

SAS® Programmer's Paradise: New Goodies in SAS® Enterprise Guide® 4.3

Audimar Bangi, Chris Hemedinger, Stephen Slocum, SAS Institute Inc., Cary, NC

ABSTRACT

Are you looking for a SAS® development environment that actually helps you to write and manage SAS programs? It's here! This paper describes the many new productivity enhancements for programmers in SAS® Enterprise Guide® 4.3. There is a new program editor with syntax completion for hundreds of SAS procedures and statements, support for parentheses matching, built-in function help and disambiguation, program "tidy" functions, and more. There is also a SAS code analyzer that can read your program and turn it into a process flow, or modify it to optimize the program steps to run in parallel on a grid computing environment. The SAS Enterprise Guide project file now supports relative paths, making it easier to manage your projects, programs, and data in standard source management systems.

INTRODUCTION

SAS Enterprise Guide 4.3 includes several new programmer productivity tools that will help SAS programmers of all skill levels to create and maintain SAS programs with greater confidence and efficiency. Whether you are creating new programs from scratch or simply maintaining existing programs, the new tools will provide you with greater insight into the syntax that you use and the flow of your programs.

GETTING MORE FROM THE SAS PROGRAM EDITOR

Traditionally, the program editor in SAS Enterprise Guide has been called "the enhanced editor", thanks to its ability (added in 1999) to color SAS keywords and constructs in order to provide you with instant cues as to whether your program is syntactically valid. As you'll see, the new program editor in SAS Enterprise Guide 4.3 takes these enhancements much further, going as far as actually helping you to write SAS programs by suggesting SAS programming keywords and statements. We could call it the "enhanced enhanced editor", although that doesn't seem like a terribly catchy name.

Note: The new program editor features are available only in SAS Enterprise Guide. The new features are not available in the Display Manager interface within the traditional SAS environment. SAS® for Windows continues to provide the current enhanced editor for keyword coloring.

SYNTAX SUGGESTION AND AUTOCOMPLETE

The biggest eye-catching improvement in the new program editor becomes obvious as soon as you begin to type. When you enter the first two characters of a SAS keyword (for example, a PROC statement or option), you are presented with a list of possible keywords that you can use to complete the word. To complete a keyword using one of the suggestions, simply highlight the desired keyword in the list and press the spacebar or the next valid punctuation (such as the period key, equals sign key, or semicolon key).

When the list of suggested keywords appears, you can navigate to the keyword that you want by:

- Continuing to type until the correct keyword is selected (since the matching improves as you type)
- Scrolling through the list using the UP and DOWN arrow keys, or the PgUp and PgDn keys
- Scroll through the list using your mouse, and then double-clicking to select the correct word.

Figure 1 shows two examples of the syntax suggestion feature within the FREQ procedure: one to match the TABLE statement, and one to match the CHISQ option of the TABLE statement.

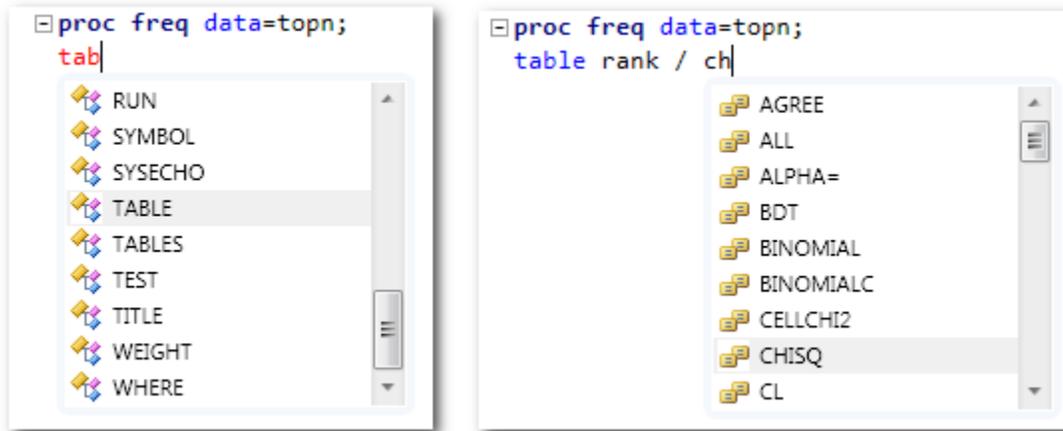


Figure 1. Two examples of the syntax suggestion feature within PROC FREQ

The program editor includes syntax information for over 200 SAS procedures and thousands of SAS statements, functions, and CALL routines. The syntax information includes the SAS macro statements and functions, which amounts to a programming language in itself.

INTEGRATED SYNTAX HELP

This feature is more subtle than the syntax completion, but programmers will find it tremendously valuable.

SAS Enterprise Guide displays brief SAS syntax documentation in a tooltip-style window when you:

- “Hover” the mouse pointer over a valid SAS keyword in your program
- Position the keyboard cursor within a valid SAS keyword and press the F1 key
- Begin to type a valid SAS keyword, and position the mouse pointer over one of the suggested keywords.

Figure 2 shows an example of the integrated help within the SET statement for the DATA step.

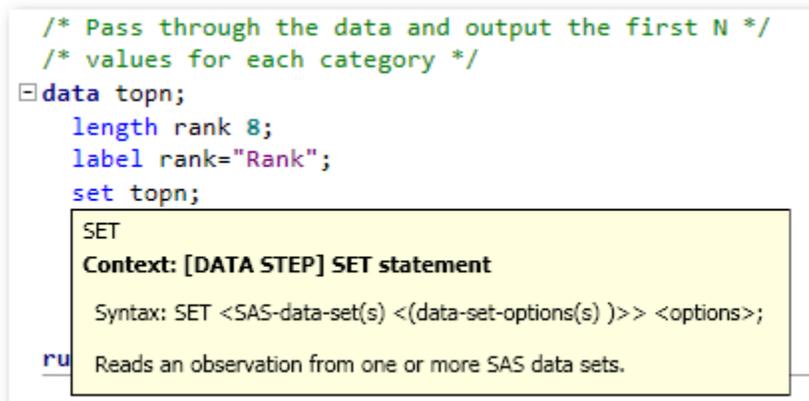


Figure 2. Example of integrated syntax documentation within the SET statement

The SAS language often uses keywords with the same name within different contexts. For example, the LABEL keyword is used within the LABEL statement, and also as part of the LABEL= data set option. The program editor can distinguish between these uses and displays the appropriate help text.

The SAS online documentation (available at support.sas.com) still provides more comprehensive usage information about the SAS language, but if all you need is a hint about the syntax or a brief description of a keyword, the integrated help will save you the trip to the Web.

FUNCTION COMPLETION AND DISAMBIGUATION

Many SAS functions have multiple function “signatures”, or versions of their syntax, to accept different arguments depending on their use. For example, the SUBSTR function can be used either on the left side or right side of an equals sign, depending on what you are trying to accomplish. The program editor can help you to resolve this ambiguity by present both versions of syntax and allowing you to select between them. It then helps you to complete the function reference by tracking the position of each argument as you type, and providing a hint as to the purpose of each argument just as you need it.

Figure 3 shows two examples of the integrated function help and disambiguation. The top example shows the SYMPUT CALL routine with its two arguments, the first of which is described in the help text. The bottom example shows how the program editor allows you to select among the three variations of the %SCAN macro function.

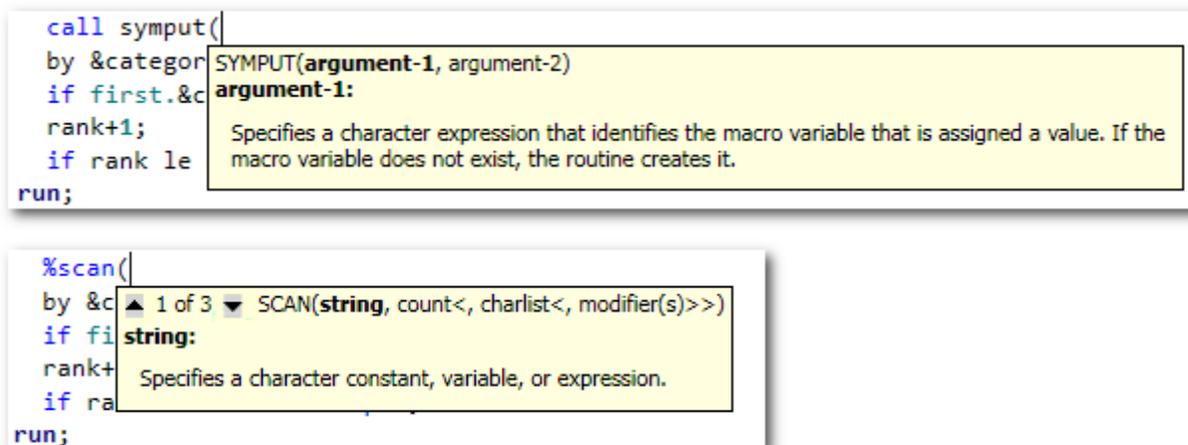


Figure 3. Examples of function help and argument tracking

INLINE MACRO VARIABLES AND MACRO FUNCTIONS

As we’ve seen, the program editor has a tremendous amount of syntax data built into it. But it also knows how to read your current SAS program and incorporate the program elements that it finds, adding them to the syntax suggestions that the program editor offers. The following program items are included within syntax suggestions:

- Macro variables that are defined using the %LET statement or SYMPUT CALL routine
- Macro routines that are defined using the %MACRO statement
- Data set names that are defined using the DATA step statement.

Figure 4 shows an example of how a custom macro routine is suggested as you type.

```
☐ %macro swap(  
    a /* first var in custom routine */,  
    b /* var to swap */  
);  
/* more code here */  
%mend;  
  
%swap(|  
    swap(a, b)  
    a:  
    first var in custom routine
```

Figure 4. Example of a custom macro displayed as a syntax suggestion.

Note that the program editor does not automatically list macro variables and routines that have already been defined and submitted to SAS (for example, in an autoexec file). It lists only those elements that are defined within the active SAS program.

AVAILABLE LIBRARIES AND DATA MEMBERS

As you type SAS statements that require data references (DATA= options), the program editor can display the SAS libraries that are available, as well as the data sets and views that are available in those libraries. This feature requires that you have a “live” SAS session at the time you are editing your program so that SAS Enterprise Guide can discover which data sources are available. To establish a SAS session, simply click the “connect” icon in the toolbar along the top of the editor window. (Note: there are other ways to connect to a SAS session. For example, you can open a data source in a SAS library, or navigate among your SAS server resources by using the server list.)

SAS Enterprise Guide issues dynamic requests to your SAS session to discover which libraries are defined (using their libref names) and which data sets are within them. If you define new libraries by submitting the LIBNAME statement, the new libraries automatically show up in the list. It might take a few seconds to generate these lists when you first connect to SAS, but subsequent accesses are usually much faster. If you already know the exact libref and member name you do not have to wait for the program editor to populate the list; you can simply keep typing and complete the syntax yourself.

Figure 5 shows two examples of the program editor providing data source suggestions: one for libraries (on the top) and one for data sets (on the bottom).

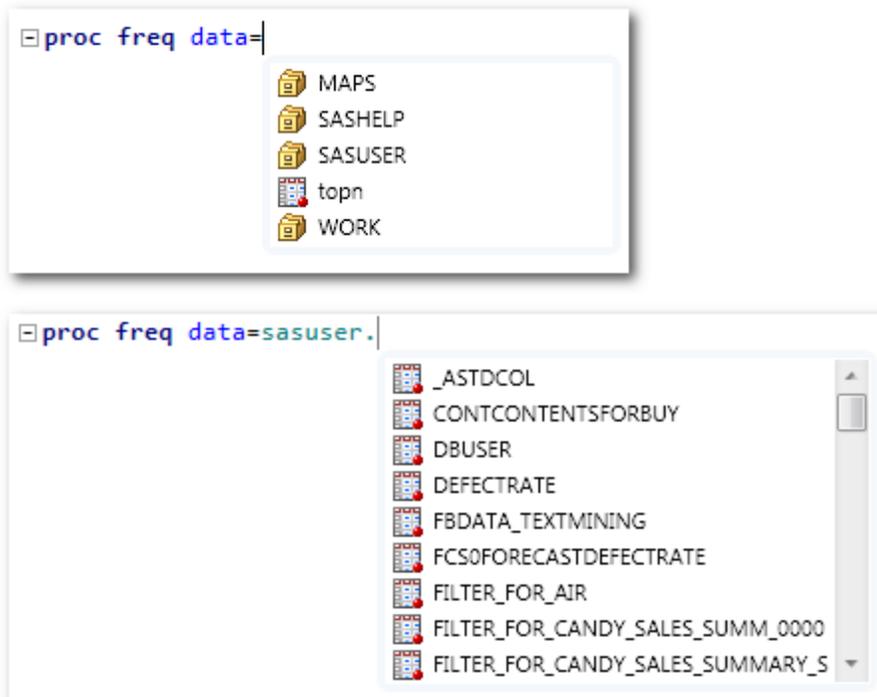


Figure 5. Data source suggestions within the program editor.

PARENTHESES MATCHING

If you've ever created a SAS macro program, you know what it's like to keep track of functions and arguments within deeply nested parentheses. The program editor now helps you to locate the correct matching parenthesis as you navigate a series of opened and closed parentheses. The editor highlights both the left and right parenthesis, and if you see only one side highlighted, you know that you have a mismatch to resolve. This behavior also applies to brackets (“[”, “]”) and braces (“{”, “}”).

Figure 6 shows an example of the highlight treatment when the cursor is positioned next to a matching parenthesis.

```
title "Top Models by MPG_City for each region of Origin";
%let newVar = %sysfunc(%str("SAS (nested functions)"));
```

Figure 6. Parenthesis highlighting shows the match ups.

ASSIGNING KEYS TO COMMANDS AND DEFINING MACROS

You might not have known it, but editor commands, macros, and abbreviations have been available in the SAS enhanced editor for years. In SAS Enterprise Guide 4.3, it's much easier to access the options to define and use these features.

The program editor surfaces over 100 commands that represent navigation and editing shortcuts. Many of these commands are already assigned to keyboard shortcuts. You can learn about these keyboard shortcuts, change them, and add others by using the **Program->Enhanced Editor Keys** menu item. Figure 7 shows the Enhanced Editor Keys window with a sampling of the available commands.

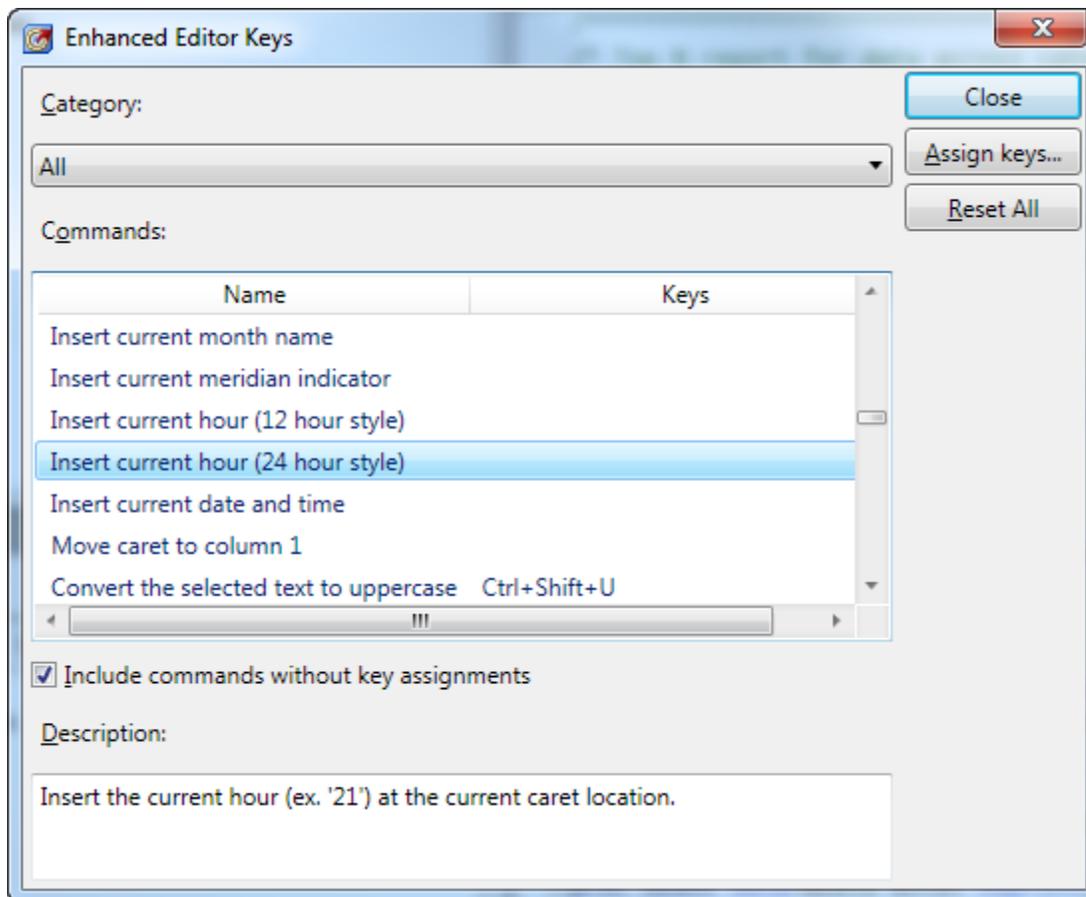


Figure 7. The program editor commands and their shortcut keys

You can also define a sequence of commands and text operations and store them in a named group called a “macro”. This is different from the SAS macro language, but it’s a similar concept. An editor macro allows you to reuse a sequence of actions that you use frequently. Use the **Program->Editor Macros** menu to record a sequence of operations or define a new macro from scratch. You can even assign a keyboard shortcut to an editor macro, and thus complete a complex custom operation with a simple keystroke.

And don’t forget about the **Program->Add Abbreviation** menu item. This option allows you to add your own autocompletion phrase that the editor will replace as you type. It’s a great way to automate adding a large block of text – for example, a standard comment header that you like to use with new programs.

CONTROLLING EVERYTHING WITH OPTIONS

Whenever we demonstrate these new autocompletion features to experienced SAS programmers, the first reaction is usually something like, “Wow, that’s awesome!” The very next reaction is usually, “I can turn that off, right?” Some experienced SAS programmers have an inherent mistrust of tools that seem to do much of the work for you. These programmers will be comforted to know that almost every aspect of the syntax completion features can be configured and optionally “turned off” by setting preferences in the SAS Enterprise Guide options. Figure 8 shows the Autocomplete options that you can control.

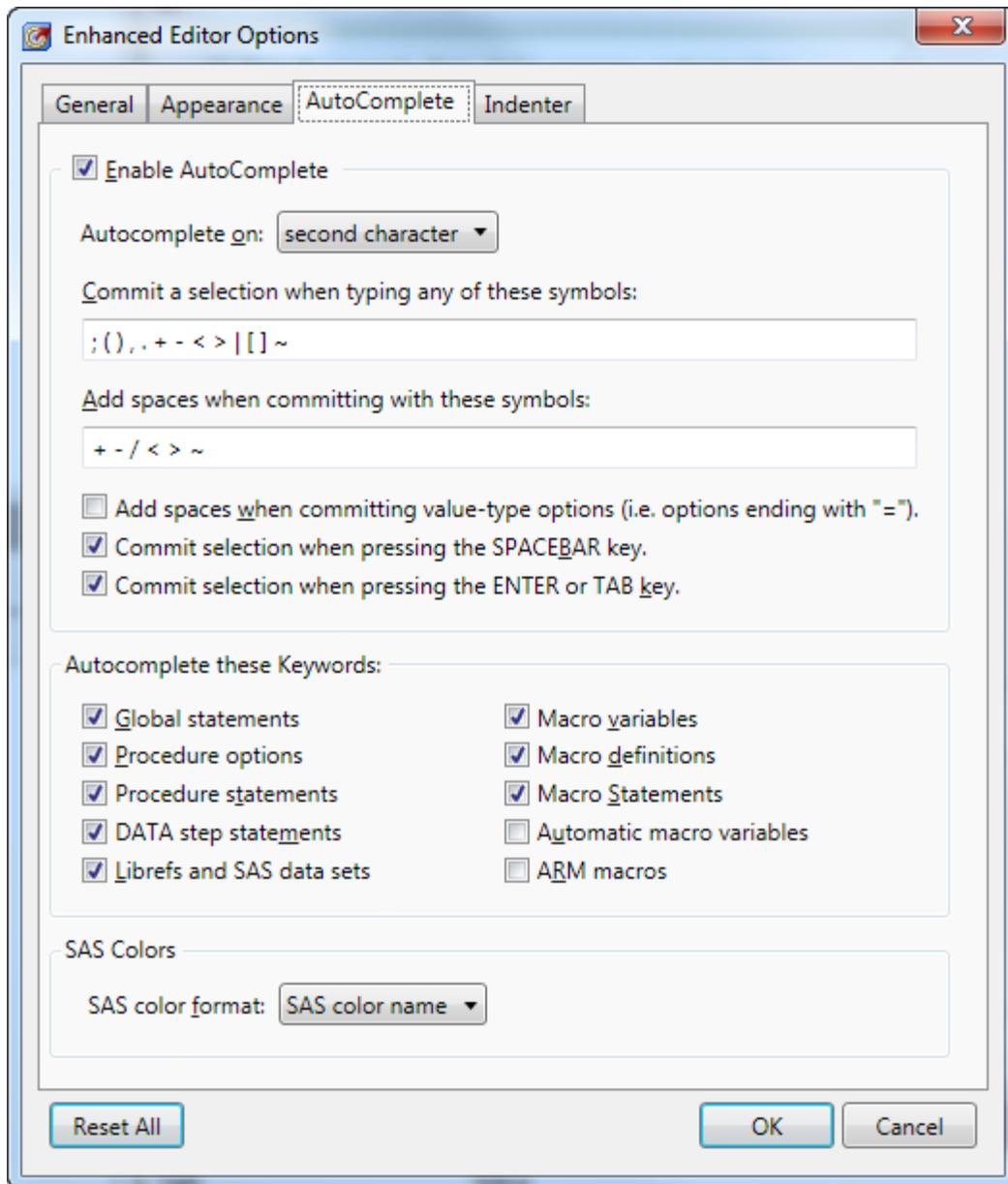


Figure 8. The program editor options for autocomplete

However, our guiding objective when adding these features has been to make sure that these enhancements do not “get in the way” of experienced programmers. Yes, you can disable these features if desired. But we believe that once you try them out, you’ll want to keep them enabled.

MAKING YOUR PROGRAM PRETTY

In most programming languages, programs are easier to read (and thus understand) when formatted with proper indenting and line breaks, and SAS is no exception. SAS Enterprise Guide provides a one-click method for reformatting your program to make it more readable. To format your SAS program, right-click within the program editor and choose **Format Code** (or use the Ctrl+I keyboard shortcut).

Many SAS programmers have their own preferred style for program appearance, and it’s possible that the SAS Enterprise Guide formatting rules don’t fit your preference. Some of the formatting behavior can be controlled within the editor preferences that you can access in **Program->Editor Options**. But if you reformat a program and find that

you don't like the result, it's easy to revert it back to what you started with. Simply click **Edit->Undo**, or use the **Ctrl+Z** keyboard shortcut.

Here is a simple example of a program that's difficult to read because it lacks indentation 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 using the Format Code feature in SAS Enterprise Guide, the program appears 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;
```

As a related feature, you can export your SAS program as HTML, which retains the indenting and keyword coloring. This is useful for teaching or documentation purposes where you might want to include a SAS program in a Web page or blog post. To produce an HTML representation of your SAS program, right-click in the program editor and choose **Export as HTML**. Or right-click and choose **Copy HTML to the Clipboard**, and then paste the contents into your favorite HTML editor.

WORKING WITH THE SAS LOG

For SAS programmers, the SAS log provides the important details about what went wrong and when, with possible hints for how to correct the problem. SAS Enterprise Guide already alerts you when you have "adverse events" during your SAS program that generate warnings or errors in the SAS log; it does this by placing red "X" or "warning triangle" icons on your program within the SAS Enterprise Guide process flow. To view the log contents, simply right-click on the program node and select **Open->Open Log**.

In SAS Enterprise Guide 4.3, the log viewer includes additional navigation shortcuts so that you can locate warnings and errors quickly. In the toolbar within the log window, use the up and down arrow buttons to find the previous and next warning or error line. As a keyboard shortcut, you can use **Ctrl+E** (to move to the next error) and **Ctrl+Shift+E** (for previous error).

Figure 9 shows an example of the Log window with the navigation buttons in place.

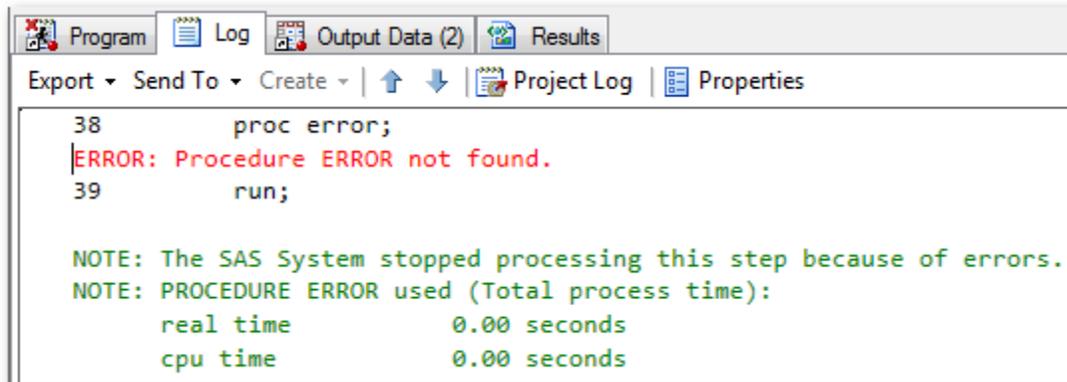


Figure 9. The Log window with buttons to move to the next and previous error or warning

ANALYZING YOUR SAS PROGRAM

Ever since the introduction of the process flow within SAS Enterprise Guide, customers have asked for the ability to view a complex SAS program as a set of logical nodes within a flow. It's now possible with the **Analyze SAS Program** feature in SAS Enterprise Guide 4.3.

SAS Enterprise Guide uses the SCAPROC procedure to compile and run a SAS program and then report on the details of program's effect, including which procedure and DATA steps are run and the file and data sources that are referenced and created. SAS Enterprise Guide then interprets the detailed report and creates a process flow representation of your SAS program. The result will be a new process flow with program nodes for each step, plus data nodes for each referenced data source or result. And they will be connected with arrows so that you can see how it all flows together.

When analyzing a program, keep the following in mind:

- SAS actually runs your program to analyze it, so the time it takes to complete the analysis is similar to how long it would take to run the program.
- If there are SAS errors that occur during analysis, you'll see this indicated in the analysis log. The analysis might not be completely accurate when errors are encountered.

In addition to creating a process flow from your program, SAS Enterprise Guide can help you to optimize a SAS program for use on the SAS® Grid Computing environment. The process is similar: SAS Enterprise Guide submits your program to SAS for analysis. However, instead of creating a new process flow, the result is a modified copy of the program that has special grid-computing statements inserted at key locations. These statements allow different parts of the program to run in parallel on a grid-enabled environment with multiple CPUs. Essentially, this allows SAS to "divide and conquer" the work in your SAS program and boost the performance.

ORGANIZING SOURCE FILES WITH RELATIVE PATHS

Customers often request for an easy way to use source management/version control systems with their SAS Enterprise Guide projects. The challenge is that a project file (with the .EGP file extension) is a single binary-formatted file that is not compatible with change management and comparison tools. However, you can maintain your important SAS content, such as data files and SAS programs, as files that are external to the project file. In this scenario, the EGP file is simply a recipe for how your work fits together, but the main ingredients are accessible on your file system.

In earlier versions of SAS Enterprise Guide, the project file stores external file references as *absolute* paths. This can present a challenge when you decide to move the project file, data, and programs for use within a different folder or machine. Before you can use your project in the new location, you need to work with the project maintenance tools to correct the file references.

In SAS Enterprise Guide 4.3, these external file references are now tracked as *relative* paths. This makes it much easier to pick up the entire bundle of files (EGP file, data files, and SAS programs) and drop them into a new folder or machine and have everything "just work". If you have a source management tool that integrates with your file system (for example, CVS or Subversion), then this behavior makes it much easier for teams to collaborate on SAS

Enterprise Guide projects and track version changes in the important assets (such as the SAS programs that you reference).

RUNNING A PROGRAM OR PROCESS FLOW AS AN AUTOEXEC

When you open an existing project, if you have a process flow in your project with a name of "Autoexec" (case does not matter), then SAS Enterprise Guide will run it for you automatically.

Here are the rules:

- Only one "Autoexec" flow per project will be considered.
- The "Autoexec" flow can contain anything: SAS programs, tasks, stored processes, and so on. It all runs as if you right-clicked on the flow and selected **Run Process Flow**.

There is a new option to control this behavior. By default, SAS Enterprise Guide will display a confirmation prompt before automatically running an "Autoexec" process flow. You can change the option setting so that such a flow will run automatically without prompting first.

The "Autoexec" flow can prove to be a convenient way to define project-specific libraries (for example, in a SAS program) or for staging data (for example, using the **Data->Upload Data** task) before you get to work by interacting with the project contents.

CONCLUSION

SAS Enterprise Guide is already well known for its point-and-click user interface and a wide array of tasks that can generate SAS programs for you. Over the years, SAS has been steadily adding features in SAS Enterprise Guide to appeal to SAS customers who work with established libraries of SAS code or who prefer to create SAS programs "from scratch", and to help these programmers to be more productive. The new features in the 4.3 version represent a big leap for productivity with the SAS language and programmer work flow.

REFERENCES

Hemedinger, Chris. 2008. "Find Out What You're Missing: SAS[®] Enterprise Guide[®] for SAS Programmers." *Proceedings of the SAS Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc. Available at support.sas.com/resources/papers/sgf2008/eguide.pdf.

RESOURCES

The SAS Enterprise Guide product support page contains links to all official documentation, samples, and SAS notes. Visit the page at: <http://support.sas.com/eguide>.

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>.

You can also find tips and articles about SAS Enterprise Guide and other SAS topics at "The SAS Dummy" blog: <http://blogs.sas.com/sasdummy>.

CONTACT INFORMATION

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

Chris Hemedinger
SAS Campus Drive
SAS Institute Inc.
E-mail: chris.hemedinger@sas.com
Blog: <http://blogs.sas.com/sasdummy>

Audimar Bangi
SAS Campus Drive
SAS Institute Inc.
E-mail: audimar.bangi@sas.com

Stephen Slocum
SAS Campus Drive
SAS Institute Inc.
E-mail: stephen.slocum@sas.com

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.