Paper 157-2009

A "SAS® Programmer's" Guide to the SAS® Enterprise Guide®

Marje Fecht, Prowerk Consulting LLC, Cape Coral, FL
Rupinder Dhillon, Dhillon Consulting Inc., Toronto, ON, Canada

ABSTRACT

You have been programming in SAS for a while….. You have been told "we are moving to Enterprise Guide" and removing Display Manager". You say …… *I can program everything just fine myself, thank – you!* OR What *am I supposed to do with all these windows?*

This paper demonstrates how SAS programmers can use SAS Enterprise Guide as their primary interface to the SAS system while maintaining the flexibility of writing their own customized code. We will look at

- how to navigate the views and menus
- how SAS Enterprise Guide can be used as your primary SAS Editor
- how you can leverage the more complex built-in capabilities available in SAS Enterprise Guide to further enhance the information you deliver
- some tips and tricks to get the most out of SAS Enterprise Guide.

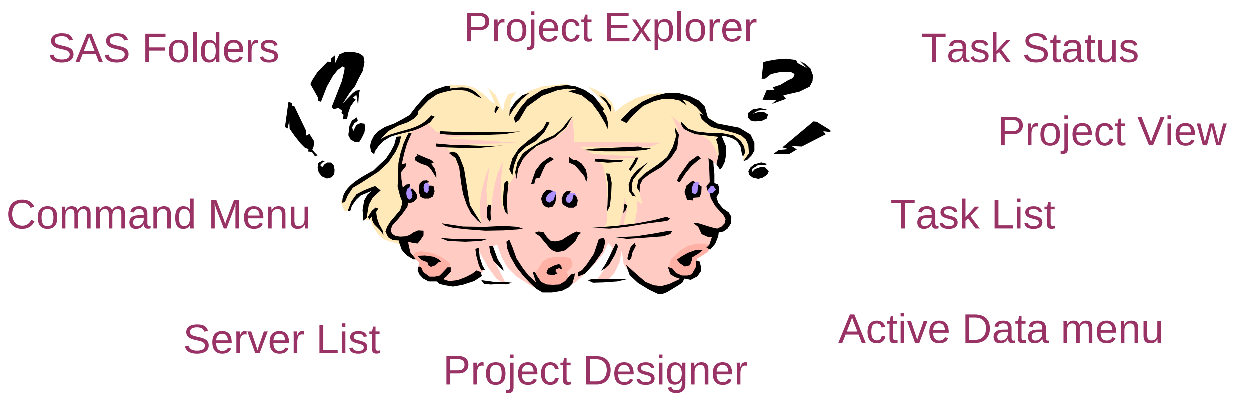
Enterprise Guide version 4.1 will be used.

INTRODUCTION

This presentation is designed as a Hands-On Workshop, where the participants follow along and step through the usage of SAS Enterprise Guide. This Proceedings paper follows a similar format.

I KNOW HOW TO PROGRAM IN SAS – HOW DO I NAVIGATE SAS ENTERPRISE GUIDE?

With all the windows and views and menus available in Enterprise Guide, it is hard to know where to start! For many "long –time" SAS programmers, the initial view can send them running right back to their programming environment. But, trust us, the benefits of Enterprise Guide are worth the time it takes to get used to all the windows.



GETTING STARTED

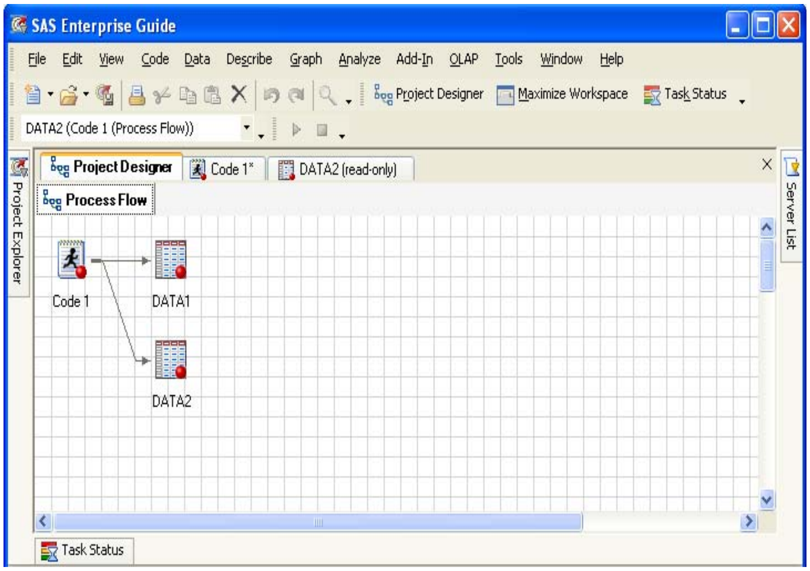
The first time you open Enterprise Guide, you will see the Welcome to Enterprise Guide Screen. This screen gives you the option of launching the Enterprise Guide tutorial or starting a new project. If you do not want this to be your default screen when you open Enterprise Guide, you can check the box labeled 'Do not show this screen again, ' after which you will be taken to the default view with the MAIN MENU displayed at the top of your screen.

The main menu allows you to access the various tasks and views that are available to you. Although you may also access the tasks and views through shortcut buttons, the main menu organizes them into logical categories, making them easy to find.

SAS Enterprise Guide
File Edit View Code Data Describe Graph Analyze Add-In OLAP Tools Window Help

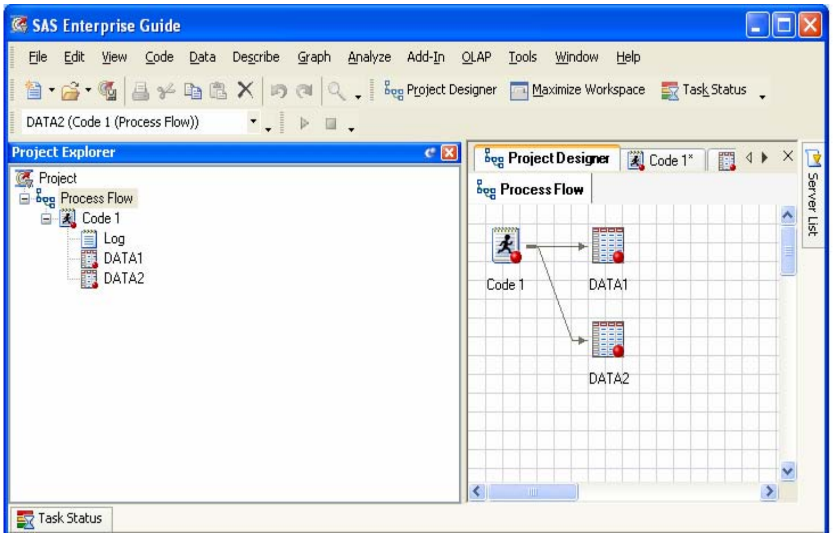
As SAS Programmers, you all know that the tasks you are expected to perform are seldom simple. More often than not, there are a series of SAS programs that you run in a sequence, with different types of output directed to different locations, along with all the intermediate and final SAS datasets that are generated and need to be saved.

Enterprise Guide provides a mechanism to organize your work by allowing you to group sequences of tasks, data items and results into **Projects**. All of the code and tasks that you run within an Enterprise Guide session are part of your active Project and are each displayed as separate **nodes**.


SAS Enterprise Guide
File Edit View Code Data Describe Graph Analyze Add-In OLAP Tools Window Help
Project Designer Maximize Workspace Task Status
DATA2 (Code 1 (Process Flow))
Project Explorer
Project Designer Code 1* DATA2 (read-only)
Process Flow
Code 1
DATA1
DATA2
Server List
Task Status

These nodes can then be linked and organized in a logical sequence to form a **Process Flow**. You can save your Process Flow as a part of a Project; then anytime the project is opened the same sequence of tasks can be easily executed. This helps you and your organization distribute and share information easily. The **Project Designer view** is used to display all of the **task**, **data** and **code nodes** that make up your Process Flow.

There are a number of ways that you can access your Project Designer view. From the main menu, select **View ➔ Project Designer**. You can also select **F4** or click the Project Designer shortcut button. Notice that the **Project Designer** view also displays the connections between your work components. Again, this allows you to see and visually inspect the organization of tasks, code and data associated with a Project.


SAS Enterprise Guide
File Edit View Code Data Describe Graph Analyze Add-In OLAP Tools Window Help
Project Designer Maximize Workspace Task Status
DATA2 (Code 1 (Process Flow))
Project Explorer
Project
Process Flow
Code 1
Log
DATA1
DATA2
Project Designer Code 1*
Process Flow
Code 1
DATA1
DATA2
Server List
Task Status

You may prefer the **Project Explorer** view, which provides a more traditional “Windows Explorer” Hierarchical view of your projects. Select **View ➔ Project Explorer** from the main menu to open the Project Explorer view. Note that the Project Explorer view does not display the flow (connection) within your projects.

BUT, I AM A PROGRAMMER – HOW DO I RUN MY FAVORITE PROGRAM IN ENTERPRISE GUIDE?

Enterprise Guide enables you to easily run existing SAS programs, using similar steps to what you would follow using the SAS Display Manager windows. Let's run a simple program and review the log and results using Enterprise Guide. Throughout this workshop, we will be using the Shoes dataset from the SASHELP library.

To run an existing program using Enterprise Guide (See Appendix – Program_1 for example program),

- from the **Main Menu**, select **File ➔ Open ➔ Code ➔ Local Computer**
- type the location and name of your program into the **File Name** box, and then click **Open**
- for this example, open Program_1
- the actual code will open in a new window tab
- notice that a code node, Program_1 is added to your Process Flow workspace
- run the program by selecting **Run On SASMain** from the POP MENU in the code node, OR by clicking ▷

In this instance, we are running our SAS program on a server instance of SAS (SASMain). The first time that you submit SAS code to run on the server, you'll be prompted to enter your credentials (your user name and your password). If you have SAS on your local machine and you are not connecting to a server, your Run option will be **Run on Local**.

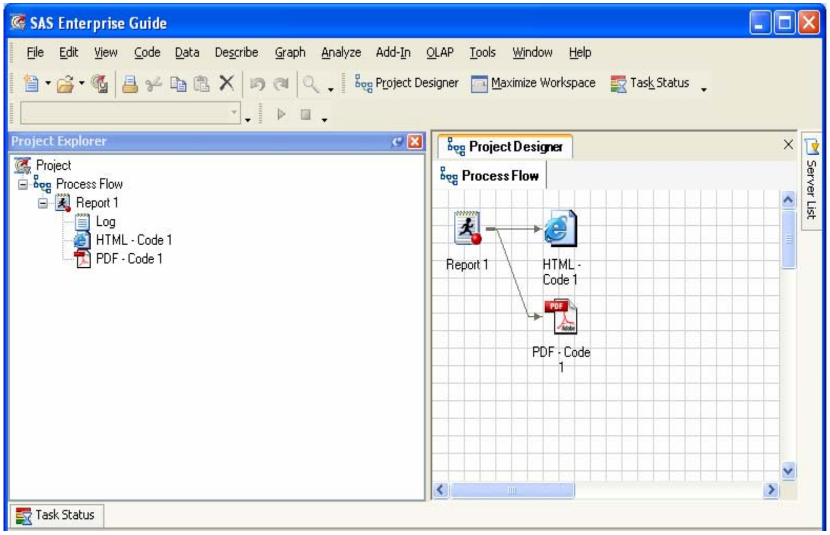
You can change your connection settings to switch from a local SAS instance to a server instance or to switch from one server to another in the Administration panel of your Options screen (**Tools ➔ Options ➔ Administration).** The remainder of the examples in this paper assume that we are running on a Server named SASMain.

VIEWING LOGS AND RESULTS

To review the log messages, double click the **Log** icon in your **Project Explorer** view. The SAS log will open up in a new window tab.

Helpful Hint: If you would like your SAS Log to open automatically when you have an error, check the **Display SAS Log when Errors Occur** option in your **Results** settings.

Since our example program produces an Excel Spreadsheet report as output, you would use your normal process for opening and reviewing the results (such as Windows Explorer). If the results were ***standard SAS output*** you would double click on the generated results node in either your Project Designer view or your Project Explorer view and the results will open in a new tab window within Enterprise Guide. Leveraging the functionality of ODS, you can prompt Enterprise Guide to produce different types of output. You can easily produce PDF results along with your standard SAS output by simply changing your Results Options settings. Each type of generated result will display as its own result node in your Project Designer or Project Explorer view.



Helpful Hint: To change the type of Results generated by a specific code node or task, **right click on the node** and select **properties**. Under the **Results**, click the checkbox beside the type of output you want to generate.

CUSTOMIZING THE WINDOWING ENVIRONMENT

Everyone has different preferences about how Enterprise Guide windows should work, and Enterprise Guide provides the flexibility for you to create a work space that works best for you. For example, Enterprise Guide windows can be

- docked and undocked
- resized to make best use of the real estate
- hidden so that you only display what you need and use
- opened as new horizontal or vertical tabs
- easily reopened using the main menu or shortcut buttons.

Helpful Hint: Unhappy with your window changes? Click the **Reset your Docking Windows** button in your Options window.

THAT WAS PAINLESS, BUT I NEED TO BE ABLE TO *CHANGE* CODE?

Suppose that you need to enhance existing code, rather than just run it. Navigate to the Program_1 code node and double click to open it in a new window tab if it isn't already open; or activate the tab if the code is already open. Once the code is open, you can make changes; you will find the editor commands are identical to the SAS Editor. Then simply rerun the code and review your results. You can save your changes from the main menu by selecting **File ➔ Save.** Remember, each time you save a code node, you are actually saving a **.sas** program file.

If you have selected 'Open the LOG when errors occur' in your Options, then your log will now automatically appear if there are issues.

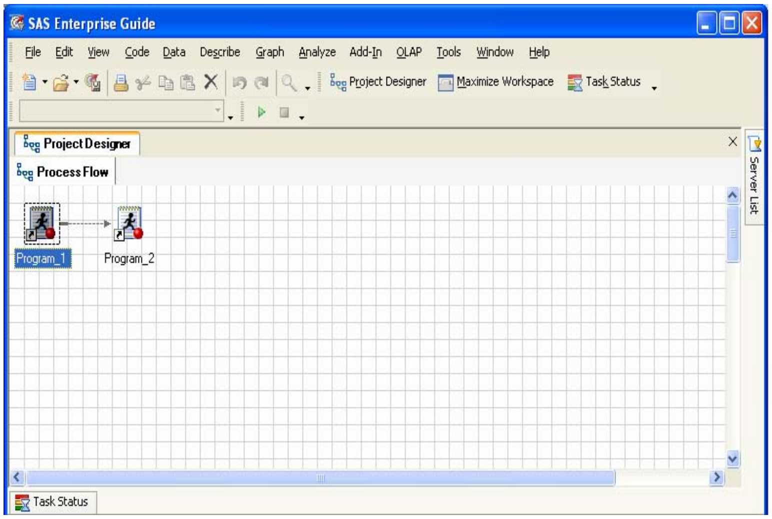
Helpful Hint: If you would like to run just a section of the code, highlight and submit just like you do in the SAS Display Manager Editors.

SPLITTING YOUR PROGRAM INTO LOGICAL STEPS

During program development and testing, do you often work with just one set of steps at a time, confirming results before you move on? If you do, that same logic can easily be applied to your code in Enterprise Guide by defining your program as a series of code nodes. Code nodes can be linked together in the Process flow window, thus providing a clear picture of the steps and intermediate results.

To create a second code node that links to the first code node,

- open an existing program by selecting **File ➔ Open ➔ Code ➔ Local Computer**, from the main menu
- supply the location and name of your program in the **File Name** box, and then click **Open**. For this example, use Program_2 (see the appendix for code stored in Program_2).
- notice that a Program_2 code node is added to your Process Flow workspace.
- link the new code node to the Program_1 code node



- right click on the Program_1 code node in your Project Designer window
- select **Link Program_1 to . . .**
- select **Program_2** and click **OK.**
- You now see the link between the two nodes and have the option of running either a single code node or running the branch that starts with Program_1.

To run both code nodes, right click on your Program_1 code node and select **Run Branch from Program_1**. Once the code in Program_1 has run, Enterprise Guide will automatically run Program_2, generating and displaying all intermediate results along the way.

Helpful Hint: To remove a link between tasks or Code nodes, right click anywhere on the joining arrow and select **Delete**. This will only delete the linkage, not the code nodes.

You'll notice that a separate log node is generated for each code node. By turning on **Project Logging**, you can easily tell Enterprise Guide that you'd like a single SAS log to be generated for all of the tasks and code nodes in your Project. This single Project Log will be created in addition to the individual logs created for each task or code node.

Helpful Hint: If Project Logging is turned on, the log represents a running log of the entire project. To turn on the Project Logging, right click on the **Project** node in the **Project Explorer View** and select **Properties**. Under the **Project Log** tab, click the checkbox labeled **Maintain Project Log**.

WOW – THAT'S EASY – ARE GRAPHICS EASY TOO?

You have now submitted and modified existing code in Enterprise Guide, so let's work on a task from scratch. A picture is worth a thousand words, so it is time to see how Enterprise Guide does with graphics. Using the same sales data used in the first example program (Program_1), you will produce a vertical bar chart of Sales by Region.

- Select **Graph ➔ Bar Chart** from the main menu. This opens the Bar Chart Wizard.
- In the **Bar Chart** screen, select **Simple Vertical Bar**
- In the **Task Roles** screen
  - Drag **Region** to **Column to Chart**
  - Drag **Sales** to **Sum of**.
- In the **Appearance ➔ Options Screen**, change the **Shape** to **Cylinder** from drop down list
- In the **Titles** screen
  - Click on **Graph**
  - Uncheck **Use Default Text**. Enter your own title for your graph.
- To see what code was generated, click on **Preview Code**.
- Close the **Code Preview Window**
- Click on **Run** to create the Bar Chart
- The bar chart opens automatically in a new Tab in your Enterprise Guide session.

LEVERAGING CODE THAT ENTERPRISE GUIDE BUILDS

Behind the scenes, Enterprise Guide created "correct" code! Like most SAS programmers, you would probably like to re-use the code elsewhere.

To access and then save the generated code,

- In the **Project Window**, double click on the **Bar Chart** task node
- Highlight the **GOPTIONS** statements and the **PROC GCHART** statements
- Right click anywhere on the highlighted code, and select **Copy**
- **Close** the **Code Preview** Window
- **Close** the **Bar Chart Task** Window.

To insert the copied code into your program, double click on the appropriate code node and **paste** the code in the desired location. To save the expanded code node, select **File ➔ Save Code As ➔ Local Computer**, and provide the name Program_3.

The expanded code can now be run in the usual fashion. Note that you could also have created a separate code node from this copied code, as discussed earlier. You can make any additional updates and customizations to code that you've just pasted, such as changing the appearance of the Legend or adding a Where clause to subset the data being graphed.

Helpful Hint: The **Task Status** Window shows you the status of the code or task you are running. If you are running large SAS programs, the Task Status will tell you which task or Procedure is currently running. You can open your Task status view from your Main menu. Select **`View`** ➔ **`Task Status`**.

Each of the built-in tasks will generate SAS code as you step through the Task Panes. This serves as a great way to learn how to code new SAS Procedures since the syntax is available to view and edit.

At this point, save the Project so it is available for another day.

- Select **File** ➔ **Save Project As** ➔ **Local Computer**
- Navigate to the file directory where you would like to save your project.
- Enter a name for your Project, and Click **ok**.

The next time you open your saved project; you will see the code nodes and links that you defined in this session. Note that any temporary work datasets that you may have created in your session will not be saved with your project. You can easily recreate the work datasets by re-running the project the next time you open it.

CUSTOMIZING THE GRAPH FORMAT

By default, graphs are generated as interactive ActiveX graphs, which are *cool* but can use up a lot of system resources. You may prefer or require a more static format such as HTML or JPEG. You can easily change the default in the Graph Results settings of Options.

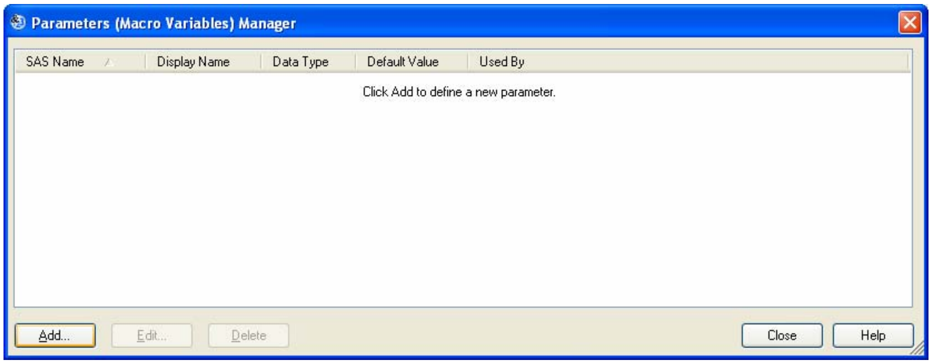
Select **Tools** ➔ **Options** ➔ **Results** ➔ **Graphs** and select a different format from the drop down list.

Helpful Hint: When working with several tasks and code nodes in a single project, you can export all of the generated SAS code as a single .sas file by selecting **File** ➔ **Export** ➔ **Export All Code..** Enterprise Guide will scan through your project to find all executable code and save it to a single .sas file. You can select export all the code or select only parts of your project.

HOW CAN I ADD SOME FLEXIBILITY TO MY PROJECTS?

Project Parameters are user prompts that are passed as macro variables to your SAS code. By adding parameters to your project, you give the user some flexibility in customizing the results. The user can enter values or select from a list of values that you provide. In our example, we would like the user to be able to create this same report for one or more regions existing in our data.

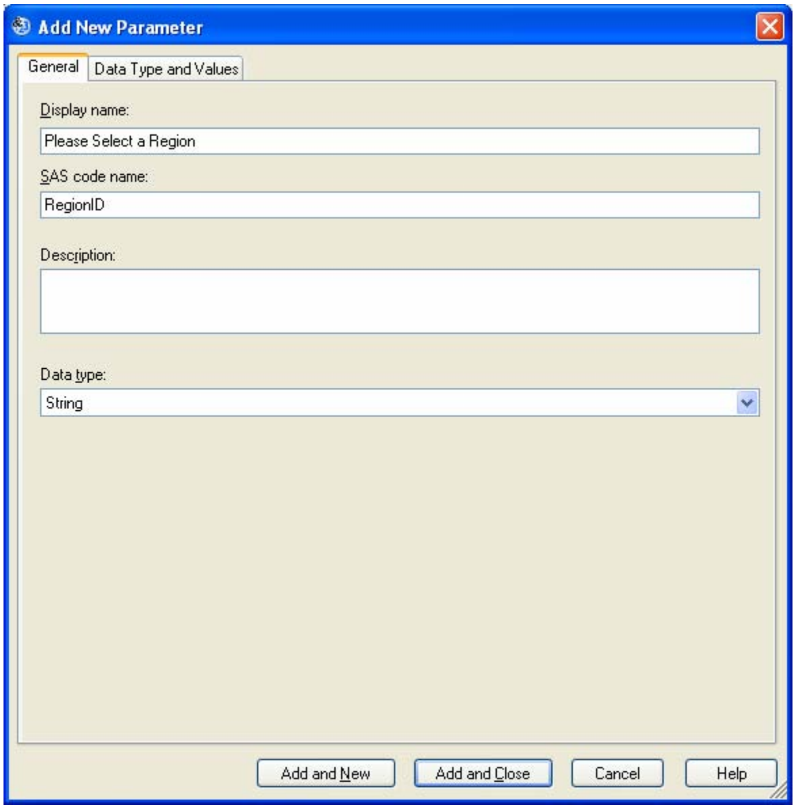
- Select **Tools** → **Parameters (Macro Variable) Manager**
- Click **Add** to add a new parameter.



Parameters (Macro Variables) Manager

SAS Name	Display Name	Data Type	Default Value	Used By

Click Add to define a new parameter.



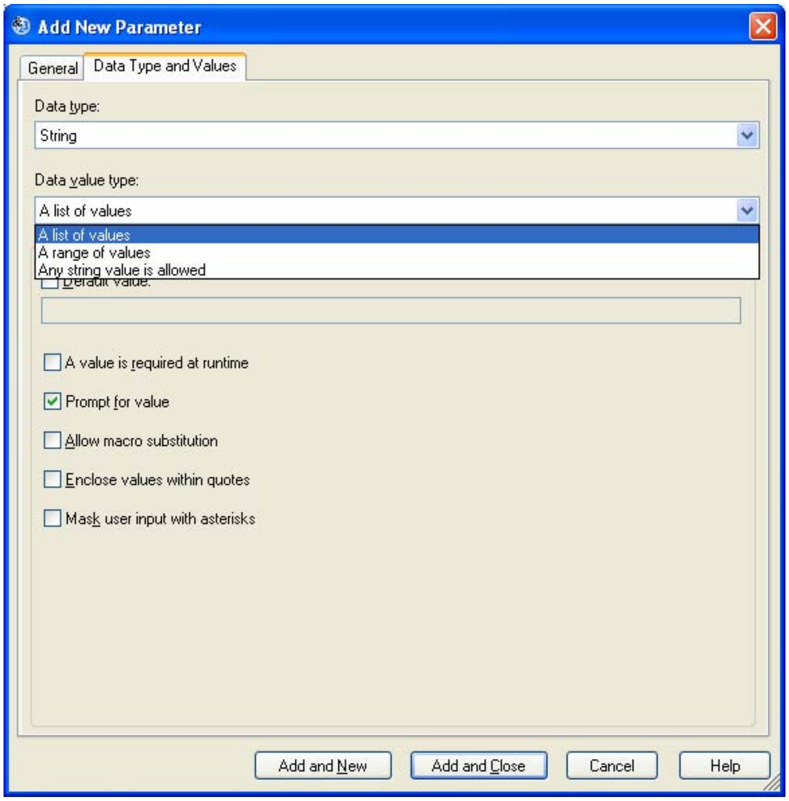
Display name: Please Select a Region

SAS code name: RegionID

Description:

Data type: String

- To provide descriptive text for the user-prompt, enter **“Please select a Region”** under the **General** tab in the Display Name section.
- To specify the macro variable that will receive selected values, enter **“RegionID”** in the SAS Code name section.
- Select the **Data Type and Values** tab.



Data type: String

Data value type: A list of values

□ Default value:

□ A value is required at runtime

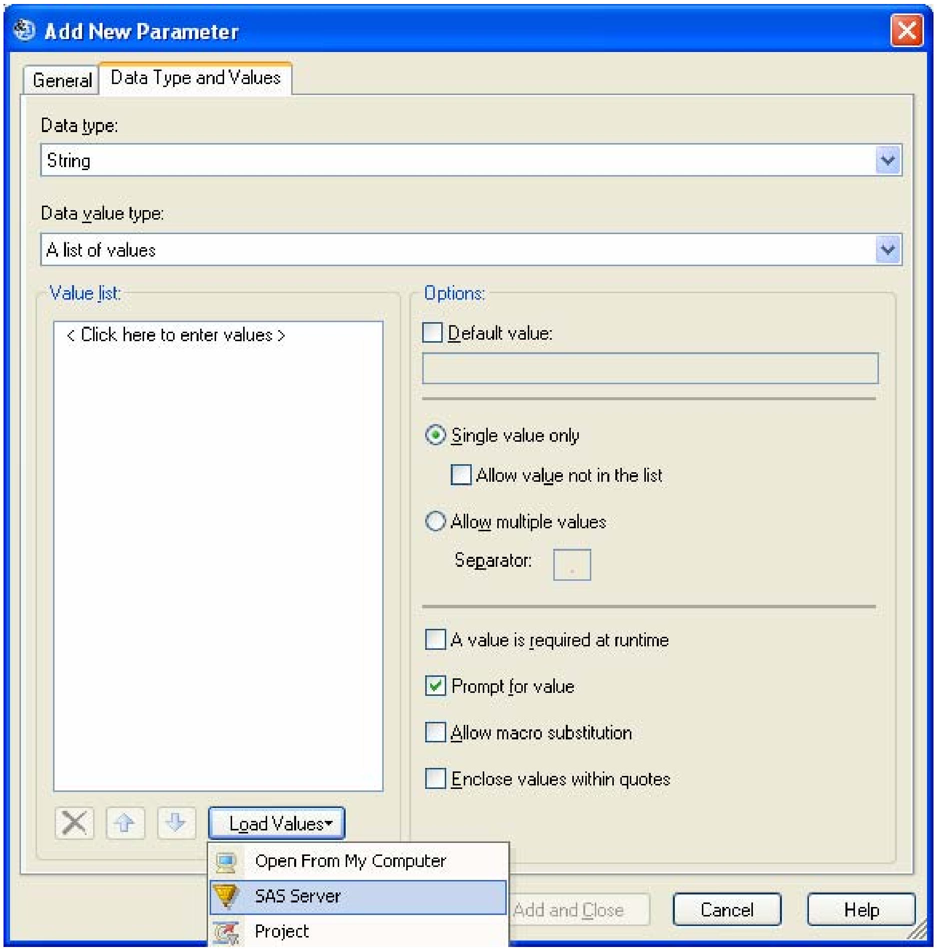
☑ Prompt for value

□ Allow macro substitution

□ Enclose values within quotes

□ Mask user input with asterisks

- In the Data Type Value dropdown, select **A List of Values**



Data type: String

Data value type: A list of values

Value list:

< Click here to enter values >

Options:

□ Default value:

● Single value only

□ Allow value not in the list

○ Allow multiple values

Separator:

□ A value is required at runtime

☑ Prompt for value

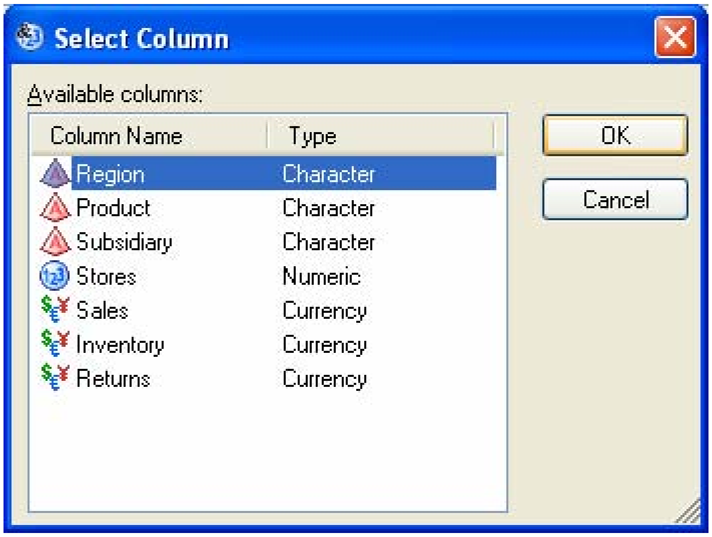
□ Allow macro substitution

□ Enclose values within quotes

- Select **Load Values**
- Specify location of data as **SAS Server**

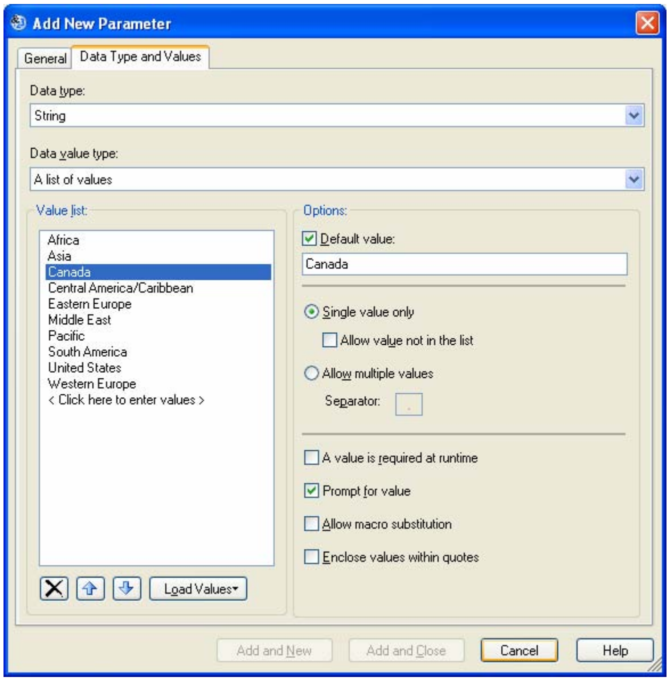
Maintain the current settings of ***Single Value Only*** as selected and ***Enclose Values within Quotes*** unselected.

- Navigate to the SHOES dataset: **SASMAIN → Libraries → SASHELP → SHOES**
- In the ‘Select Column’ window, **Select Region, Click OK**



Available columns:

Column Name	Type
Region	Character
Product	Character
Subsidiary	Character
Stores	Numeric
Sales	Currency
Inventory	Currency
Returns	Currency



Add New Parameter

General | Data Type and Values

Data type: String

Data value type: A list of values

Value list:
- Africa
- Asia
- Canada
- Central America/Caribbean
- Eastern Europe
- Middle East
- Pacific
- South America
- United States
- Western Europe
- < Click here to enter values >

Options:

☑ Default value: Canada

● Single value only

☐ Allow value not in the list

○ Allow multiple values

Separator: ,

☐ A value is required at runtime

☑ Prompt for value

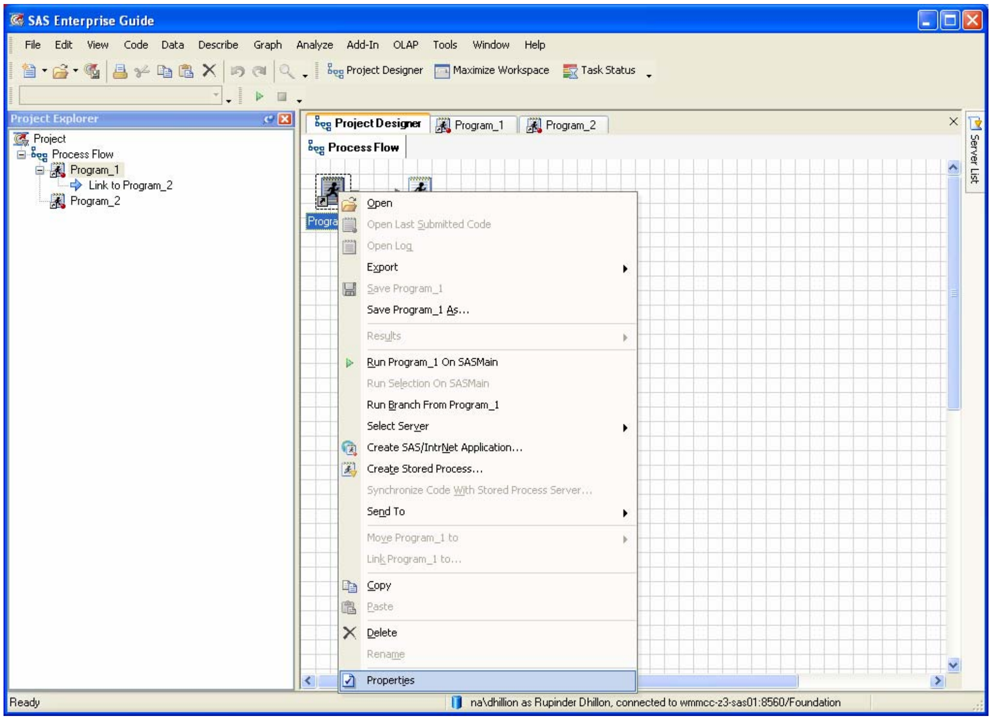
☐ Allow macro substitution

☐ Enclose values within quotes

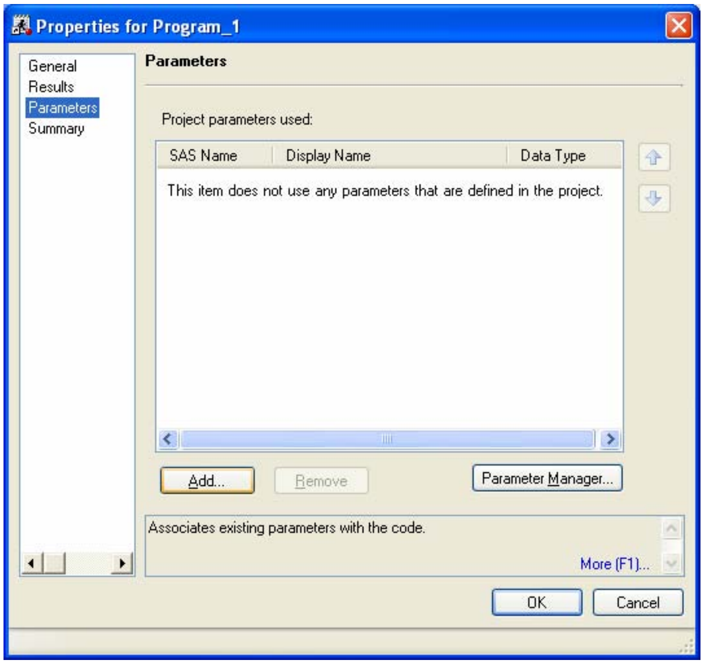
- To designate the default value that will display, Check **“Default Value”, double click on “Canada”**
- Click **Add and Close**
- Click **Close**.

Now that parameters have been defined, we need to modify the code to accept parameters. From the **Project Designer** window:

- **Right click** on the Program_1 Code Node and select **Properties**



From the Properties window, select **Parameters** and Click **Add.**



Parameters

Project parameters used:

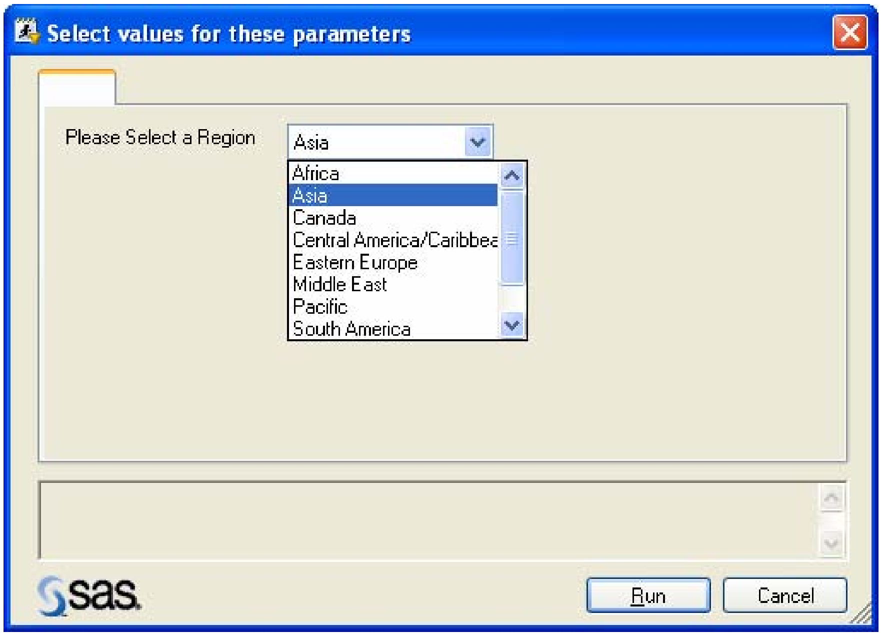
SAS Name	Display Name	Data Type
This item does not use any parameters that are defined in the project.		

Associates existing parameters with the code.

- Select **'RegionID' and click 'OK'**
- **Click 'OK'** again to close the Code node Properties window

Now, modify the program so that the parameters control the data selected.

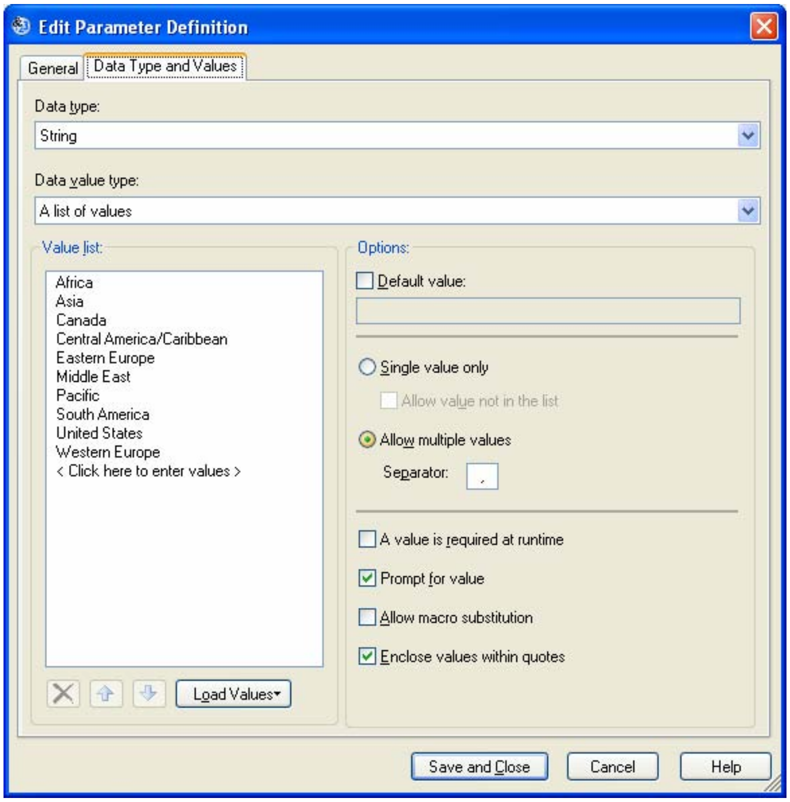
- **Double Click** your Program_1 code node
- **Uncomment** "**where Region = "&regionID**" from your Ex_Shoes_Data datastep.
- **Save** the changes to your Program_1 code node.
- **Right click** on the Program_1 code node, **Select Run Branch From Program_1**.
- When prompted to select a Region, **select 'Asia'** from the drop down list. **Click Run.**



Please Select a Region: Asia

Instead of forcing just a single selection, what if you wanted to provide a LIST of values that the user can select from?

- Select **Tools** and then **Parameters (Macro Variable) Manager**
- Select the RegionID parameter and click **EDIT**
- On the Data Type and Values tab, click the radio button beside **Allow multiple Values** and also select **Enclose Values within Quotes**
- Click Save and Close.



Edit Parameter Definition

Data type: String

Data value type: A list of values

Value list:

Africa
Asia
Canada
Central America/Caribbean
Eastern Europe
Middle East
Pacific
South America
United States
Western Europe
< Click here to enter values >

Options:

□ Default value:

○ Single value only

□ Allow value not in the list

● Allow multiple values

Separator: ,

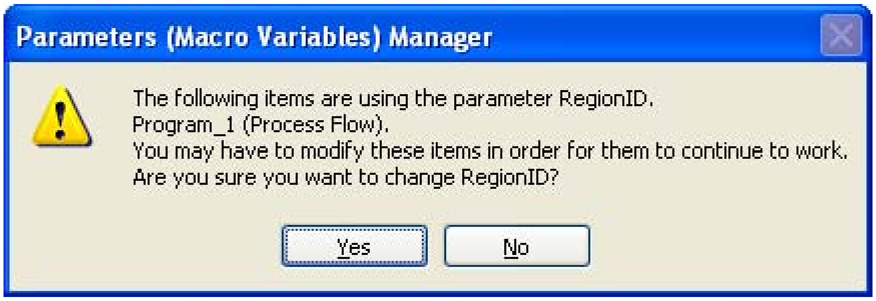
□ A value is required at runtime

☑ Prompt for value

□ Allow macro substitution

☑ Enclose values within quotes

- You'll receive the following warning message reminding you that you're about to modify a parameter that is being used by the Program_1 code node in your project.



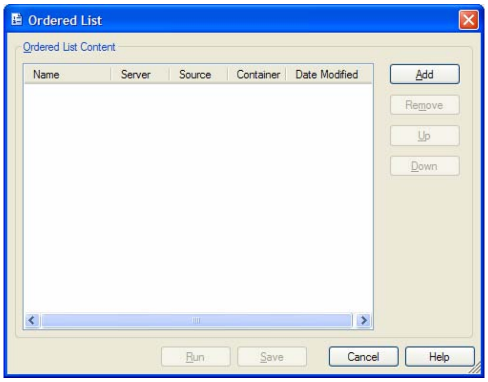
Parameters (Macro Variables) Manager

The following items are using the parameter RegionID.
Program_1 (Process Flow).
You may have to modify these items in order for them to continue to work.
Are you sure you want to change RegionID?

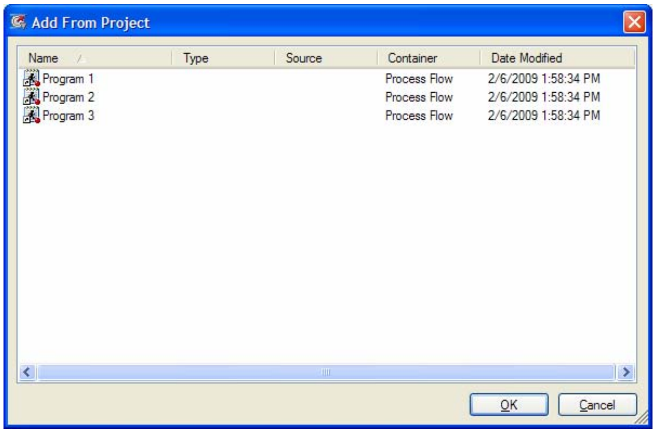
- Click **Yes** and **Close**
- **Double Click** on the Program_1 node to open the code
- **Change** 'where region = "&regionID" **to** "where Region IN (&regionID);
- **Save** the changes to your Program_1 code node.
- **Right click** on the Program_1 code node, **Select Run Branch From Program_1**.
- Holding down the **CTRL** key, **select Africa, Canada and Eastern Europe**. **Click Run.**

I'M NOT WORKING WITH PROJECTS – I JUST NEED TO RUN THESE JOBS

Thinking about our SAS jobs within the context of projects is a new concept for a lot of us. We are used to working with individual SAS jobs that are run separately and are not always related or dependent on each other. What if we have 3 SAS jobs that we'd like to run, one after the other? Or perhaps we have some processes that we'd like to submit at the end of the day so we can have the results ready the next morning. In this case, we don't really want to build a project; we want to be able to submit a 'list' of SAS jobs or tasks that we want Enterprise Guide to run. This is where we would 'Create an Ordered List'. The Ordered List task in Enterprise Guide allows you to specify a list and sequence of tasks and jobs that you want Enterprise Guide to run. Let's use the code from the previous example to create an Ordered List.



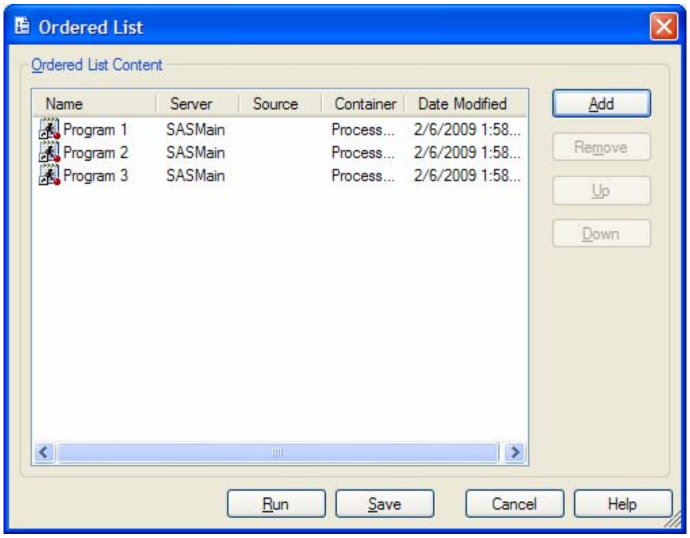
- First you need to make sure that the code or tasks that you want to run have been added to your project window.
- Select **Tools** and then **Create Ordered List**
- **Click Add**



Name	Type	Source	Container	Date Modified
Program 1			Process Flow	2/6/2009 1:58:34 PM
Program 2			Process Flow	2/6/2009 1:58:34 PM
Program 3			Process Flow	2/6/2009 1:58:34 PM

- Enterprise Guide will scan through your project for all executable SAS tasks and list them for you
- Select the tasks that you would like to add to your Ordered List. You can select more than one by holding the SHIFT or CNTL keys. Click OK.

- The tasks have now been added to the Ordered List (above)



Name	Server	Source	Container	Date Modified
Program 1	SASMain		Process...	2/6/2009 1:58...
Program 2	SASMain		Process...	2/6/2009 1:58...
Program 3	SASMain		Process...	2/6/2009 1:58...

- We can change the order that these tasks are executed using the **Up** and **Down** buttons.
- We now have the option of **running** this ordered list or **Saving** it to run later
- **Click Run** to submit these tasks.

You'll notice that your Task Status window shows you the task that is currently running as well as the tasks that are queued to run next.

OK – ENTERPRISE GUIDE REALLY WILL HANDLE MY NEEDS BUT HOW CAN I OFFLOAD SOME OF MY WORK (OR DO MORE WORK IN LESS TIME)?

For most SAS programmers, it is a rare occurrence to write a program that is never used again. More typically, you write programs that either run in production, or that are rerun with different input (parameters) on a regular basis. Furthermore, YOU are not always the person that runs the job after it is written. Someone else may be responsible for providing input parameters (changes) and running your code when results are needed. With this in mind, we explore how you can easily set up your code so that others can access and get the results they need. You will do this by creating a Stored Process.

A **Stored Process** is a SAS program that is typically located on a central server (ie: available to multiple users across multiple clients) so that users can run the SAS Code but can't change the SAS code. Designed properly, the Stored Process enables user input to maximize the applicability of the code. For example, you

- can embed business rules and logic so that everyone is creating consistent results
- have change control around any coding updates
- can make your code available to more people
- can provide the users with input parameter choices to take care of minor tweaks in the requirements.

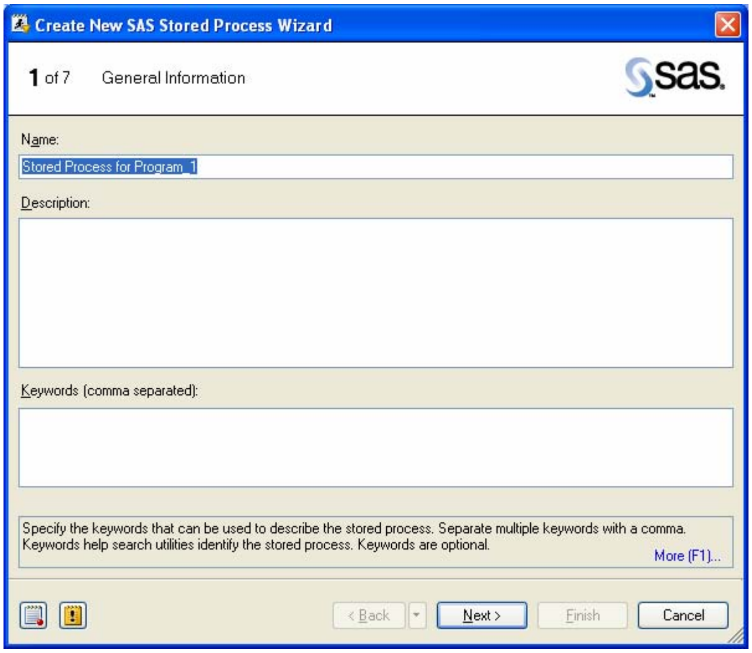
CREATING A STORED PROCESS

Earlier we said that a Stored Process is a SAS Program. That is only half the story; the Stored Process also consists of Metadata that describes the mechanics. So, before we get started, you must have a Metadata Server configured and running in your work environment. Assuming that is complete, create a simple stored process in your project:

- First, lets remove the Project Parameter associated with our Program_1 code. Right click on your **Program_1 code node**
- Select **Properties** → **Parameters.**
- Highlight **Region ID** and click **Remove.** Click **Ok.**
- Back to the Project Designer, **double click** on your Program_1 code node and **comment out** "where Region IN (&regionID);" → `*where Region IN (&regionID);` **Save** the changes to Program_1.

Let's start by creating a Metadata location for our Stored Process:

- Open the SAS folders View by selecting **View** → **SAS Folders**
- Right Click on **BIP Tree** and Select **New Folder**
- Right Click on the **new folder**, and select **Rename**
- Name the new folder, **WSEG1**
- Back to the Project Designer, **right click** on your Program_1 code
- Select **Create Stored Process**, which will launch the **Create New SAS Stored Process Wizard**



Create New SAS Stored Process Wizard

1 of 7 General Information

Name: Stored Process for Program_1

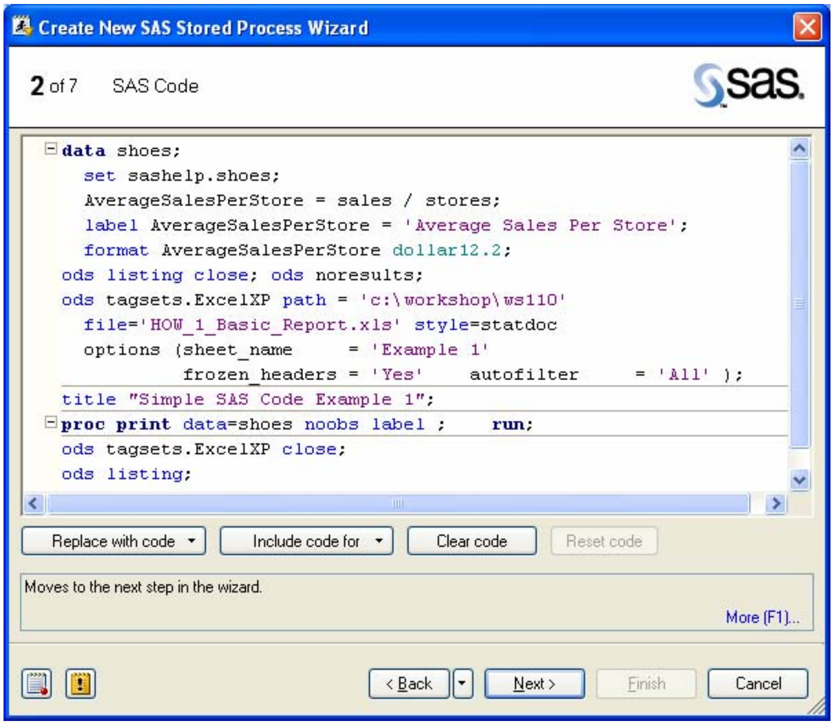
Description:

Keywords (comma separated):

Specify the keywords that can be used to describe the stored process. Separate multiple keywords with a comma. Keywords help search utilities identify the stored process. Keywords are optional.

- In the **General Information** screen, enter a name for your Stored Process and click **Next**

- The second screen will show you the **SAS Code** that will be used in your Stored Process (in this example, Program_1 code). No changes are required so click **Next**



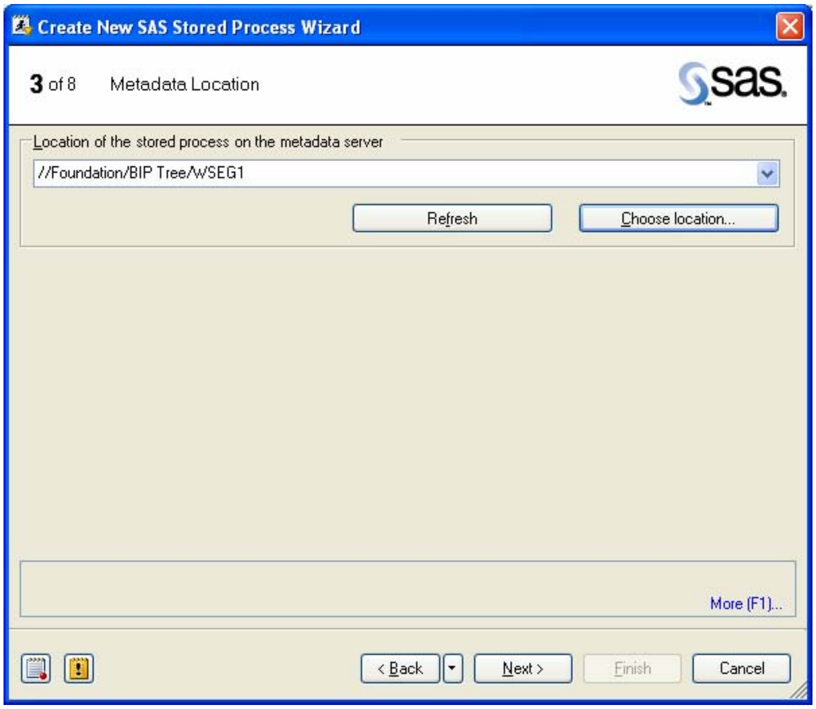
Create New SAS Stored Process Wizard

2 of 7 SAS Code

```
data shoes;
  set sashelp.shoes;
  AverageSalesPerStore = sales / stores;
  label AverageSalesPerStore = 'Average Sales Per Store';
  format AverageSalesPerStore dollar12.2;
ods listing close; ods noresults;
ods tagsets.ExcelXP path = 'c:\workshop\ws110'
  file='HOW_1_Basic_Report.xls' style=statdoc
  options (sheet_name     = 'Example 1'
           frozen_headers = 'Yes'     autofilter     = 'All' );
title "Simple SAS Code Example 1";
proc print data=shoes noobs label ;    run;
ods tagsets.ExcelXP close;
ods listing;
```

Moves to the next step in the wizard.

- in the third screen, you need to specify the **Metadata Location** where the metadata for your Stored Process will be saved, then click **Next**. Your administrator can help you determine the location for your Stored Processes. In this example, store the Stored Process metadata in the Samples directory. **Click** on the **Choose Location** and navigate to **//Foundation/BIP Tree/WSEG1.** Click **OK** → Click **Next.**



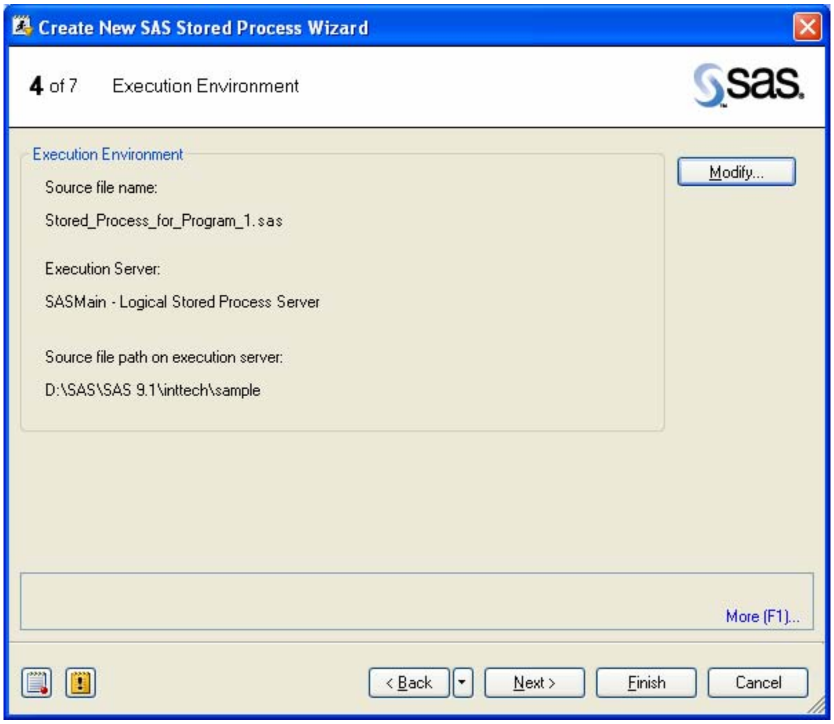
Create New SAS Stored Process Wizard

3 of 8 Metadata Location

Location of the stored process on the metadata server

//Foundation/BIP Tree/WSEG1

- Screen 4 defines the **execution environment**. Here you can modify the source file (where your SAS code will be saved), the Execution server (Workspace server or Stored Process server). Select Modify. For your Execution Server, select SASMAIN – Logical Workspace Server from the drop down list.
    - Click **ADD** and **select OK**. Click **OK** again and **Save**. Lastly, click **Next**.


Create New SAS Stored Process Wizard
4 of 7 Execution Environment
sas
Modify...
More (F1)...
< Back
Next >
Finish
Cancel

Execution Environment

Source file name:

Stored_Process_for_Program_1.sas

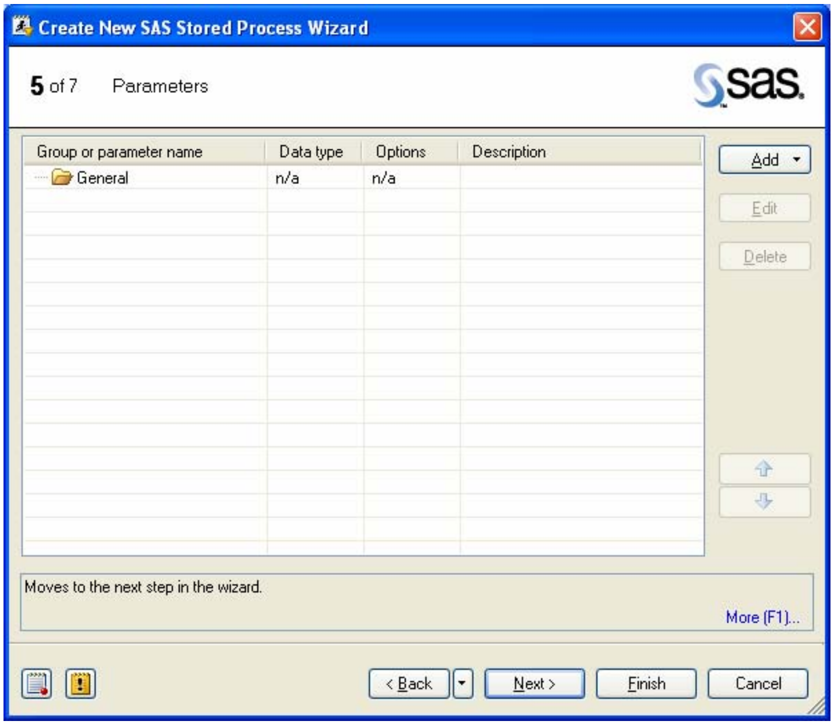
Execution Server:

SASMain - Logical Stored Process Server

Source file path on execution server:

D:\SAS\SAS 9.1\inttech\sample

- Screen 5 enables input of **Parameters**. We will not be using Parameters in our example, click **Next**

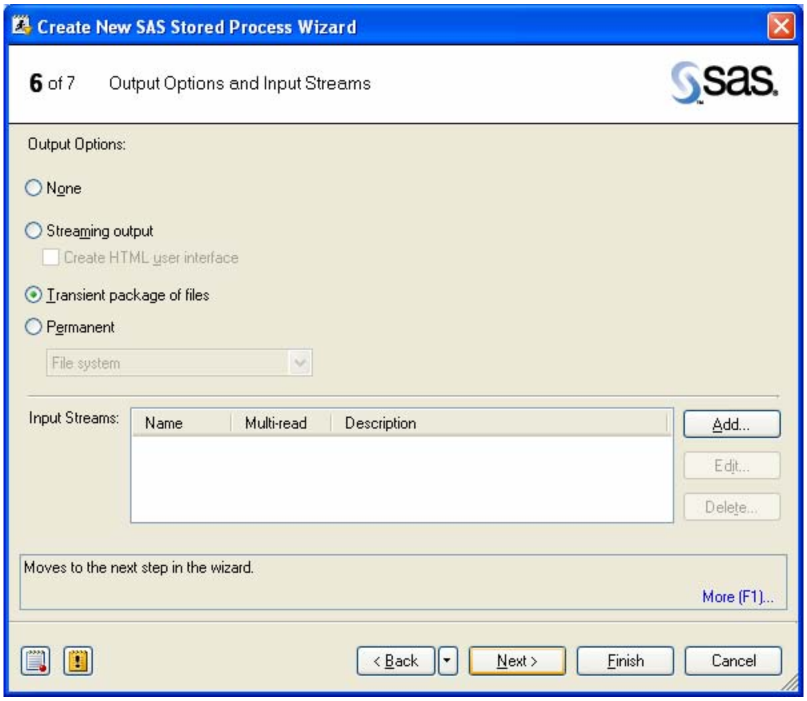


Group or parameter name	Data type	Options	Description
General	n/a	n/a	

Moves to the next step in the wizard.

Note: An additional screen would be included if ***librefs*** needed to be defined.

- Screen 6 defines the available **Output Options**. Select **Transient Package of Files**, and click **Next**


Create New SAS Stored Process Wizard
6 of 7
Output Options and Input Streams
sas
Add...
Edit...
Delete...
More (F1)...
< Back
Next >
Finish
Cancel

Output Options:

○ None

○ Streaming output

□ Create HTML user interface

● Transient package of files

○ Permanent

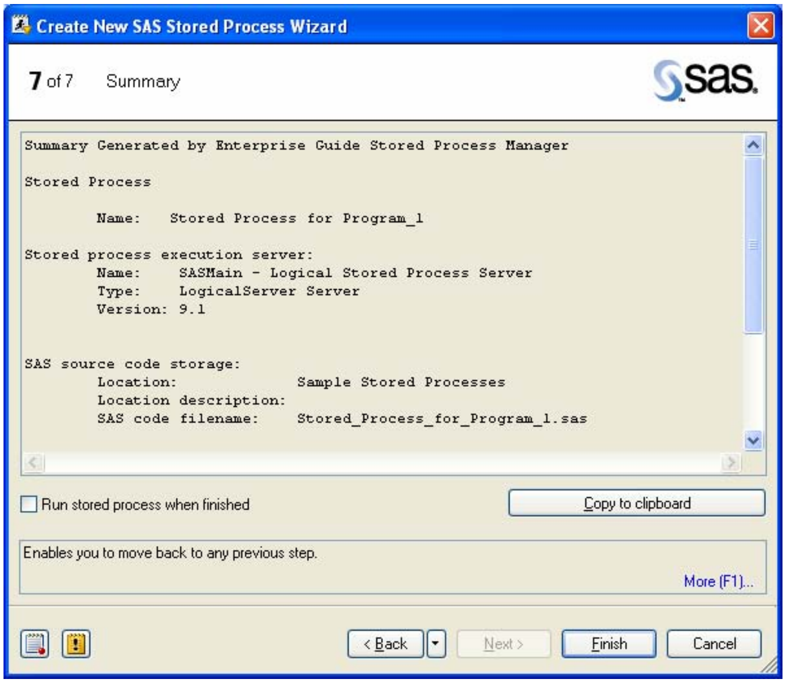
File system

Input Streams:

Name	Multi-read	Description

Moves to the next step in the wizard.

- Finally, screen 7 is a **Summary** of your selections for this stored process. Click **Finish**.

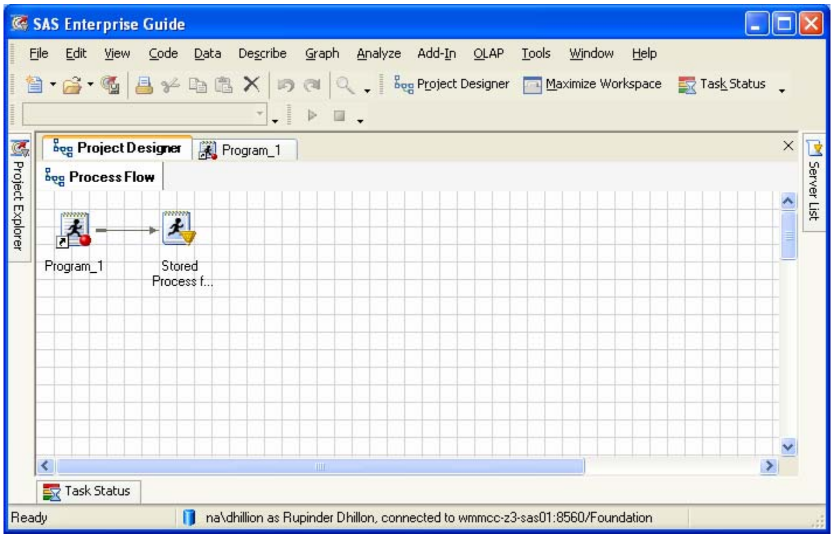


```
Summary Generated by Enterprise Guide Stored Process Manager

Stored Process

        Name:   Stored Process for Program_1

Stored process execution server:
        Name:    SASMain - Logical Stored Process Server
        Type:    LogicalServer Server
        Version: 9.1


SAS source code storage:
        Location:             Sample Stored Processes
        Location description:
        SAS code filename:    Stored_Process_for_Program_1.sas
```

□ Run stored process when finished

Enables you to move back to any previous step.

The stored process is now available in the Project Designer View.



Ready na\dhillion as Rupinder Dhillon, connected to wmmcc-z3-sas01:8560/Foundation

To **run** the Stored Process, right click on the node and select **Run Stored Process Name**. To view the resulting log, right click on the node and select **Open Log**. You may have noticed that we skipped over the parameters screen in this workshop. Parameters are a useful way of providing the user with some flexibility when running a Stored Process. Parameter values that the user enters are passed as macro variables to the SAS code and can be used as such throughout your coding logic. For more information on Parameters, refer to the recommended reading at the end of this paper.

Helpful Hint: To get easy access to your Stored Processes, use the SAS Folders View. Select **View → SAS Folders** from the main menu.

WHAT'S NEXT

Now that you know the basics of using Enterprise Guide, there are lots of bells and whistles and additional features to explore. First, check out the use of parameters with Stored Processes. Then, begin looking at the other many features that Enterprise Guide and the SAS BI Suite offer.

CONCLUSION

Enterprise Guide 4.1 provides a powerful environment and toolset for end-to-end information delivery. As a SAS programmer, you can easily begin taking advantage of Enterprise Guide to provide a flexible reporting and analytics environment to meet your organization's business needs.

RECOMMENDED READING

To gain a more complete understanding of Enterprise Guide 4.1, the authors recommend the following

- Getting Started with SAS Enterprise Guide – free learning tutorial available from http://www.sas.com/apps/elearning/elearning_details.jsp?pubcode=59271
- Little SAS Book for Enterprise Guide 4.1 – By Susan J. Slaughter, Lora D. Delwiche

ACKNOWLEDGMENTS

The authors appreciate the feedback and suggestions provided by Peter Eberhardt.

CONTACT INFORMATION

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

Marje Fecht
Prowerk Consulting
marje.fecht@prowerk.com
www.prowerk.com

Rupinder Dhillon
Dhillon Consulting
Rupinder@dhillonconsulting.com
www.dhillonconsulting.com

APPENDIX – PROGRAMS

PROGRAM_1

```
/*******************************************************************/
/*  HOW - A SAS Programmer's Guide to SAS Enterprise Guide         */
/*  Example Program 1 - Output Excel Spreadsheet w/ Shoes data     */
/*******************************************************************/
data Ex_Shoes_Data;
  set sashelp.shoes;
  *where Region = "&regionID";
  AverageSalesPerStore = sales / stores;
  label AverageSalesPerStore = 'Average Sales Per Store';
  format AverageSalesPerStore dollar12.2;
run;

ods listing close;
ods tagsets.ExcelXP path = 'c:\HOW\fecht'
  file='HOW_1_Basic_Report.xls' style=statdoc
  options (sheet_name     = 'Example 1'
           frozen_headers = 'Yes'    autofilter     = 'All' );

title "Simple SAS Code Example 1";
proc print data=Ex_Shoes_Data noobs label ;
run;

ods tagsets.ExcelXP close;    ods listing;
```

PROGRAM_2

```
/***********************************************************/
/*  HOW - A SAS Programmer's Guide to SAS Enterprise Guide */
/*  Example Program 2 - Demonstrate Code node links        */
/***********************************************************/

ods listing close;
ods pdf file = 'c:\HOW\fecht\HOW_1_Shoes_Report.pdf';

proc report data = Ex_Shoes_Data nowd;

column ("Location" (Region Subsidiary Stores))
       Product
       ("Sales"    (Sales AverageSalesPerStore))
       ;
define region / order "Region";
define Subsidiary / order "Subsidiary";
define Stores / display "Number of Stores";
define Product / display "Product";
define Sales / Sum "Sales";
define AverageSalesPerStore / Sum "Avg Sales per Store";
break after region / summarize;
rbreak after / summarize;
run;

ods pdf close;     ods listing;
```