of SAS Enterprise Guide and SAS, SAS/AF applications are not accessible within SAS Enterprise Guide. These are full-screen applications hosted in SAS, and with SAS operating as a server there is no *screen* to host these windows. In fact, any SAS language feature that would normally produce a prompt or window within an interactive SAS session is off limits with SAS Enterprise Guide. This includes %WINDOW statements, PROMPT options on LIBNAME statements, and interactive environments such as the Report window and the DATA step debugger.

In general, SAS statements that require user interaction and that do not work well in a SAS batch program do not work well in SAS Enterprise Guide either.

Fortunately, SAS Enterprise Guide offers modern replacements for many of these interactive features. You can achieve much of the same experience, and more, through project prompt, the query builder, and built-in tasks. In fact, you can even extend SAS Enterprise Guide with custom tasks, fulfilling the needs served by SAS/AF programs for many years.

ENDING CONTROL WITH THE ENDSAS STATEMENT

In the world of sophisticated batch SAS programs, it is common practice to use the ENDSAS statement to control the program flow. The ENDSAS statement, as the name implies, ends the current SAS session. You might use this in a batch program to terminate processing when you encounter certain conditions.

However, in SAS Enterprise Guide the SAS session is your lifeline to your results and SAS log. If your SAS program executes the ENDSAS statement, it is comparable to hanging up the telephone before you have heard all of the important information. Your results become disconnected and are not retrievable from your SAS Enterprise Guide project.

Before you run such SAS programs with SAS Enterprise Guide, it's a good idea to rework the logic to avoid ENDSAS (as well as the ERRORABEND system option). Instead, you can change the structure, perhaps using macro statements (or the new conditions feature in SAS Enterprise Guide 4.2!), to conditionally execute just the code that you want instead of terminating the SAS session.

CONCLUSION

SAS Enterprise Guide enables you to leverage your SAS programming experience to the fullest. SAS Enterprise Guide gets out of the way as you write code, but it continues to provide services to help you manage the non-programming tasks of organizing your work. With the process flow, project prompts, the new conditions feature, and ordered lists, you have unprecedented control over how you run your SAS processes.

REFERENCES

McDaniel, Stephen, and Chris Hemedinger. 2007. *SAS for Dummies*. Hoboken, NJ: Jon Wiley and Sons, Inc.

ACKNOWLEDGEMENTS

Bill J. Gibson reviewed and contributed to this paper. Mike Porter, Stephen Slocum, and Lina Clover served as additional reviewers.

RECOMMENDED READING

Looking for a community of other SAS Enterprise Guide users, including SAS programmers? Visit the SAS Enterprise Guide discussion forum at <http://support.sas.com/forums>.

CONTACT INFORMATION

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

Chris Hemedinger
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Email: Chris.Hemedinger@sas.com
Blog: <http://blogs.sas.com/sasdummy>

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

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