

## SAS Enterprise Guide® – Implementation Hints and Techniques for Insuring Success with Traditional SAS Programmers

Roger D. Muller, Ph. D., Data-to-Events, Inc, Carmel, IN 46033

### ABSTRACT

Traditional SAS programmers develop SAS code in files and submit it for processing regardless of the operating environment (PC, Unix, mainframe, etc.). SAS Enterprise Guide (EG) follows this model, but adds some unique additional capabilities. This paper addresses setup, initialization and workflow ideas to smoothen and enhance the transition to the EG graphical user interface centered around a process flow window. This paper will address local vs. remote (server) processing. Starting with hardware and network capabilities, the paper then moves into a discussion on data location and how that affects work flow. Certain features in SAS EG may either be hidden or exposed with advantages to doing either. The number of process flows in an Enterprise Guide Project offers flexibility in constructing the total programming effort. The ability to store code at a central facility as Stored Processes for sharing with other users will be discussed. Techniques for submitting and developing code for step-by-step processing vs. the submission of entire project files is discussed. Backup strategies will be discussed. And lastly, the handiest key in EG, the F4 key, which is used to toggle back-and-forth between the current Process Flow window and the most recent window (program, data set, log, output, etc.), will be repeatedly emphasized.

### INTRODUCTION

The first step in assessing how you are going to use Enterprise Guide is to look at your job role, your skill set and the SAS computing environment. If you have access to Enterprise Guide, the conversion to a workflow that incorporates EG use as all, or as a part of, your day-to-day activities should be easy. In all SAS systems, code is developed and/or maintained in file-based systems. Behind the scenes, every operation in Enterprise Guide is done with SAS code that can be captured and retained in “.sas” files that can be run in any SAS environment.

### ASSESSMENT OF YOU AND YOUR COMPUTING ENVIRONMENT

**LET'S FIRST LOOK AT SOME POTENTIAL SCENARIOS (TABLE 1) FOR YOUR COMPUTING ENVIRONMENT (TOPOLOGY):**

1. **Unix Server, Windows Server or Mainframe with SAS Being Installed Only on the Remote Device. (neither EG or PC-SAS is installed on local PC workstations).** You do not have an environment that supports EG. Sorry, but you are “out of luck”. Your PC is serving only as a “dumb terminal” to the remote environment.
2. **PC-SAS and EG on User's Local Workstation PC.** You have a platform on which you can use all the capabilities of EG in a local environment. The accessing of data from other environments would follow your standard workflow for PC-SAS used in the past.
3. **PC-SAS and EG on User's Local Workstation PC. A Unix Server, a Windows Server, or a Mainframe with SAS Installed is on the Network, But the Integration Technologies Option Is Not Installed on the Remote Machine.** Again you have a platform capable of using all facilities of EG in a local environment. If you have SAS/Connect installed, you have the capability of using RSubmit to remotely submit jobs or parts of jobs to the remote device from within EG, and full capability of returning outputs back to your EG session. Most installations with both SAS on servers and SAS on local PC's have SAS/Connect installed. See Muller (2012), and Muller and Penix (2012) for documentation on how this is done. While this is not “state-of-art” and dates back over 20 years, it does provide a workflow with a long proven track record. This RSubmit technique avoids heavy network data traffic and unnecessary potentially dangerous duplication of data.
4. **PC-SAS and EG on Workstation PC. A Unix Server, a Windows Server, or a Mainframe with SAS Installed is on the Network, the Integration Technologies Option Is Installed on the Remote Machine.** This is the best environment that you can have. From your Windows PC EG session, you can run code on the server, using your EG session essentially as an intelligent graphics terminal. When you need, you can switch back to your local machine as the processor. Data movement between the machines is minimal with no need to duplicate data on multiple machines to have the fastest access. Generally the computing processes are being run on the largest processor with the most memory.
5. **EG Only on the Workstation PC. A Unix Server, a Windows Server, or a Mainframe with SAS Installed is on the Network, the Integration Technologies Option Is Installed on the Remote Machine.**

With this topology, all SAS processing takes place on the remote station, but you have the convenience and features of a strong graphical interface with all of its benefits residing on the intelligent workstation. Some sites like this configuration as updates and maintenance are easier and can reduce licensing costs. You must be connected to the server to process anything.

As you can see, there are various topologies that provide you with the conveniences of EG. Some are stronger than others, but all can achieve the same goals providing you know how to work around their features and/or limitations. Even with a minimal EG installation, you can enhance your workflow significantly.

No.	PC Software	Remote Software	Implications
1	None	SAS only	You can't use EG
2	PC-SAS and EG	No SAS	Local use only, remote data access possible through libname statements, or data file transfer
3	PC-SAS and EG	SAS, Integration Technologies <b>not</b> installed	Local use plus you can use "Rsubmit" feature of SAS/Connect to process remotely in a "batch" type of mode
4	PC-SAS and EG	SAS, Integration Technologies installed	EG can run locally or can operate interactively with remote machine. This is by far the best configuration
5	EG Only	SAS, Integration Technologies installed	PC EG session acts to run jobs on remote server. Preferred by some for reduced licensing and maintenance costs. You must be connected via networking. PC is functioning as an extremely intelligent graphics terminal.

**Table 1. Topology (SAS Machines and Licensing) Scenarios and How They Affect Programmer Interaction With EG and SAS Processing.**

**NOW LET'S LOOK AT YOUR JOB ROLE AND WHAT YOU DO:**

- Maintain and Run Validated Production Code on a Server.** There would be occasions where you would benefit from having access to some of the functionality of EG for "ad hoc" types of inquiries, but you might legitimately question the investment in time taken to learn these capabilities in EG.
- Develop Code to Enhance Production Code for the Server.** If you have EG, learn to use it. Follow your standard workflows for any PC-SAS efforts in the past. All code you develop can be exported to other environments.
- Do All Sorts of "ad hoc" Inquiries of Data, Reports and Processes Associated With a Production Server.** Learn EG! This is particularly true if you are doing "look-see" kind of efforts that do not directly result in submission tables, listings etc., but rather give you a "behind the scenes" view of the operation with minimal programming time.
- Do Quality Control Validation Against Results Produced by Production Servers.** Use EG. Develop your QC validation code interactively in EG and export the code to the permanent servers where validation code is stored and run.
- Complete ad hoc reporting requests to find out "What's in the data".** You may be doing this for exploratory purposes for yourself or a scientist you are supporting. Learn EG, minimal programming will produce maximum results quickly and accurately.
- Do you work alone, or in a team where code development is a shared responsibility?** In a single user environment EG is valuable. In a team environment, it is even more valuable as certain processes can be shared on a Stored Process Server. Learn EG and make sure your SAS installation rep gets a remote server with Integration Technologies installed. For the long-term classical file-based SAS programmer, think of stored processes as SAS code stored as macros and placed into an environment on a remote shared server where librarian types of functions such as check-in, check-out and updating are maintained with security controls.

**SAS PROGRAMMING SKILLS**

Actually, there is nothing new to learn from a pure programming perspective. This may seem like a very curt answer, but the same SAS code as always is being generated to be submitted to SAS for processing. There are, however, many ways to improve this via the expanded workflow. Training needs are minimal.

## SAS WORKFLOW PROCESSES

All of the techniques to be learned regarding workflow with Enterprise Guide allow you to work faster, easier and smarter once you have invested the time to learn the workflow. Furthermore, there will be many new boundaries for you to push and enhance the quality of the product you deliver.

## HOW ENTERPRISE GUIDE WORKFLOW DIFFERS FROM TRADITIONAL – THE PROGRAM FLOW

The balance of this paper covers workflow that is different from traditional file-based flows. To point out some of the key differences in using Enterprise Guide, we will work thru just a few initial screen captures from the user interface at startup. Specific items will be numbered on screens and explained in the text.

Since most long-term SAS programmers are working from text-based SAS code files, the drag-and-drop interactive interface aspects of SAS EG are all but ignored in this paper. The opening of existing SAS datasets by opening them into the Process Flow is ignored. Dragging-and-dropping of programming icons from the Task List to the Process Flow will be ignored. For a more complete discussion on these techniques, see Muller (2012), and Muller and Penix (2012), and other references at the end of this paper. These are important features and techniques, but their values many not be immediately grasped by traditional programmers

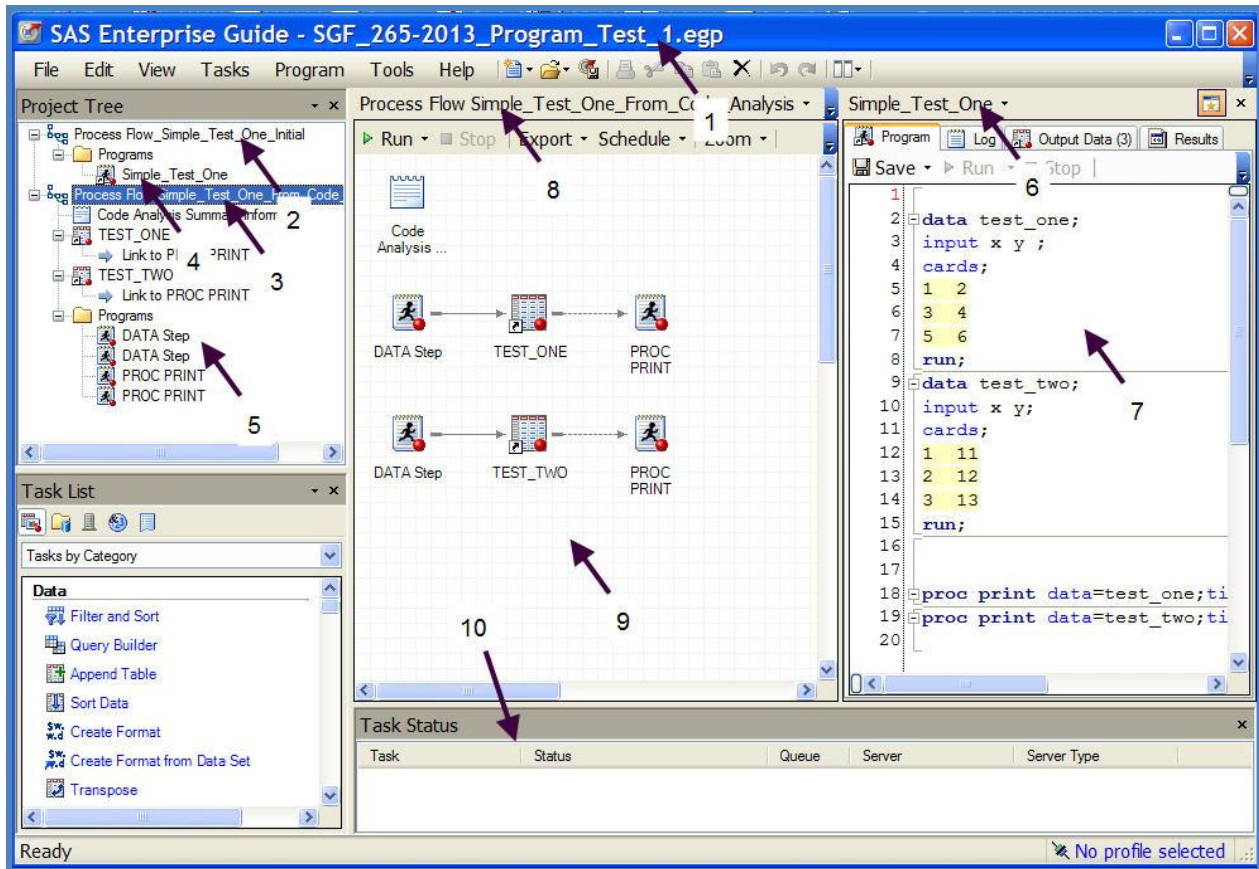
This paper covers, for the most part, file-based programming actions in order to most closely parallel the on-going day-to-day actions of long-term programmers. For purpose of discussion, the terms “process flow” and “program flow” are used interchangeably through this paper to refer to the content of the diagrammatic window.

## THE INITIAL SCREEN ONCE EG IS LAUNCHED AND A PROGRAM HAS BEEN RUN

To create Display 1, Enterprise Guide was launched. A new project was started (File, New Project). A new process flow was started (File, New, Process Flow) and a new program was coded (File, New, Program). The program could as well have been one that preexisted prior to this launch. This program was then ran. **The concepts of project files, process flows and programs are all covered in Muller (2012) and Muller and Penix (2012) and will not be discussed further in this paper.** The goal of this paper is to surface techniques and settings useful to the traditional file-based programmer to get up and running in the EG workflow.

The project name is noted in Display 1 (Item 1). There are two Process Flows (Items 2 and 3) shown in the Project Tree Window. The initial simple SAS program is shown in the project tree (Item 4). The actual code in this file is shown in a split window of the Process Flow (Item 7). Notice that the file name is shown as Item 6 at the top of that split window. The name of the process flow being displayed shown at the top of that window (Item 8).

**What’s New and Different?** What’s new to the traditional file-based SAS programmer? (1) Everything is stored in a file of type “\*.EGP” which is an Enterprise Guide Project File. (2) The highly visual Program Flow diagram. **STRONG HINT: To get used to switching in and out of the diagram from other windows such as a SAS code window, an output file, or a log, use the F4 key to toggle to-and-from.** Having this window so convenient gives you an appreciation for its functionality. (3) The Task List and associated activities at the lower left (which are not the primary subject of this paper). This area gives you access to data and assorted programming by drag-and-dropping into the program flow window.



Display 1. The Enterprise Guide Startup Window After Running an Existing SAS Program.

**Workspace Splitting.** The workspace shown in Display 1 is not the default. It was split by using the menu commands shown in Figure 1. The “Side By Side” option works particularly well when the EG display is stretched over a dual monitor configuration and allows for two simultaneous operations to be viewed. It is wonderful for editing a file.

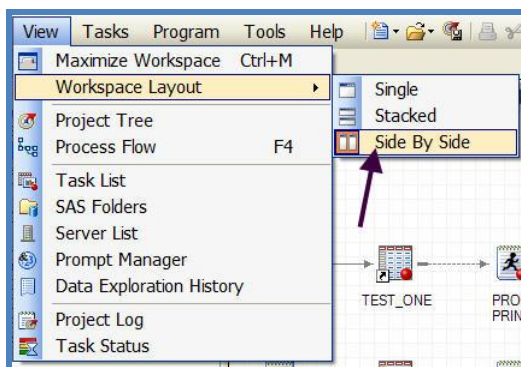


Figure 1. Splitting the Workspace.

**PC Computing Equipment and Display.** Enterprise Guide is very flexible in adapting to and optimizing the resources available. In general, the more “screen real estate” you have, the greater the flexibility and ease

of use. As shown in Display 1, the workspace splitting could crowd small screens. In addition, an enhanced code editor window can be further split (Muller, 2013a).

**Creating Process Flows from Existing Code.** In Display 1, there is a second process flow (Item 3). This process flow was created from the SAS Code file (Item 6) by choosing a tab (hidden from view to the right of the tabs displayed) “Analyze Program”, “For Process Flow”. This resulted in the generation of the second process flow (Item 3) and the SAS code files in it. **SAS code files run in the EG environment produce a much more detailed process flow diagram if they are subjected to this technique** (results shown as Item 9 in Display 1). The analysis and “flowing” of existing code became available with the release of EG 4.3.

**The Task Status Window.** When a job is running, detail about what is happening on a step-by-step basis is shown in the Task Status Window (Display 1, Item 10). This is job monitor that allows you to detect system-lockups, poor performance, etc. and also allows you to stop the processing that is occurring in a method that is far more eloquent than Cntl-Alt-Delete. **Do not turn this window off.** It provides maximum benefit for a very small amount of screen real estate.

## FLEXIBILITY TO CHOOSE THE OUTPUTS YOU WANT GENERATED

Enterprise Guide makes generating the types of outputs needed for the task at hand very easy. You then assemble these outputs into more complete reports. To pick one or more outputs to be generated after every proc, go to Tools, Options (Figures 2 and 3 ) and select one or more of the output types. As a side note, most other options for EG that are available here are best left to default when you are getting started with EG workflows. This is the only “EG” option discussed in this paper. Types of output include straight text, a new generation SAS report, HTML, RTF and PDF. Multiple selections can be made to generate multiple outputs for differing uses.

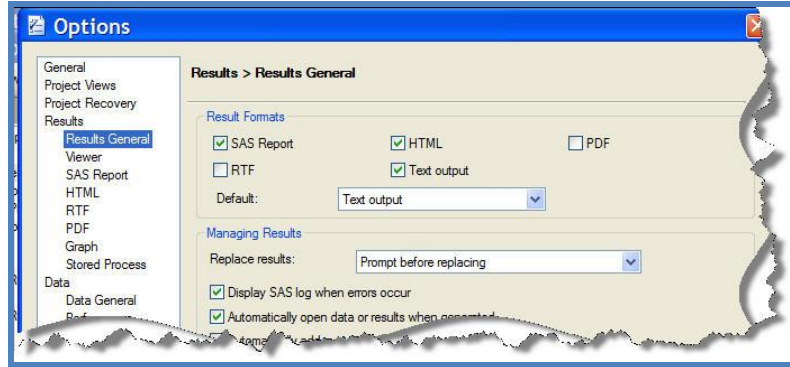
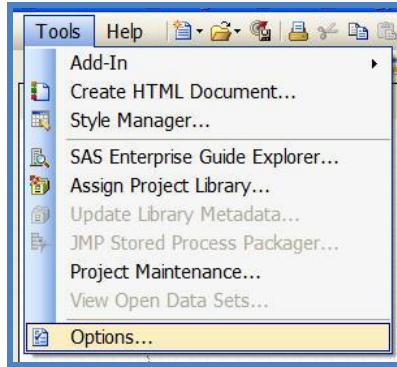
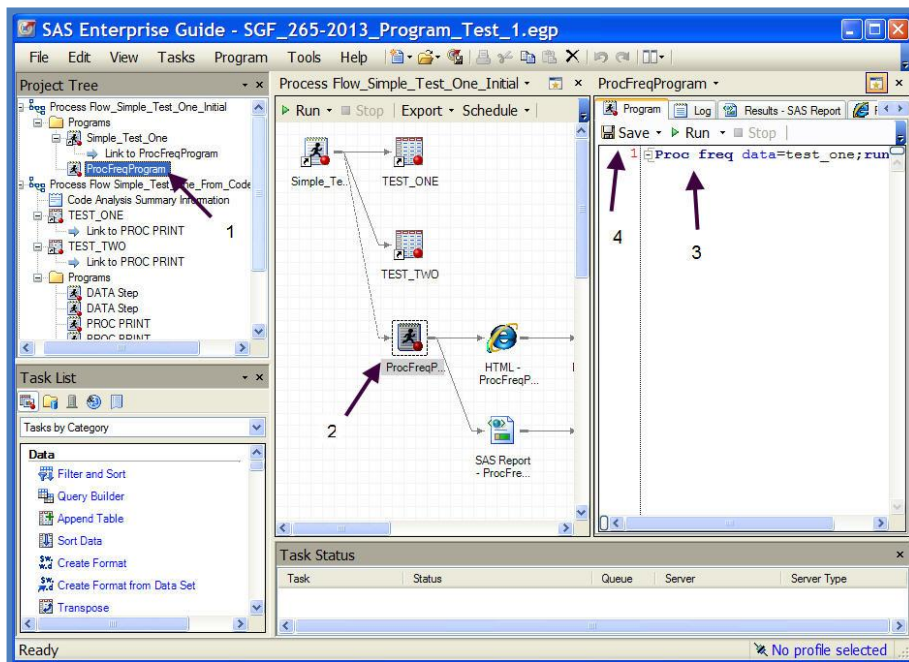


Figure 2. Selecting Options for EG. Figure 3. Select As Many Output Result Types As You Wish.

While all of these options for output exist in SAS without EG, the use of EG makes selecting and deselecting much more convenient.

## THE RUNNING OF A MORE COMPLEX PROCESS FLOW, GENERATING REPORTS



In Figure 4, a minor adjustment was made in the process flow in Display 1. The frequency code file is linked to the original “Simple\_Test\_One” program (Details not shown), the entire process flow was then run, resulting in some listings from Proc Prints (icons are not showing) and some output from the Proc FreqProgram. These are shown to the right of item 2.

the Process Flow Is Run.

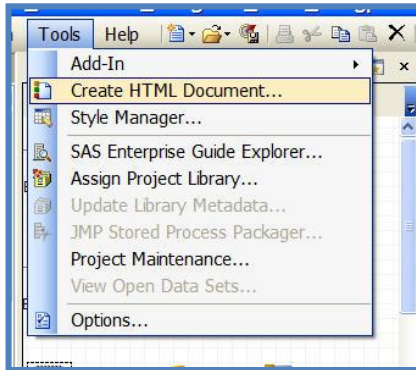
Figure 4. EG Display After

## WHERE’S THE “LISTING” FILE?

What do we do with these outputs? Yes, we can interactively look at them and do individual printouts, etc. In most other SAS workflows, outputs accumulate into one big “listing” file and you simply print or edit it. While these outputs

on the process flow can be processed/printed one at a time, we may want to make a more complete cumulative report where at least some of them are “assembled” into a combined entity. A wonderful option with this workflow is that some of the outputs may be ignored for the time being and be available at some other time should they be desired.

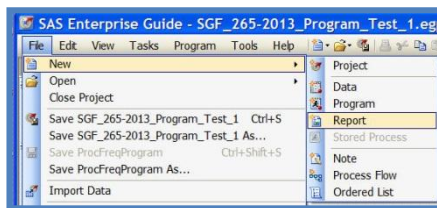
## ASSEMBLING AN “HTML REPORT”



Complete details of making one HTML file from output elements are not shown here. Start in Figure 5 by using the menu commands “Tools, Create HTML Document...”. The process essentially involves interactively selecting any or all of the HTML outputs and arranging them into a final document. The end result is an icon showing on the process flow which can be double-clicked and the consolidated document viewed in a browser.

**Figure 5. Creating an HTML Document.**

## ASSEMBLING A REPORT



A “report” document similar to that created as a “listing file” in traditional SAS file-based program operations is created in similar fashion. Select “File, New Report”. The steps to be followed are fairly straight-forward (See Muller 2013 a). Outputs on the program flow can either be used or ignored. All of the actions result in programming code that can be used/modified in the future.

**Figure 6. Getting Started With a Report Document.**

## MANY WAYS TO SELECT AND RUN CODE

The selection of SAS code to run could be the subject of a paper in its self. Most of a programmer’s development efforts involve running portions of the code. This is often very labor intense in that sections of code need to be tediously commented out. Here are a few options to “run” code:

- Run the entire project
- Run a process flow
- Run a branch
- Select a node or several nodes on a branch
- Open the file editor for a program and either run the entire program, or select code sections with the mouse and hit the right mouse button to run.

The actual “Run” command, may be at the top of a window, may be in a pulldown associated with a menu or may be on a right mouse button series of commands that surface.

## DOCUMENTATION WITH NOTES FEATURE

Code documentation has traditionally been done via comments in file-based workflows. This commenting feature can also be used in code developed in an EG workflow. A new capability, the ability to insert a Note has been added to the workflows. These do not reside within the traditional SAS program file and are a separate entity (object) within the EGP project file. Assorted information can be placed in these including directions to other programmers, difficulties being encountered, pasted in content from the Windows clipboard and more. Unfortunately, at the current

time, hyperlinks to other documents and URL's are not supported. The important benefit: these are not comments cluttering up the program code. They do not export out with SAS code. They are simply part of the project file (see Muller 2013a and Muller 2013b).

## IMPLICATIONS OF DATA LOCATION

Many different topologies were presented in Table 1. The following items also should be considered when evaluating how to establish your workflow.

- Data ideally is processed on a machine where both the data and SAS reside, or the data should be very conveniently accessible to SAS.
- Data movement across networks should be minimized for the sake of efficiency and time. This is particularly true for remote networked PC's.
- Data duplication should be avoided, not only for efficiency reasons, but for version control issues, security issues and more.
- Processing should take place on the largest processors with the most memory.

For these reasons, remote SAS servers with Integration Technologies installed are highly favored.

## THE ADVANTAGES OF EG WORKFLOW FOR THE TRADITIONAL SAS PROGRAMMER

The following list of 17 benefits has been developed to better give the traditional SAS programmer a listing of key benefits to be obtained when adapting an EG-based workflow:

1. Familiarity. The entire process is very similar to existing interactive SAS programming. There are program files, logs, lists, graphics, outputs, etc.
2. Project Organization: There is the new overarching concept called a SAS Enterprise Guide Project. This Project is a SAS file with an extension of ".egp" and contains many SAS programming elements: SAS programs themselves, the coding to do the process flows, comments and much more. The Enterprise Guide Project is composed of Process Flows which in turn are composed of Branches and Nodes. In some ways, the project file provides some of the benefits of a traditional folder in the computer's file management system as a "container of content".

WORD TO THE WISE: AS YOU CAN WELL IMAGINE—WITH A PROJECT FILE, YOU ARE "PLACING ALL OF YOUR EGGS IN ONE BASKET". BACKUP FREQUENTLY AND BACKUP TO MULTIPLE LOCATIONS. Protect yourself from your own bad programming and from systems failures. Backup as frequently as needed. If a project file is lost, you have to recreate what you have done since the last available backup.

3. Visualization: The Process Flow window can be used in several ways to provide an easy way to understand the relationships of program nodes, branches, datasets, etc. The window can be toggled between diagrammatic program flow and other elements such as text-based code files by simply **toggling with the F4 key**.
4. Run. The "run" step process consists of running anything from an entire project file down to running a single node, all at the ease of a few button clicks – make the selection and hit run. In addition, lines of code in the program editor can be selected and submitted separately. All of this leads to a much more efficient program development process as small elements can very rapidly be selected, run and checked for correctness or errors. All of this is accomplished without commenting out code.
5. Editor. The program editor is the best SAS has to offer.
6. Documentation. Note fields can be added to easily to document the project (comments, concerns, general messages, etc.).
7. Output. Output can be directed to any or all of a number of types of files at the click of a button. These include text, RTF, PDF, HTML and others.
8. Output Usage. Selected portions of output can be extracted for printing, email, etc.
9. Output to Office. There are many options to extract/send content to Microsoft Office products.

10. Data Query. Datasets can be examined, queried, etc. interactively right in the development environment. This provides for an excellent quick check of data quality issues.
11. SQL Development. The interactive query facility can readily generate PROC SQL code for use in programs.
12. Training. Training needs are minimal for the competent traditional SAS programmer.
13. Help. The editor has context sensitive help.
14. Code Export. All, or part of the elements, in an entire EG project can be exported to a single SAS file for running on SAS in the EG, PC, Unix, or other environment. This code can be shared with other programmers.
15. The Real Benefit! Data quality will improve and programming quality will improve as the programmer has many opportunities to interactively work with original and intermediate datasets, reporting steps etc. The programmer can see not only errors, but gains a far better understanding about the underlying data and business processes, thus making a better finished product.
16. Stored Processes. Stored process can be created. Stored processes are macros stored in a centralized facility and can be used by other programmers, Microsoft Office Users, etc.
17. We have just listed 16 benefits for experienced SAS programmers and we haven't mentioned "drag-and-drop" or "point-and-click" operations. These features are also useful to the experienced programmer! Of particular note are the over 90 tasks that have been developed to work with an assortment of SAS challenges. Lots of functionality is being "left on the table" if these are not used by pasting them into the process flows.

## CONCLUSION

SAS Enterprise Guide is a tool along with its workflows that should be adapted by nearly all SAS programmers. While there are a few SAS programmer's whose job role might not be enhanced substantially by investing in the minimal learning time, this paper showed 16 reasons why most programmers should adapt it. The click-and-point drag-and-drop interface which such programmers generally tend to dismiss is a 17<sup>th</sup> reason!

So, the answer the question posed by: "What's In It for the Long-Term Highly-Experienced SAS Programmer?" ---- Lots!

## REFERENCES

- Muller, Roger D. Assorted Videos on SAS Topics Including SAS Enterprise Guide.. Available at: [SAS Video Playlists by Roger Muller](#)
- Muller, Roger D., 2012. "Program Development with SAS Enterprise Guide® and SAS/Connect® in a Combined PC and Unix Environment" Proceedings of PharmSUG-2012. Available at: <http://pharmasug.org/proceedings/2012/AD/PharmaSUG-2012-AD17.pdf>
- Muller, Roger D. and Donald L Penix, 2012. "SAS Enterprise Guide® - Why and How the Programmer Should Adapt It Now" Proceedings of the 2012 Midwest SAS Users Group. Available at: <http://www.mwsug.org/proceedings/2012/BI/MWSUG-2012-BI08.pdf>
- Penix, Donald L. and Roger D. Muller, 2012. "Enterprise Guide® – Moving Beyond Your Initial Startup". Proceedings of the 2012 Midwest SAS Users Group. Available at: <http://www.mwsug.org/proceedings/2012/BI/MWSUG-2012-BI10.pdf>
- Muller, Roger D., 2013a "SAS Enterprise Guide® -- Implementation Hints and Techniques for Insuring Success With the Traditional SAS Programmer". Proceedings of the 2013 SAS Global Forum, Paper 265 (Video and Proceedings).
- Muller, Roger D., 2013b "SAS Enterprise Guide® -- What's in it for the Long-Term Highly-Experienced SAS Programmer?". Proceedings of the 2013 SAS Global Forum, Paper 274 (Video and Proceedings).
- Muller, Roger D., 2013c "SAS Enterprise Guide® --Implementation Hints and Techniques for Insuring Success With Traditional SAS Programmers in a Pharmaceutical Development Role". Proceedings of PharmaSUG 2013, Paper PO20.



## ACKNOWLEDGMENTS

Mr. D.J. Penix, President of Pinnacle Solutions Inc., Indianapolis, IN has been very helpful in developing the author's SAS BI expertise.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. The author has core competencies not only in SAS, but in videography and technical writing. **Inputs on the future use and direction of video for SAS Training and Documentation are sought.** Contact the author at:

Name: Roger D. Muller  
Company: Data-To-Events, Inc.  
Address: 14475 Stephanie St.  
City, State ZIP: Carmel, IN 46033  
Work Phone: 317/846-5782  
Mobile: 317/985-0132

E-mail: [rdmuller@hotmail.com](mailto:rdmuller@hotmail.com)

Web: [www.data-to-events.com](http://www.data-to-events.com) [www.rogermullervideography.com](http://www.rogermullervideography.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.