

Boost Your Programming Productivity with SAS® Enterprise Guide®

Chris Hemedinger, SAS Institute Inc., Cary, NC

ABSTRACT

SAS Enterprise Guide (with its easy-to-use interface) has been a boon to SAS users, especially non-programmers. However, SAS Enterprise Guide provides some powerful new features that are geared toward the SAS programmer, including an integrated process-flow view of your project; the ability to create, publish, and execute stored processes; and increased flexibility for output format and destination with the SAS Output Delivery System (ODS). This paper provides an overview and detailed examples of how SAS programmers can leverage SAS Enterprise Guide to create, organize, and execute their SAS programs.

MORE THAN SYNTAX

It's easy to see the value of SAS Enterprise Guide for those who are not familiar with SAS or SAS programming. As a front end to the SAS System, SAS Enterprise Guide provides dozens of tasks and wizards that enable users to generate SAS programs without having to write any code. This offers a tremendous advantage for the novice, but what value does it provide to the experienced SAS programmer? The answer is: **Plenty**—especially when you consider that knowing syntax and writing code is just part of the world of SAS Programming.

SAS Enterprise Guide continues to offer the features that you expect from a SAS programming interface: integration with the syntax-savvy Enhanced Editor; easy integration with metadata and administered data sources; the ability to run multiple programs simultaneously across multiple SAS sessions; and a task status window that enables you to track job progress. The latest version of the product offers even more. This paper discusses how to use SAS Enterprise Guide to:

- Organize your project, including the process-flow view and your SAS programs
- Combine your SAS knowledge with built-in features
 - Augment built-in tasks with your custom code
 - Use custom ODS destinations
 - Take advantage of SAS macros provided by the SAS Enterprise Guide environment
- Work with stored processes (create, modify, execute)
- Automate your processes so that you can run them unattended
- Extend SAS Enterprise Guide by adding your own custom tasks.

ORGANIZE YOUR WORK—THE SAS ENTERPRISE GUIDE PROJECT

SAS Enterprise Guide uses a project metaphor to help you organize your work. You can use the project file as a place to store your SAS code (or references to SAS program files), references to data, notes, and results. The integrated process-flow view provides a convenient way to view your project's content and understand its organization. Look at the simple project that's shown in Figure 1.

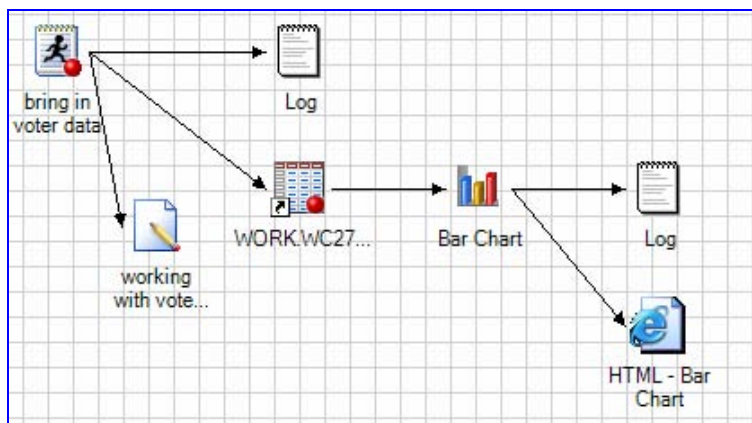


Figure 1. An Example Process Flow in SAS Enterprise Guide

At a glance, it's easy to understand the work of this project at a high level: there is a SAS program ("bring in voter data") that produces the output data set WORK.WC27. That data is then used as input to a Bar Chart, which produces HTML output ("HTML – Bar Chart"). The SAS code item has a note attached that offers more explanation, if needed. In addition, you can easily examine the logs from the code item or the bar chart (with code generated by SAS Enterprise Guide) to see those details that appeal to the SAS programmer.

FLEXIBILITY IN WORKING WITH SAS PROGRAMS—HAVE IT YOUR WAY

In the preceding example, the SAS code item ("bring in voter data") is stored directly in the project file. But suppose you already have a collection of SAS programs that are stored on the file system that you want to use? No problem—you can add those programs to a project and run them from there. The project contains a reference to the external SAS file; you can make changes in SAS Enterprise Guide and save them back to disk. Also, any changes that you make to the SAS program outside of SAS Enterprise Guide will be reflected the next time that you run the process. If you ever want to pull the program into the project so that it is completely separate from the copy on your disk, you can do that with a couple of clicks by adjusting the properties of the code item in the project. In addition, you can go the other way too—you can convert your embedded SAS program into a SAS program file that is referenced from the project.

WORK WHERE YOU LIVE, NOT WHERE YOUR FILES LIVE

Because SAS Enterprise Guide is a client application that can work with local or remote SAS servers, you can reference and save SAS programs that reside on remote servers. In fact, you can store entire projects on remote SAS servers, either for your own convenience or to collaborate with others in your organization. In this way, SAS Enterprise Guide acts like a built-in file transfer service to work with remote files as if they are stored on your local machine.

When you open or save a file in SAS Enterprise Guide, you are first presented with a choice between **Local Computer** or **SAS Servers**. If you select **SAS Servers**, you can navigate the file system of any remote SAS server as if it were your own machine and select a file or a destination as appropriate.

YOU'RE THE PROGRAMMER—SO PROGRAM!

We've established that you can use SAS Enterprise Guide to create and manage your own SAS programs. However, this application offers approximately 70 tasks that can generate code for a variety of purposes, including data manipulation, reporting, creating graphs, and performing statistical analyses. It would be a shame to let all that code-generating power sit idle!

FILLING THE GAP

Have you ever looked at the code that's generated by an automatic process and thought, "This code would be so much more useful if it only had an option for *<your desired option here>?*" Usually, you end up using the code as a starting point, copying it to another process, and then filling in the gaps. The problem with that approach is that the resulting code is permanently separated from the process that generated it, and you cannot go back and maintain the code from that original process.

SAS Tasks in SAS Enterprise Guide were designed for ultimate flexibility, and they offer a feature called Insert User Code, which allows you to insert your own code at predefined locations in the generated task code. These changes are then maintained in the task item in the project so that you always have access to the point-and-click interface that helped you generate the code.

Here's how you get to the Insert User Code feature. In the dialog window for the task that you're working with, click the **Preview Code** check box so that the code window is shown. This window shows you the code that's been generated based on selected options. In the **Code Preview** window, click the **Insert Code...** button. This opens the **User Code** window (see Figure 2). The custom code "insertion points" are highlighted. The location and frequency of the insertion points vary by task, but there are always enough insertion points to offer plenty of flexibility for you to add your favorite PROC statements or options.

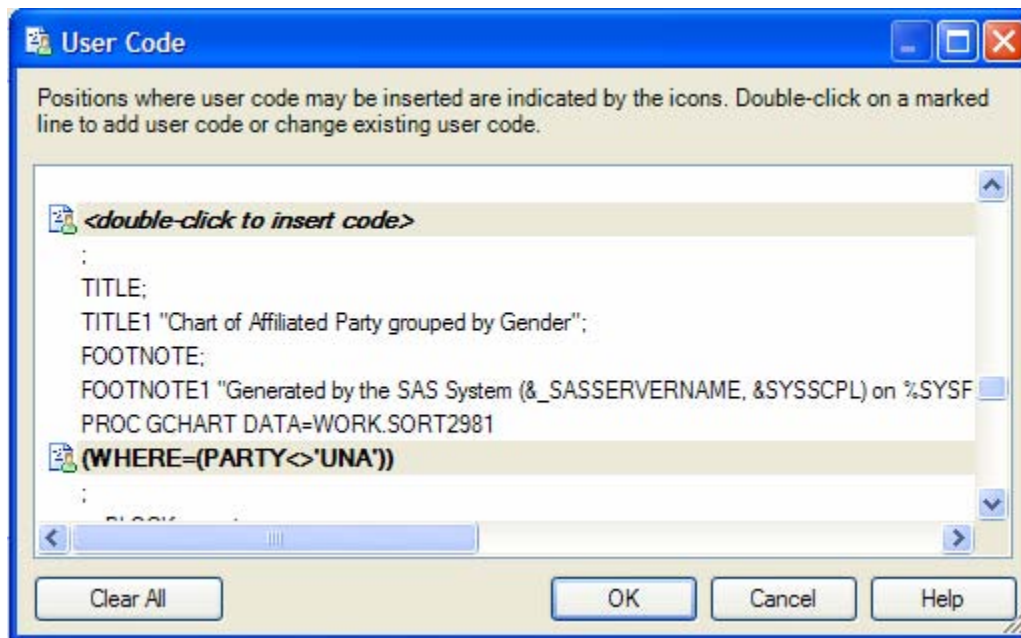


Figure 2. Custom Code Insertion Points in User Code Window in SAS Enterprise Guide

PUT RESULTS WHERE YOU WANT THEM

SAS Enterprise Guide automatically generates ODS statements that help redirect output back to the project. By default, you'll get HTML output for reports—but you can select to get the traditional text listing as well as PDF or RTF output, or any combination of these. For those output types that support appearance styles, SAS Enterprise Guide offers a simple method to select and apply the style that you want. This makes it easy to work on a process and develop results iteratively, all within the SAS Enterprise Guide environment.

After you have things working just the way you want them, you'll probably want a way to share those results with others. SAS Enterprise Guide offers a few point-and-click methods for assembling results and exporting or publishing them from within the application. If all you want to do is share your ODS results as-is, add the ODS statements to your SAS programs to write the results to whatever location you choose. What could be easier?

In the past, you had to choose whether you wanted your results to be returned to your project or to be written automatically to an external file. Now you can have it both ways. Here's how to do it:

- For code items (SAS programs), add your ODS statements in the code as you normally would. Specify ODS targets (HTML, PDF, RTF, etc.) along with the appropriate style and pagination options. When specifying output filenames, specify the file path in terms of the SAS server that the code will run on. For example, if the SAS server is running under a UNIX operating environment, use a UNIX style file path that is valid for that server.
- For built-in tasks, you can use the technique that was described in the preceding section to insert your own code into the task. Add the desired ODS statement at the first insertion point in the task (see Figure 3).

If needed, you can also add ODS CLOSE statements for your ODS destinations. Because SAS Enterprise Guide submits the catch-all ODS ALL_CLOSE statement at the end of each SAS job, this step is optional.

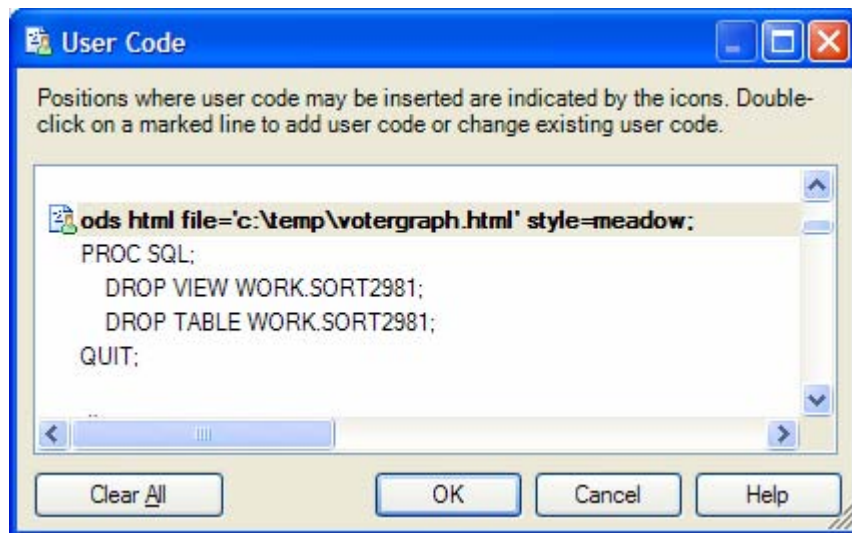


Figure 3. ODS Statement Added in User Code Window in SAS Enterprise Guide

KNOW THE TRICKS – IMPRESS YOUR FRIENDS!

This section of the paper describes a few little-known features in SAS Enterprise Guide. You can file these away for discussion at the next SAS programmer party and give the impression that you are really in-the-know!

First, some background. The SAS System traditionally provides some automatic macro variables that supply environmental information—for example, &SYSUSERID for the current user, &SYSDATE and &SYSTIME for the SAS session date and time, etc. Because SAS Enterprise Guide is a client application that connects to a SAS server, there is an additional environment full of information that you might find interesting to surface in your SAS programs. Table 1 lists additional macros defined in your SAS Enterprise Guide session.

&_CLIENTUSERID	User ID of the client (Windows) user
&_CLIENTUSERNAME	Full user name of the Windows client user
&_CLIENTMACHINE	Client-machine node name (your Windows PC)
&_CLIENTVERSION	Application version, including build number
&_CLIENTAPP	Name of the client application (for example, "SAS Enterprise Guide")
&_SASSERVERNAME	Name of the logical server (for example, "SASMain")
&_SASHOSTNAME	Server node name (IP address, DNS name, or DCOM name)
&_CLIENTTASKLABEL	Label for the current task. This is the text label that is displayed in the project tree and the process flow diagram.

Table 1. Additional SAS Enterprise Guide Macros

Try it! Here's a SAS program that puts these macros to use:

```
data _null_;
  file print ods;
  length macroname $ 50 macrovalue $ 50;
  label macroname='Macro' macrovalue='Value';
  macroname = 'Current application';
  macrovalue = "&_CLIENTAPP";
  put macroname macrovalue;
  macroname = 'Application version';
```

```

macrovalue = "&_CLIENTVERSION";
put macroname macrovalue;
macroname = 'Windows user ID';
macrovalue = "&_CLIENTUSERID";
put macroname macrovalue;
macroname = 'Windows user name';
macrovalue = "&_CLIENTUSERNAME";
put macroname macrovalue;
macroname = 'SAS server name (logical name)';
macrovalue = "&_SASSERVERNAME";
put macroname macrovalue;
macroname = 'SAS server machine address';
macrovalue = "&_SASHOSTNAME";
put macroname macrovalue;
macroname = 'Task label (that produced this output)';
macrovalue = "&_CLIENTTASKLABEL";
put macroname macrovalue;

run;

```

STORED PROCESSES—PLAY AND SHARE WITH OTHERS

Stored processes are a core feature of the SAS®9 platform. They are a great way for you to take your tremendous SAS programming knowledge and deploy it to the masses, whether they are using SAS Enterprise Guide, the SAS Add-In for Microsoft Office, or a variety of Web-based clients.

A stored process is a SAS program that has optional parameters and metadata to run the program. SAS Enterprise Guide enables you to concentrate on creating the core piece of a stored process, that is, a SAS program, and provides a wizard to help you to deploy it.

Creating and using stored processes requires that you have a SAS Metadata Server set up, and either a stored process server or a workspace server that is designated to run stored processes. Here are the basic steps for creating a stored process in SAS Enterprise Guide:

Start. Select the SAS program or the task in the process flow that you want to use as the core of the stored process. Right-click the program or task and select **Create Stored Process...** from the resulting menu.

Select a location. The wizard that displays is very straightforward. It enables you to provide a name and a description, in addition to information in other metadata fields. As you continue with the wizard, you can select the server that will be designated to run the process and the location where the resulting SAS program file will be stored.

Set up data libraries (optional). If your task or SAS program relies on SAS libraries that might not be defined on the stored process server, you are given the opportunity to specify LIBNAME statements to define them.

Define parameters (optional). You can specify parameters to be used in your stored process. Parameters are macro variables that you can use in your SAS program. The end user of the stored process will be prompted for values for these macro variables, based on the information and constraints that you create when you define the macro variables. The wizard lets you define parameters “from scratch” and provides options to discover potential parameters that are based on the SAS program or the task, as shown in the following menu (Figure 4).

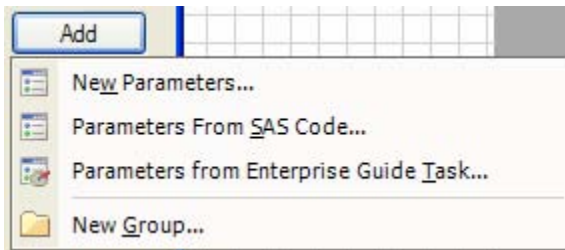


Figure 4. Parameters Menu

Selecting **Parameters From SAS Code...** scans the SAS program for the use of macro variables and uses the results to seed the parameter definitions. Selecting **Parameters from Enterprise Guide Task...** uses the knowledge that's built into the task to surface complete parameter definitions, as appropriate.

Select your output options. The final page of the wizard allows you to select an output medium. Usually, you would select Transient package for output, but you can also direct stored processes to store results in a permanent location or to stream output back to the client. For Web-based clients, you should choose "Streaming", which streams the output back for display in the client Web browser.

Once created, you can use SAS Enterprise Guide to run your stored processes. You can also run existing stored processes, regardless of whether they were created in SAS Enterprise Guide. And finally, you can use SAS Enterprise Guide to modify the content of existing stored processes, which includes editing metadata (such as parameters) as well as editing the SAS program code.

Are you a SAS/IntrNet programmer and not using stored processes? SAS Enterprise Guide offers a parallel wizard to deploy a SAS program as a SAS/IntrNet Application (including a Web page to prompt for parameters!).

LATHER, RINSE, REPEAT...USING AUTOMATION

One of the great strengths of SAS and SAS Enterprise Guide is that you can design a series of processes to be repeated multiple times, and have the results change accordingly with changes in the data and other variables. After you have designed your process in a SAS Enterprise Guide project, you can run that process simply by opening the project and selecting to run the flow.

Is this not convenient enough for you? Well, you can also schedule your process to run automatically. This is similar to the traditional SAS batch process except that, while you schedule it to run on your Windows computer, your SAS Enterprise Guide project can use local and remote SAS servers. You can also customize an automation script to automatically save and publish results outside the project.

SAS Enterprise Guide can automatically generate a basic automation script to run the entire project or a single process flow and integrate with the Windows scheduler. The script uses the Visual Basic Script language (VBScript), a well-documented scripting facility that is available on any Windows computer.

The SAS Enterprise Guide scripting model can also help you to automate otherwise tedious tasks. For example, you can use automation to "harvest" information from your SAS Enterprise Guide projects. The following is an example script (VBScript file) that reports the contents of a project:

```
' Globals
Option Explicit
Dim oEGApp ' SAS Enterprise Guide Application Object

' Run main routine
Main()

' handle arguments, call report routine
' Recommend using CSCRIPT.exe to run, as output will be directed to console.
Sub Main()
    If (WScript.Arguments.length <> 1) Then
        WScript.Echo("Usage: [WScript | CScript] ProjectReport.vbs
<project_directory>")
        WScript.Quit()
    End If

    ReportOnDir(WScript.Arguments(0))
End Sub

' *****
' This sub finds all of the SAS Enterprise Guide projects in the specified folder
' *****
Sub ReportOnDir(sDir)
    Dim oFSO ' FileSystemObject
    Dim oEGFolder ' Folder of SAS Enterprise Guide Projects
```

```

Dim oEGFiles          ' Potential SAS Enterprise Guide Project Files Collection
Dim oEGFile           ' Potential SAS Enterprise Guide file object

' Get a collection of files in the directory
Set oFSO = CreateObject("Scripting.FileSystemObject")
Set oEGFolder = oFSO.GetFolder(sDir)
If oEGFolder Is Nothing Then
    WScript.Echo("Bad input directory: " & sDir)
    Exit Sub
End If
Set oEGFiles = oEGFolder.Files

' Create an instance of SAS Enterprise Guide 3.0
Set oEGApp = WScript.CreateObject("SASEGObjectModel.Application")

For Each oEGFile In oEGFiles
    If UCase(Right(oEGFile.Path, 4)) = ".EGP" Then
        ' This appears to be a project file
        WScript.Echo("Project file: " & oEGFile.Path)
        ReportOnProject(oEGFile.Path)
    End If
Next

' Close SAS Enterprise Guide
oEGApp.Quit

WScript.Echo("Report generated successfully!")
End Sub

' *****
' This sub opens a SAS Enterprise Guide project and reports on its contents
' *****
Sub ReportOnProject(sProject)
    Dim oEGProject ' SAS Enterprise Guide Project object

    ' Application.Open takes two parameters
    ' The second one represents the password, if any. Use "" if none.
    On Error Resume Next
    Set oEGProject = oEGApp.Open(sProject, "")
    ' Handle error if project couldn't be open
    If Err.Number <> 0 Then
        WScript.Echo(vbTab & "Project " & sProject & " could not be opened -- " &
            Err.Description & vbCrLf )
        oEGProject = Nothing
        Exit Sub
    End If

    ' placeholders for project items and collections
    Dim objItem
    Dim pi
    Dim dc
    Dim tc
    Dim results
    Dim resultItem
    Dim task

    ' Project summary
    Set pi = oEGProject.ProjectItems
    Set dc = oEGProject.DataCollection
    WScript.Echo vbTab & oEGProject.Name & " - Project Items: " & pi.Count & " Data
references: " & dc.Count

    ' Flat list of all TOP-LEVEL project items (data, code, notes, etc)
    WScript.Echo vbTab & "Project items: "
    For Each objItem in pi
        WScript.Echo vbTab & " " & objItem.Name
    Next

```

```

' Find all of the data items in the project and go through them one by one
WScript.Echo vbTab & "Data items: "
For Each objItem in dc
    WScript.Echo vbTab & "  " & objItem.Name
    Set tc = objItem.Tasks

    ' go through each task hanging off of this data
    For Each task in tc
        WScript.Echo vbTab & "    " & task.Name
        Set results = task.Results

        ' get a reference to each ODS result item
        For Each resultItem in results
            ' could also do a resultItem.SaveAs here
            WScript.Echo vbTab & "      " & resultItem.Name
        Next
    Next
Next
Next
'---- Close the project
oEGProject.Close

WScript.Echo "-----" & vbCrLf
End Sub

```

NEED MORE FUNCTION? ADD ON!

It's rare that a business application can meet 100% of an end-user's needs. As a result, users often have to compromise by resorting to other applications to fill the gap or work without a desired feature. SAS Enterprise Guide, with its extendibility via custom add-in tasks, eliminates the need for this type of compromise.

A custom add-in task is a task that **you** create and add to the list of the 70-plus tasks that are already included with SAS Enterprise Guide. You can use a custom task to satisfy a need specific to your industry or organization, or to simplify a SAS programming task that non-programmers in your organization need to perform as part of their jobs.

The add-in mechanism in SAS Enterprise Guide enables you to create a custom task that functions just like the tasks that ship from SAS. You build your add-in by using Microsoft .NET technologies and develop code in Visual Basic .NET, C#, or any other .NET language that you prefer. The Add-In API (documented at <http://support.sas.com/eguide>) accesses all the same application services that are provided to the built-in tasks, including: access to SAS servers, access to data, and the ability to submit code and retrieve logs.

The applications of the add-in approach are limited only by your imagination. Here are some examples:

- Create a user interface for your SAS macro library that enables the dialogs that you create to pass user input as arguments to the macros.
- Use .NET technologies to access data that is external to SAS and bring it to your project—perhaps using Web services as a data source.
- Fill in a missing function in SAS Enterprise Guide. For example, there is a specific feature in the REPORT procedure that you would like to make available to end-users, but you don't want them to have to understand SAS code to use the new feature.

The SAS Customer Support Center Web site contains complete documentation and several examples of using the add-in feature in SAS Enterprise Guide.

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, while providing services to help you manage the non-programming tasks of organizing your work, performing SAS tasks outside of your normal routine, and sharing your results and work with the rest of your organization. By creating stored processes and custom add-in tasks, the SAS programmer

can “do the heavy lifting” while others in the organization benefit from the experience in a simplified, yet powerful, environment.

REFERENCES

Hemedinger, Chris and McDaniel, Stephen (2005), “The New World of SAS—Programming with SAS Enterprise Guide,” *Proceedings of the Thirtieth Annual SAS Users Group International Conference*

RECOMMENDED READING

Looking for a community of other SAS Enterprise Guide users? See <http://www.segus.org>, the home of the SAS Enterprise Guide User’s Group.

See <http://support.sas.com/eguide> for:

- documentation and examples of SAS custom tasks using SAS Enterprise Guide and the SAS Add-In for Microsoft Office
- documentation and examples for automating SAS Enterprise Guide using VBScript and OLE automation.

CONTACT INFORMATION

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

Chris Hemedinger
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Work Phone: 919-677-8000
Email: Chris.Hemedinger@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.