

## SAS Enterprise Guide® - Why and How the Programmer Should Adapt It Now

Roger D. Muller, Ph.D., First Phase Consulting, Carmel, IN  
Donald L Penix (DJ), Jr., Pinnacle Solutions, Indianapolis, IN

### Abstract

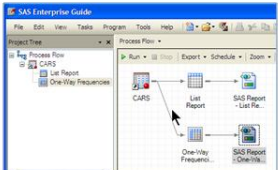
SAS Version 9 for the Microsoft Windows environment is an excellent tool both for developing SAS programs and for running them in production mode. SAS Enterprise Guide (EG) has matured into a developmental tool useful to the advanced SAS programmer. This paper examines the use of SAS EG on the PC in combination with a Unix server (or any other server) which: (1) also has SAS, (2) is the storage depot for production runs of the final code including logs and outputs, and (3) contains the raw data which is being summarized. Workflow will be shown both with and without the use of SAS/Connect which serves to minimize the flow of large amounts of data between machines and takes advantage of the most powerful and appropriate processor for the task at hand. Strong differentiation will be made on the advanced capabilities of exploring SAS data sets in EG with the Query Builder as opposed to the Viewtable facility in PC SAS. Approaches to structuring SAS code that is under development to smoothly transition to Unix will be covered. In addition to the discussion on working in conjunction with Unix, this paper is an introductory beginner's guide to the use of EG in standalone mode.

### Introduction

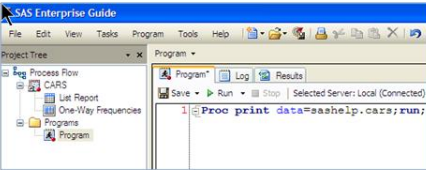
SAS Enterprise Guide (EG) has moved on from its beginnings of being a simple means of developing interfaces for less-experienced (novice) users to fill in blanks and make simple selections to summarize data interactively. Now EG serves as a means of allowing the experienced users to interactively explore their data and to develop and run sophisticated SAS code. How different users might view SAS/EG is shown in the following figure.

**How the SAS/EG Interface Might be Viewed**

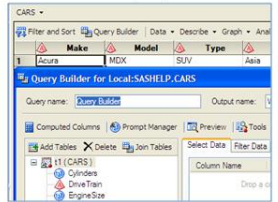
*NonProgramming Professional, Drag  
and Draw Analysis*



*Traditional SAS Programmer, Code  
Log and Listing*



*Both ---Query Tool*



*Today's SAS Programmer*

*All of the Other Rectangles  
Combined  
Plus Advanced Tools for Individual  
and Enterprise Use*

As you can see, today's SAS programmer should be using not only the classical SAS code files, but use EG with extended capabilities to work interactively (the Process Flow), and view the dataset interactively through the Query facility. This paper addresses the use of EG 4.3 on a PC with SAS 9.2 installed. The objectives of this paper are not to give a complete in-depth overview of EG, but rather to show how it might be used in the development of SAS programs that eventually will be run in batch mode on a Unix server. The interactive query tool provided by the Query Builder in EG will be shown as an effective way of exploring SAS data sets and will be used as a superior alternative to Viewtable in interactive PC SAS.

This paper covers the essentials of getting started with EG in a production environment with a very basic "keep it simple" approach covering workflow and steps necessary to understand the product and get productive fast. It is by no means an all inclusive review of EG. It takes a learning approach of "learn only what you need to know to get going, learn the rest later". As an example, the "stored process" capability in SAS EG will be completely ignored.

## 1. Why Do I Want to Use EG?

EG is something else to learn. Why do I want to go to the trouble of learning to use it?

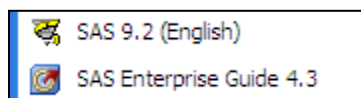
1. Outstanding interactive data exploration to "find out what is in the dataset". This is provided by the Query Builder. Code developed here can be stored between sessions and reused. The capabilities far exceed those of the Viewtable facility in interactive PC SAS which does not store code for future use.
2. Interactively build SQL code with the query builder. There are numerous capabilities here far above and beyond simple data retrieval --- filtering, sorting, sub setting, merging, etc. Any code that EG generates can be saved to a file and used elsewhere.
3. Ease of controlling program execution when testing code. Entire projects, process flows, or nodes can be executed by techniques such as mouse controls, menu controls, etc. This is much easier than commenting out sections of code with SAS commenting facilities prior to submission.
4. The SAS code generated can be exported to other environments with only slight modifications such as libnames being required.
5. Everything is kept in one place -- the project file. This can include not only numerous SAS programs, but supportive documentation such as text files. (See discussion at end of paper for pros and cons on this).
6. Not much new to learn to get started, but you can grow with the capabilities.
7. The editor for editing program code is the most advanced and capable in the SAS system. Syntax help is available when hovering over key words and can be captured and copied into the programming code via Windows shortcuts (highlight, Cntl-C, Cntl-V).

### 2.1 Getting Started, What SAS Products Do I Have/Need?

#### PC SAS and EG:

The default PC SAS installation for Windows usually includes an installation of SAS Enterprise Guide. If this is not available on your machine, check with your SAS installation representative. This paper is based on a machine with Enterprise Guide 4.3 and PC SAS for Windows vs 9.2. These programs are not completely independent of each other in that components are shared. Thus, check with your SAS installation representative or SAS Institute to check compatibility of your software versions.

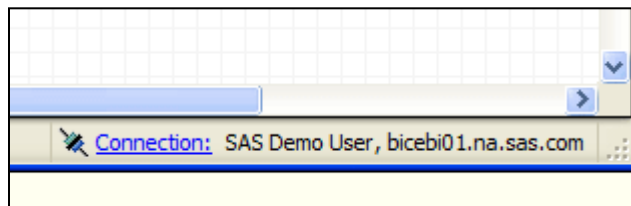
You should find following program icons in Windows.



SAS on Unix, SAS Connect. If you wish to use PC SAS in conjunction with SAS software on Unix, contact your SAS installation representative if you do not already have access to the software on Unix. The SAS Connect software allows for a number of capabilities to use SAS in multiple environments. This paper assumes that you have SAS Connect software both on the PC and Unix. This software allows for

remote submission of SAS programs from the PC to the Unix server for processing

SAS Integrated Technologies (IT) components. This software which has components installed both on the PC and the Unix server allows for almost "finger-tip access" to various servers as they show in your user interface in EG. This paper assumes you do not have the "IT" components installed. Other servers installed that are available to you will be listed in the lower right of the Process Flow screen. By default you do have access to your local SAS PC installation (referred to as the local server).



SAS Add-in for Microsoft Office. This paper does not address the use of these optional tools. In the author's experience, these are very useful for dataset exploration, communications with others, etc. when you have work in process. They are less useful for production batch operations, but do have their place in creating applications for non-systems personnel to execute on a daily basis.

## 2.2 Getting Started -- Training and Documentation

Upon opening EG, you will be presented with a welcome screen containing a link to a training tutorial. It is an excellent place to start and will serve as an excellent refresher when you need it. This tutorial is also available from the Help menu pulldown.

There is not currently an "Enterprise Guide Manual" as such, although there are several third-party books. There are excellent references on the World Wide Web. Here are some links:

<http://support.sas.com/software/products/guide/index.html#documentation>

[http://www.sas.com/technologies/bi/query\\_reporting/guide/index.html](http://www.sas.com/technologies/bi/query_reporting/guide/index.html)

<http://support.sas.com/documentation/onlinedoc/guide/index.html>

<http://www.youtube.com/watch?v=AWBn7IQBMRM>

Due to the highly interactive nature of this software, video tutorials will most likely become more numerous as a means of training and documentation.

## 2.3 EG Key Concepts

The EG "world" consists of developing and running SAS software in a multi-windowed environment, of which the key component and most visible element is the **Process Flow window**. The key "file" of the EG world is the Project File (.egp). Project Files are composed of one or more Process Flows. Process flows are composed of branches. Nodes on the branch may be traditional SAS programming code, SAS procedures and other elements. The following outline and diagram show how these entities are related.

**IMPORTANT: IF YOU EVER GET LOST WHILE NAVIGATING YOUR PROJECT, HIT THE F4 KEY TO RETURN TO YOUR CURRENT PROCESS FLOW TO REGAIN YOUR BEARINGS.**

SAS Project File (all of this is stored as Project\_PharaSug2012\_Simple.egp)

    Process Flow 1 (Currently showing in the Process Flow window)

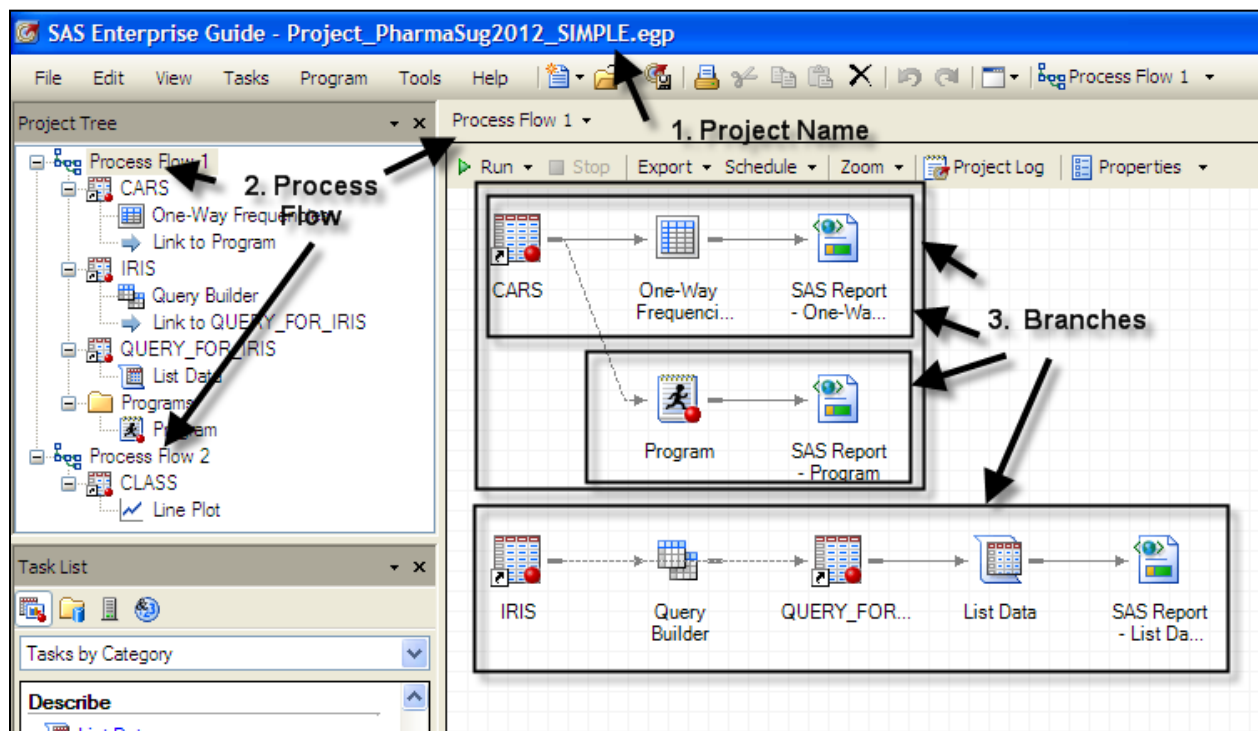
        Branch 1 (Starts with Cars dataset)

        Branch 2 (I am calling this the Branch which starts with Program)

        Branch 3 (Starts with Iris dataset)

    Process Flow 2 (Not currently showing in Process Flow window, look in Project Tree)

        Branch 1 (Not showing, but from Project Tree, we know that it starts with the dataset Class and has a lineplot).



## 2.4 The EG Interface

Upon opening a new project in EG, the user is presented with the following screen. This discussion is limited to key elements/areas on the screen and is by no means all inclusive.

**1. The Welcome Window.** If you have previously created and opened EG projects, they will be shown here. For security reasons, some detail has been hidden in the snapshot. You can open these projects from here, or via the File pulldown menu. The tutorial is an excellent means of not only getting started with EG, but also a means of refreshing your knowledge of advanced techniques that you might not frequently use.

**2. The Process Flow Window.** This is the hub of activity when using EG. Learn how to return to it if you get lost --- the most important shortcut key in EG is F4 which is "Toggle the current Process Flow back-and-forth with the Current Window". The current window most likely is a program or dataset or results window.

**3. The Project Tree Window.** Shows the various Process Flows and their elements in the current project (not populated in the figure as no project is open)

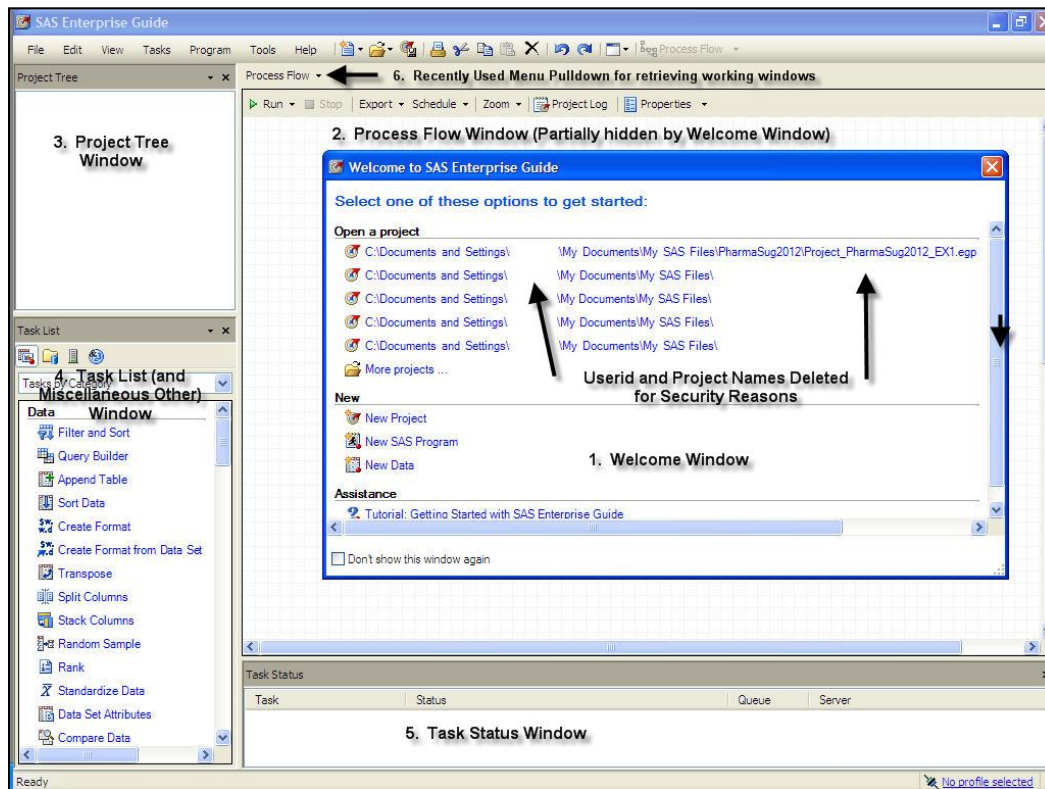
**4. The Task List (and Miscellaneous) Window. (Also Referred to as the Resource Window).**

Provides convenient access for selecting tasks and dragging them to the Process Flow window. Tasks are selected by function or name. Via the icons shown, this window also provides access to other resources such as SAS Folders, a prompt manager (not addressed in this paper) and a listing of SAS servers currently available for processing your requests (again not addressed in this paper). The SAS server also gives you access to SAS datasets on that server and the Local Server contains your current work library.

**5. The Task Status Window (also called the Monitor Window).** This window is your monitor to what is going on when a "Run" is being executed. This window shows the current task in the SAS Process Flow that is being executed. Normally you do not interact with this window and it simply serves as a means of observing what is occurring. **But ---- if your system appears to be hung up while executing a step, right click on the task in this window and select "Stop", or in more severe cases "End SAS Processes"**.

**6. The Menu Pulldown for project elements.** This is the small triangle at the very top left of the

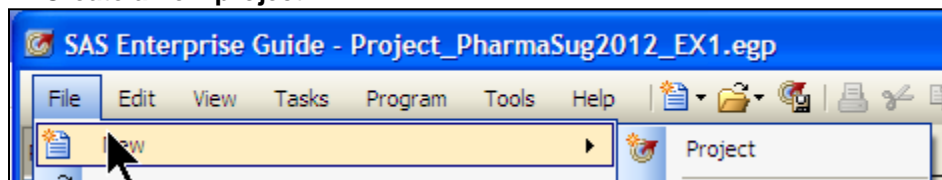
Process Flow and lists Process Flows, programming nodes, etc. Click to access.



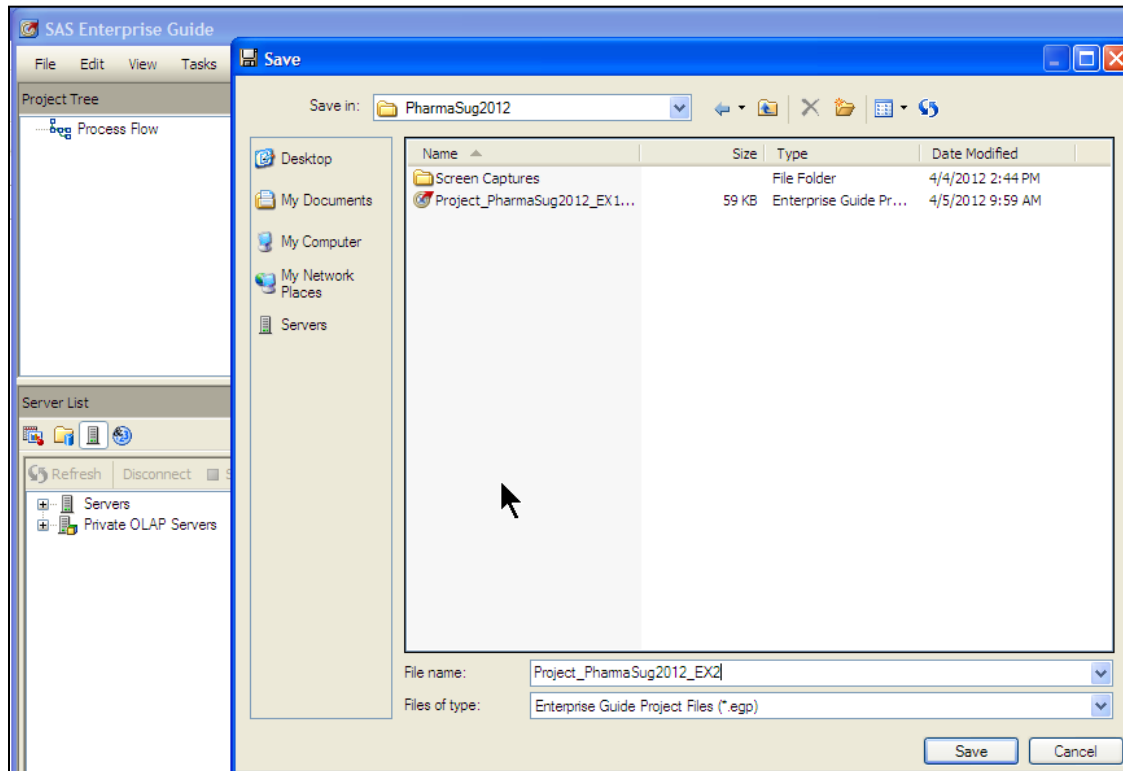
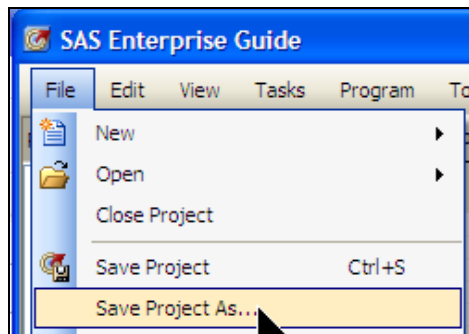
### 3.1 Getting Started, A Simple Project and Process Flow

We will create a very simple SAS program with code that generates a SAS dataset. We will then print it with a task dragged in from the task window.

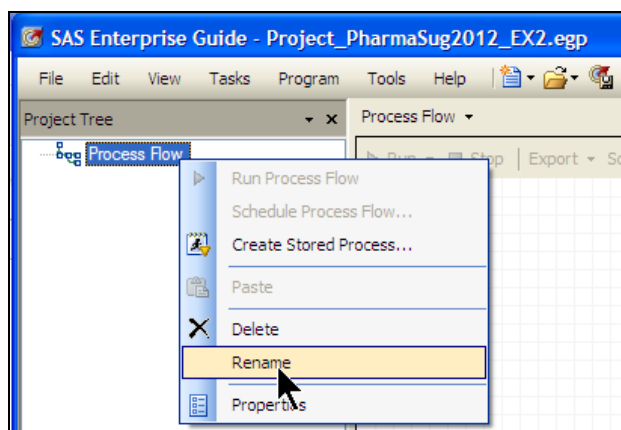
#### 1. Create a new project

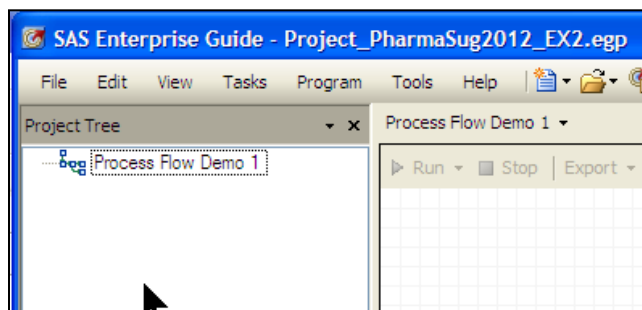


#### 2. Save the project with a new name. (Project\_PharmaSUG2012\_EX2.egp).

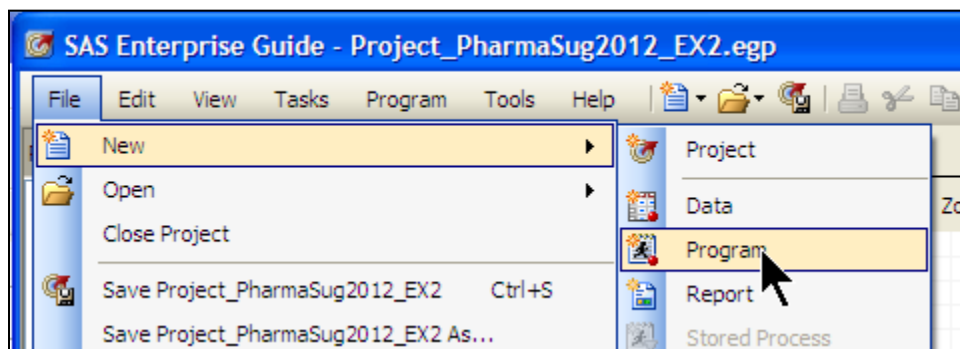


3. **Start a new Process Flow** within that project. Note that a blank "Process Flow" already exists. We will rename it to "Process Flow Demo 1"

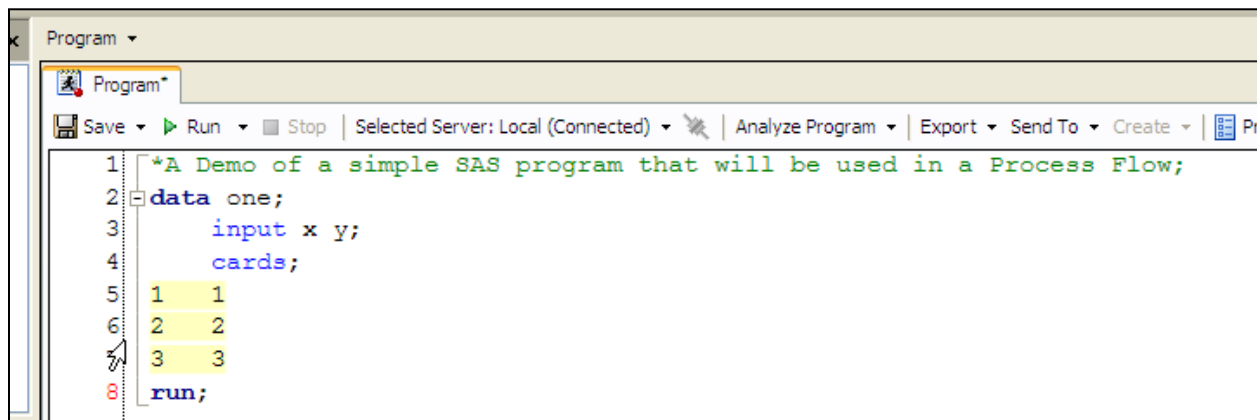




4. Now add a SAS Dataset via a program (rather than accessing an already existing dataset, we will do that later).



5. A program editor will open and we will enter the code in the program editor using classical SAS programming statements.



6. At this stage, we will hit the Run selection immediately above the code and execute the code. The resulting SAS dataset will be opened in a Window.



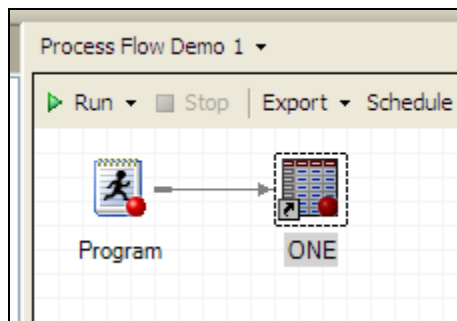
Program ▾

Program\* Log Output Data

Filter and Sort Query Builder

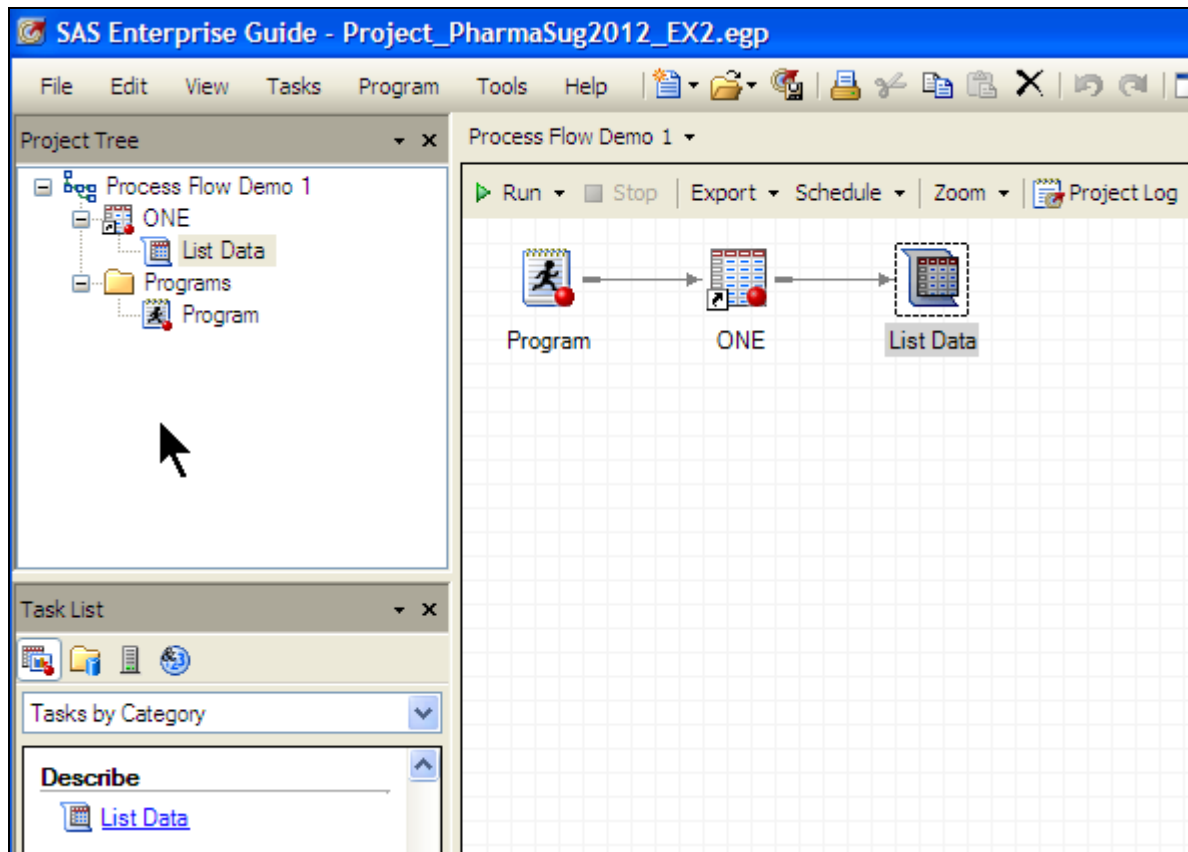
	x	y
1	1	1
2	2	2
3	3	3

7. Hit **F4** to return to the **Process Flow Diagram** and the dataset appears in the flow.

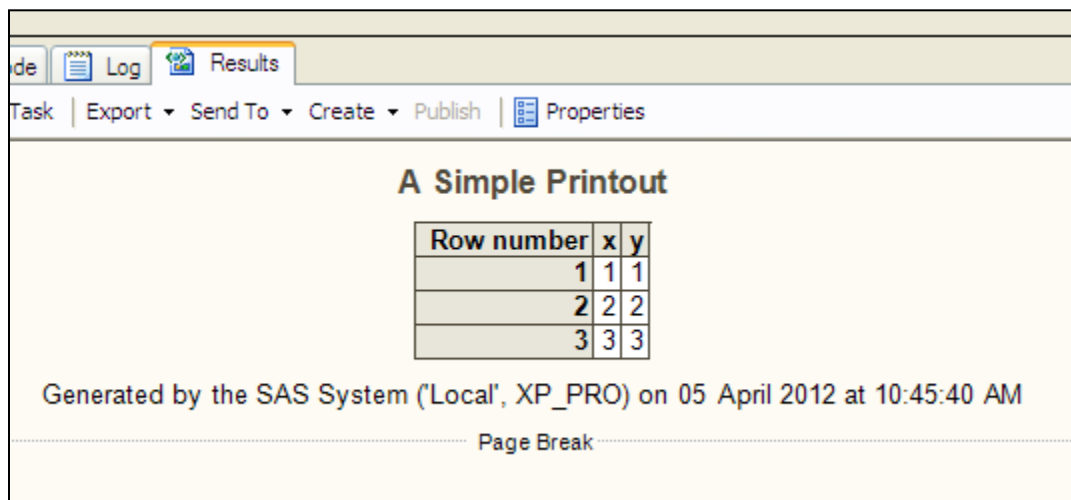


8. In the **Task List Window**, find the **List Data** icon, select it and drag and drop it on top of the work.one dataset in the Process Flow. The arrow is automatically drawn.



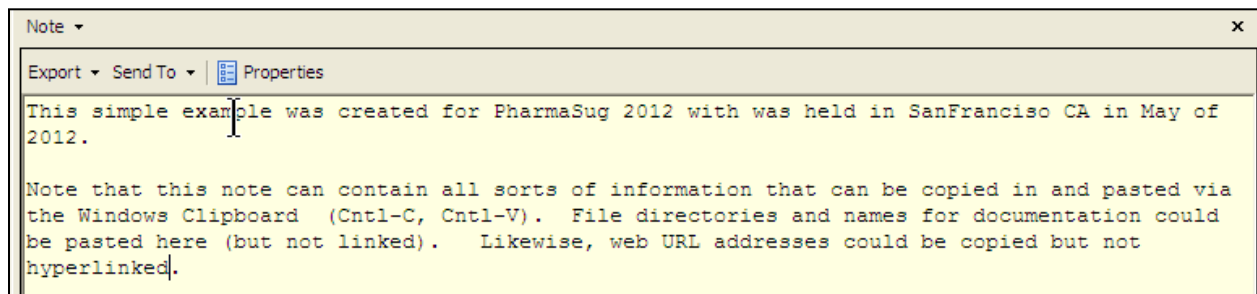
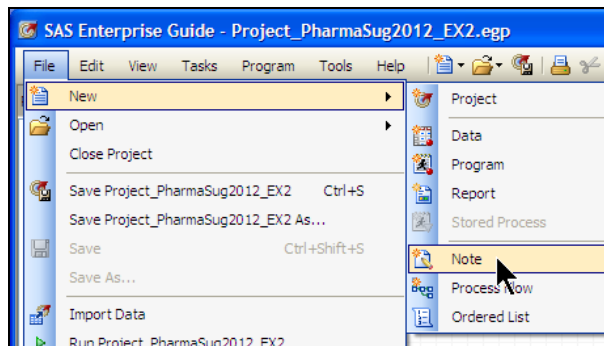


9. Now hit **Run** button above the **Process Flow** and the Process Flow including the report is generated. The report opens:

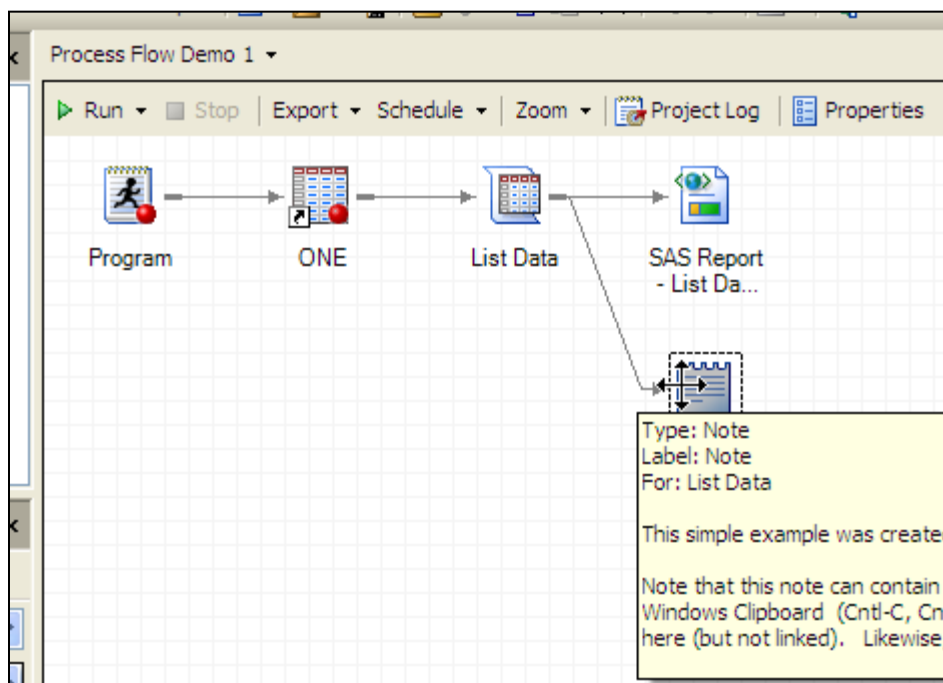


10. Hit the **F4** button to return to the **Process Flow**, go ahead and save the project. There in 10 easy steps is a simple introduction to the concepts of using all key elements in EG.

11. Insert a note for a little documentation. This note appears in the Process Flow.



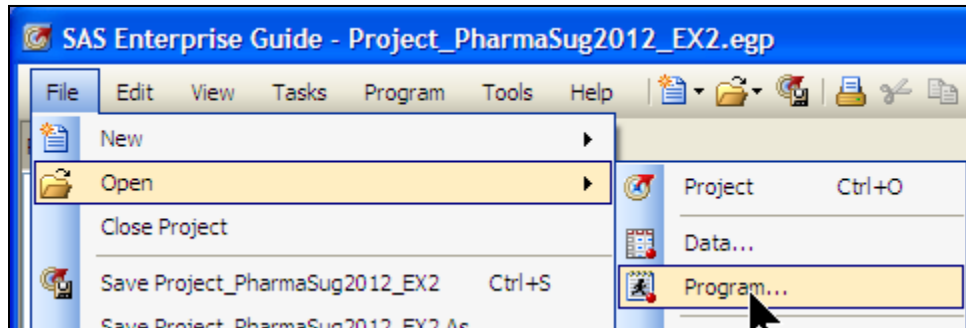
**12. Hit F4 and return to the Process Flow.** The note shows in the Process Flow and is context sensitive.



#### 4.1 What Is the File Structure? What are the Key Component Concepts of EG?

As seen in the short example above, EG projects are stored in an "\*.egp" file. This file contains Process Flows contained within a project as well as other elements..

SAS program code may be included in the Process Flow or it may be stored externally (as a .SAS file) and referenced in the Process Flow. Most programmers are likely to store the code externally for ease of access outside of SASEG. Externally-coded SAS files may be opened into the Process Flow as shown below:



Permanent SAS datasets on the other hand are not usually included in the project file while "work." SAS datasets may or may not be stored internally (you will be prompted on exit) when closing a project.

Within Process Flows, when moving from left to right, the structure is called a "Branch". On a branch, each element is a node. The node may be a SAS program, a query, a note, a task, or whatever. These also can be executed individually by highlighting, right clicking and hitting run.

#### 4.2 What Can I "Run" To Make Something Happen?

You can run the entire Project, a Process Flow, a branch, or a node on a branch. You can also run programs or highlighted sections within programs from the program editor. There are multiple places in menus, pull-down menus, right mouse clicks and more where you can run various items.

#### 4.3 How is a Process Flow code processed?

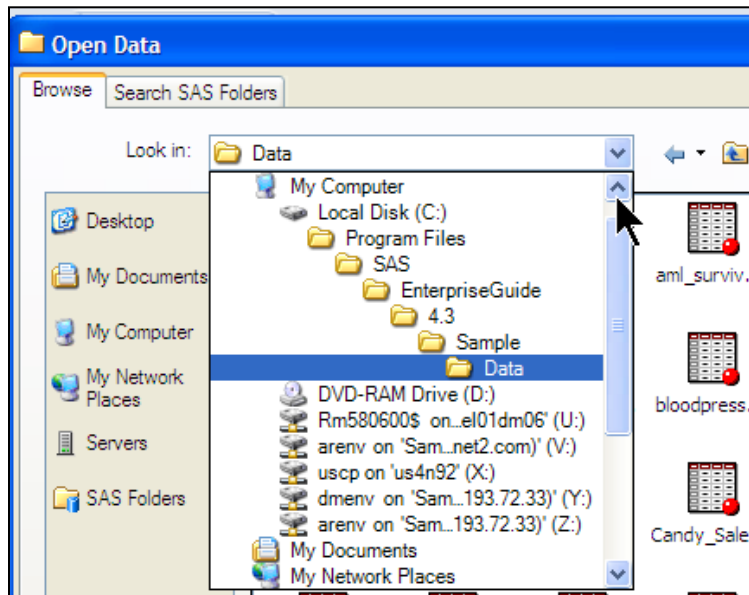
Code processing starts at the top of the window and moves left to right. Processing then moves down the diagram and again works left to right. Eventually, the diagram is processed in its entirety. Errors and comments are flagged by indicators on the icons in the flow (red X's, etc).

#### 4.4 How to Interrupt Processing

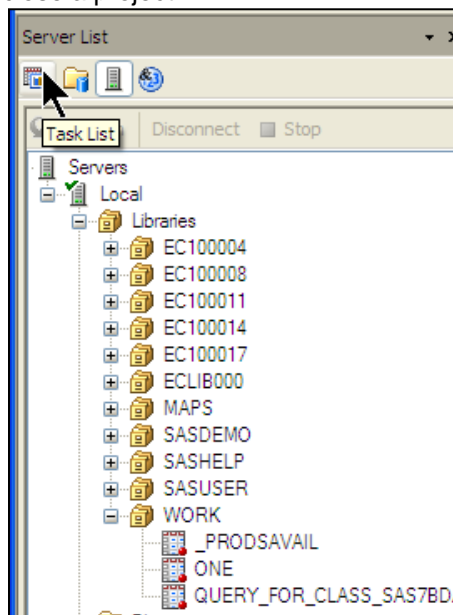
If the computer is taking a long time and appears to be "hung up", go to the task window (bottom of screen). This window serves as a monitor. Select and right click on the executing task and select "Stop". If that does not work, click on "End SAS Process". Be aware that this technique may end the process in the EG, but it might still be executing on a server (in the case of the host PC ---- the local SAS session (note this does not say EG session, it says SAS session)). Go to Task Manager in Windows (Cntrl-Alt-Del) to kill the SAS session which is "out of control". Consult your SAS administrator on stopping SAS sessions on Unix machines that are running remotely if you do not understand how to stop these.

#### 5.1 SAS Datasets -- The Sample Datasets

The default installation of EG will point you to a host of sample datasets for you to practice with as shown below. You get to this by File, Open, Data then follow the drilldown shown. Of course you could use similar techniques to get to other SAS datasets located on any accessible drive on your system.



In your default installation, libnames are issued with the names as shown in the following figure. To see this listing, switch the Task List Window over to Server List and drill down as shown. **Important: Note that your work files for SAS Datasets are also here.** You will be prompted to keep these when you close a project.



## 5.2 SAS Datasets -- Your Personal Datasets and Others

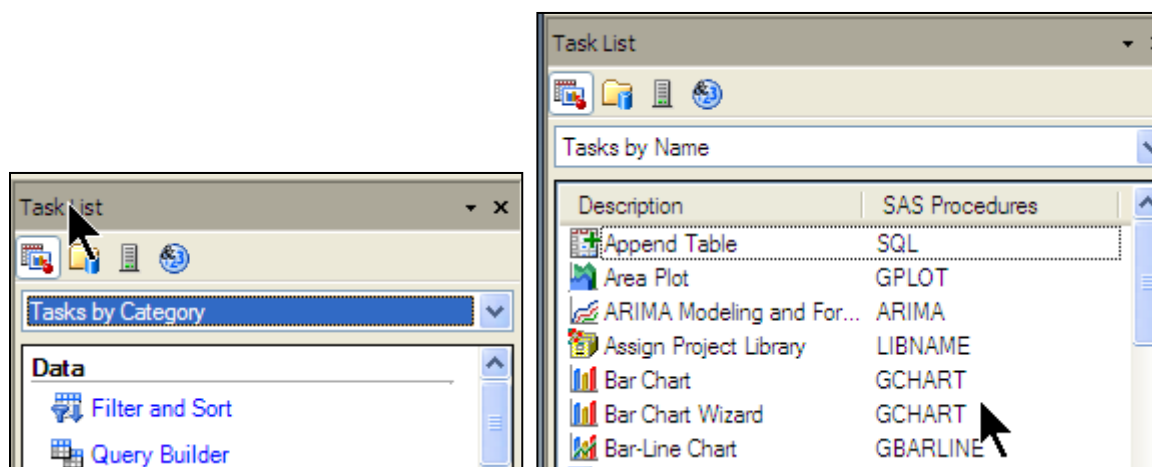
The opening and saving of SAS datasets within EG is the same as it is in any other SAS programming effort with one slight exception which will be addressed later. In programming code mode, a libname

must be issued to the location of the SAS dataset and then reference the dataset by a 2 level name in the form of "libname.dsname" to read or write.

Additional capabilities to open SAS datasets include doing a File Open Data as shown above. The dataset will open on the Process Flow Screen. Also, a listing of SAS datasets can be made from the Server List Window and a specific dataset on a specific drive can be selected and dragged to the Process Flow. Any dataset in a folder of a drive accessed by your PC can be used.

## 6.1 Tasks and Wizards

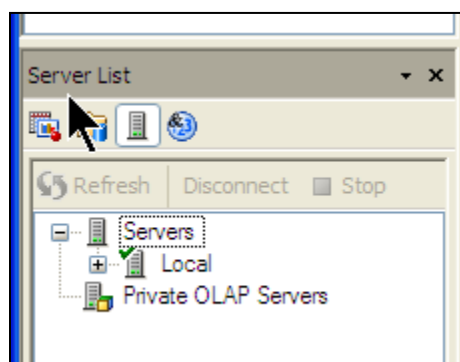
The "Tasks and Miscellaneous" window contains a list of SAS Procedures that can be selected and dragged to the Process Flow windows, generally on top of a dataset to connect to it. Items in the Task List are presented either by its SAS procedure name or by its function as shown alternatively in the two windows below (Category on Left, Procedure Name on Right). Once the icons are opened on the Process Flow, tasks in the Process Flow such as selecting variables, query operations to be performed, etc. are very straightforward and easy to learn.



Several tasks also have wizards, an alternative for developing the code with a more "hand-holding" prompting type of approach and will not be discussed further in this paper.

## 6.2 Servers

By clicking on the Server icon in the "Tasks and Miscellaneous Window", you will see a listing of the current servers available to you. Most likely, this will look like the following:

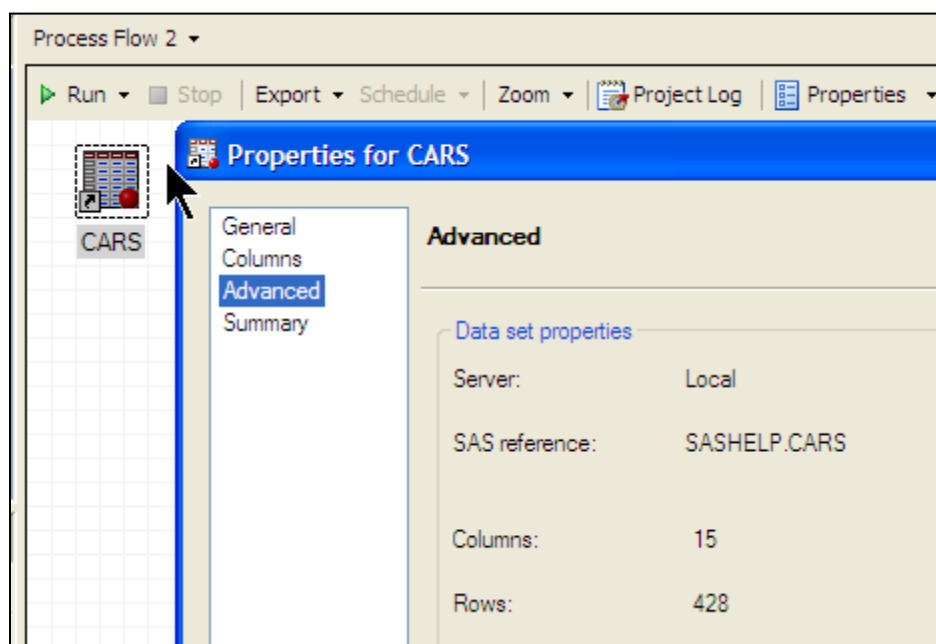


The "Local" server is your machine (indeed it is your PC Windows SAS). If, and only if, you and your

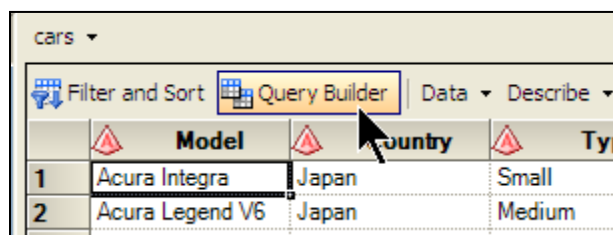
Unix servers have SAS with the IT (Integrated Technologies) components installed and properly configured will other servers surface in this list. This capability is highly desirable as you can execute your Process Flow, including Query Builder, on a remote machine effortlessly without the overhead of moving data files etc. The use of the IT components and hence, the use of SAS on other servers in an interactive fashion, is beyond the scope of this paper. BUT ---- and this is a BIG BUT, if you have SAS Connect software on your PC and on your Unix servers, you can do what is called RSUBMIT, which stands for remote submit in more-or-less a batch mode to allow code to be executed on the remote server and the results bounced back to you. RSUBMIT will be covered later in this paper.

### 6.3 The Query Builder

Perhaps the most useful tool for data exploration is the Query Builder. Among other things, data are filtered, sorted, hidden, reordered, merged, etc. via an interface that is relatively easy to use despite its deep capabilities. Behind the scenes, programming is done in the SQL (Structured Query Language). This code can be retained for future use. An in-depth coverage of the capabilities of this tool is beyond the scope of this paper. As an example of this capability, consider the following Process Flow. The dataset sashelp.cars has been dragged into the Process Flow window, the dataset was highlighted and right mouse clicked on it. Properties - Advanced was selected. The dataset contains 428 rows (observations) and 15 columns (variables).



I will run a query and filter on this to generate a quick small dataset for the European makes of cars in the report. I will click on the query builder tab above the data; or alternatively, I could drag and drop the Query Builder icon from the Task List onto the dataset icon in the Process Flow.



Query name: Query Builder

Computed Columns | Prompt Manager | Preview | Tools | Options

Add Tables | Delete | Join Tables

t1 (CARS)

- Cylinders
- DriveTrain
- EngineSize
- Horsepower
- Invoice
- Length
- Make
- Model
- MPG\_City
- MPG\_Highway
- MSRP
- Origin
- Type
- Weight
- Wheelbase

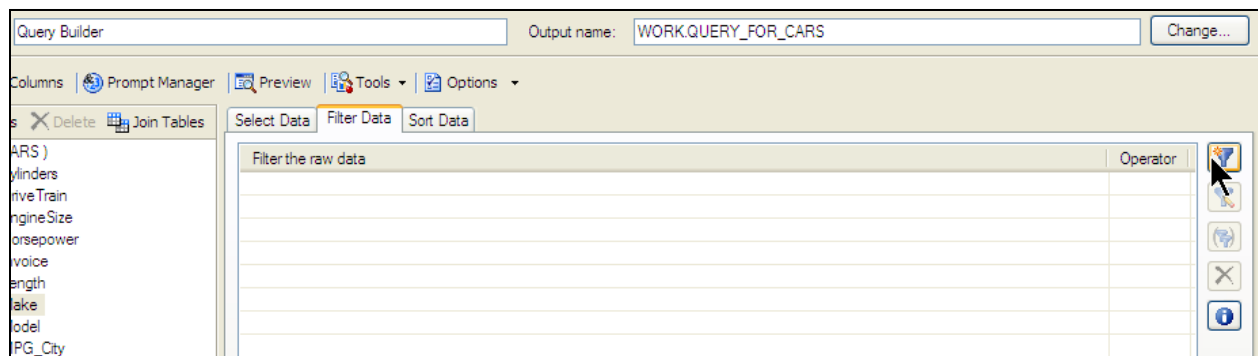
Select Data | Filter Data | Sort Data

Column Name	Identifier
Make	t1.Make

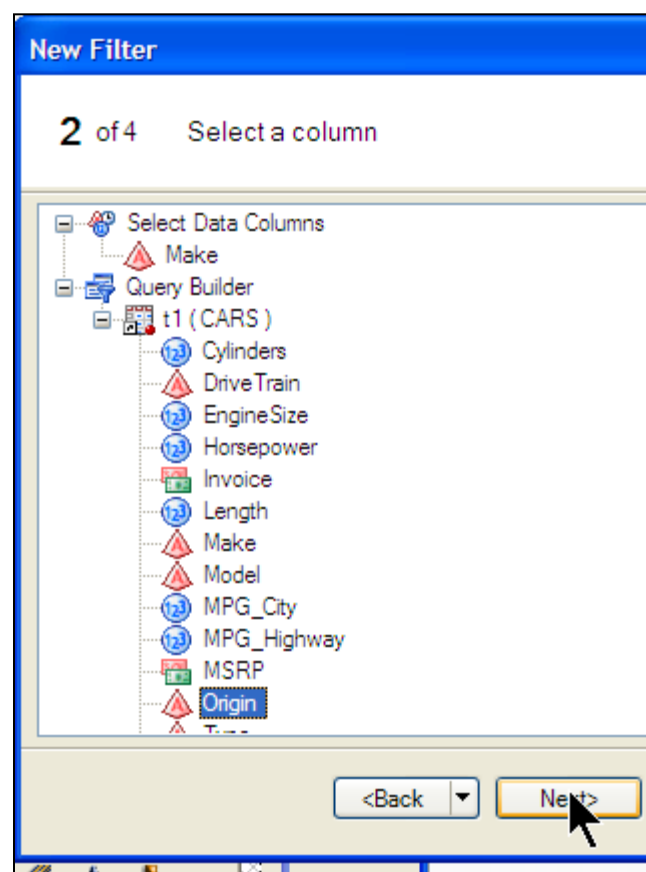
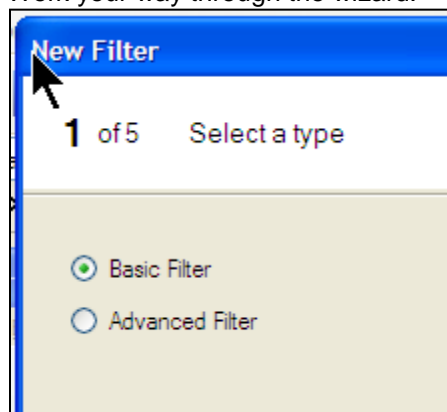
☒ Select distinct rows only

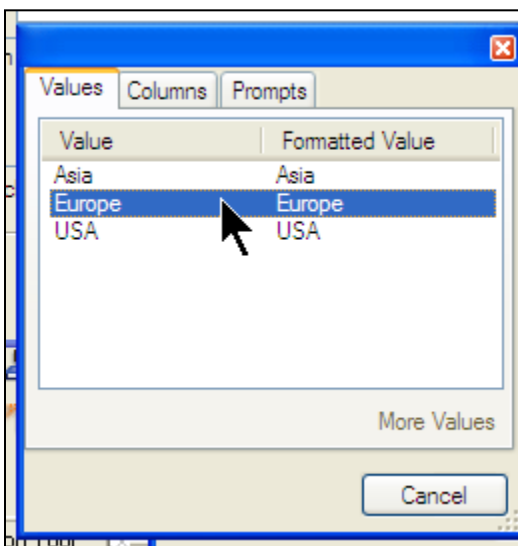
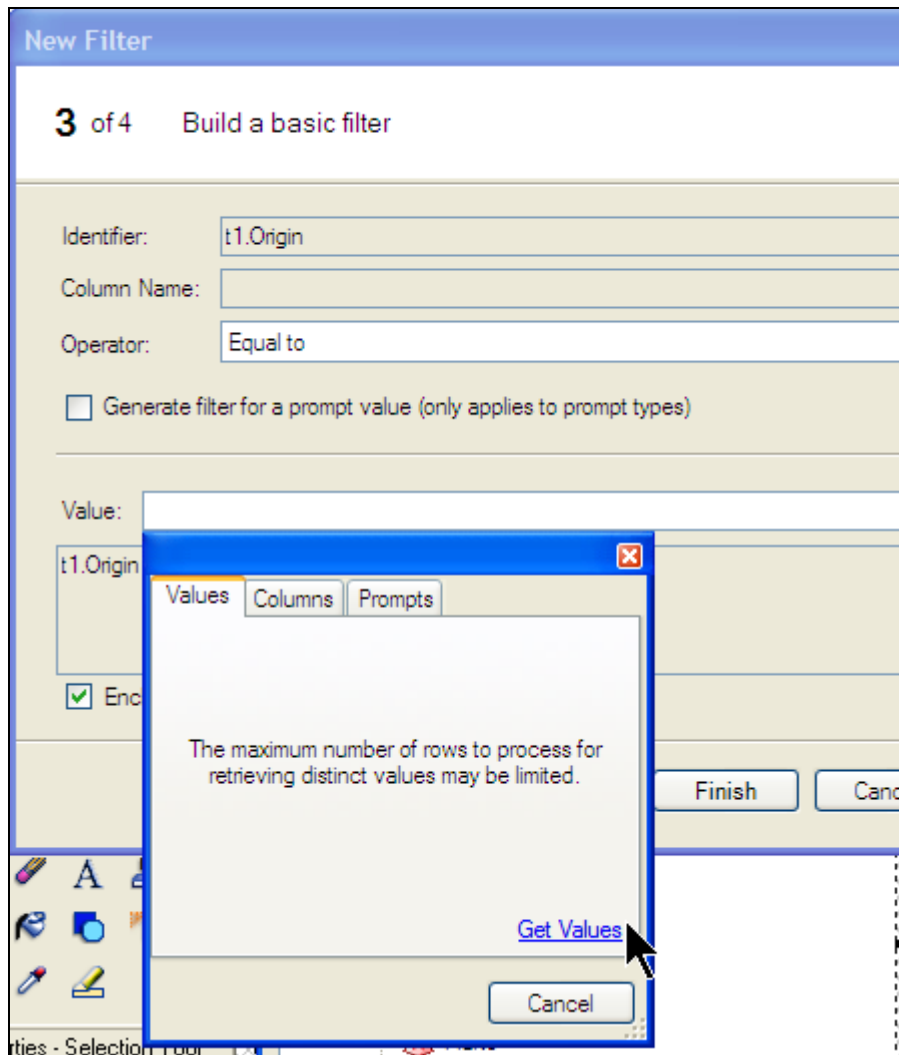
15





Work your way through the wizard.





**New Filter**

**3** of 4    Build a basic filter

Identifier:

Column Name:

Operator:

☐ Generate filter for a prompt value (only applies to prompt types)

Value:

`t1.Origin = 'Europe'`

☒ Enclose values in quotes

<Back    Next>    Finish

Click on Finish and you will see the filter.

**Query Builder for Local:SASHELP.CARS**

Query name:     Output name:

Computed Columns    Prompt Manager    Preview    Tools    Options

Add Tables    Delete    Join Tables

t1 (CARS)

- Cylinders
- Drive Train
- Engine Size
- Horsepower
- Invoice
- Length
- Make
- Model
- MPG\_City
- MPG\_Highway

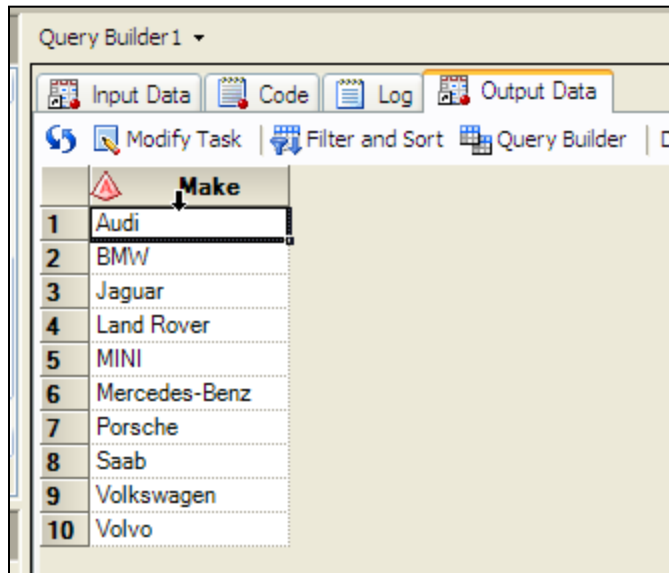
Select Data    **Filter Data**    Sort Data

Filter the raw data

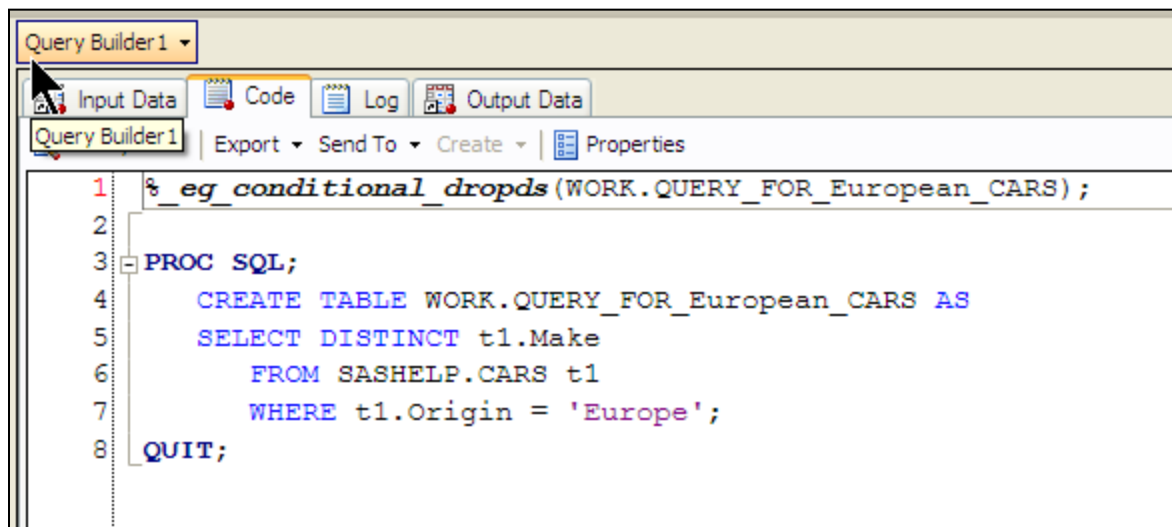
Where

t1.Origin = 'Europe'

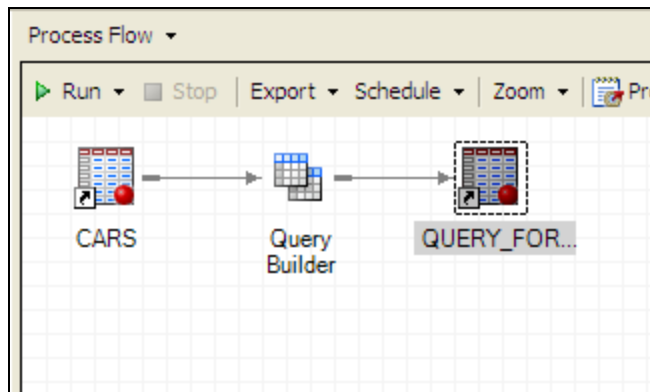
After getting our query code developed, we hit the “Run” button at the bottom of the screen. The SQL code is generated and run behind the scenes and the resulting dataset is presented in the Process Flow. The resultant dataset is composed of the makes of European cars in the dataset..



The SQL Code is available for other uses (if desired) from the Code tab.



Here is the Process Flow that has been built for this example (hit F4 to see it):



Obviously the Query Builder is a quick easy means of exploring the contents of a SAS dataset. Likewise, it is a great way to build SQL code for use in programs that might be run on Unix, etc. The capabilities are much more extensive than those available in the interactive SAS Viewtable facility and can be saved between sessions

### 7.1 Using Enterprise Guide on the PC with SAS Datasets Residing on Unix.

The next section will deal with issues surrounding how to address the use of Enterprise Guide on the PC when the datasets are located on Unix or other platforms. These issues are the same when using Enterprise Guide that they are on Interactive PC SAS. Machine performance can be greatly hampered when a less-than-well-thought-out work approach is used. Large complex datasets will suffer from very poor performance if they are addressed directly remotely without being downloaded, or being processed via Rsubmit (remote submit).

### 7.2 Where Should Your SAS Datasets Reside for Program Development?

If we assume that you do not have the Integrated Technologies software components installed, we have 3 choices to consider:

1. Leave the datasets on Unix, attach the data disk on Unix as a drive on your PC.  
Advantage: easy to do.  
Disadvantage: performance can be miserable with large datasets on underperforming networks.
2. Copy the dataset from Unix to a local PC drive for quick access while developing code.  
Advantage: quicker response, particularly with Query Builder.  
Disadvantage: takes time and resources, particularly if the data is being updated frequently. Data maintained in duplicate places is dangerous and inefficient.
3. Submit code requiring resource intensive processing from SAS Enterprise Guide over to the Unix server where both SAS and the SAS dataset reside using the RSUBMIT facility associated with SAS Connect software.  
Advantage: extremely efficient use of the most powerful processor being used in conjunction with data where it natively resides.  
Disadvantage: It takes a little bit of learning to figure this out. Read on!

Results of tasks and programs which were run on Unix will display on the Process Flow. Logs and Outputs are bounced back to EG.

Any work datasets (or other datasets) in the Unix session can be downloaded from Unix to the PC, Generally these are smaller and can then be used in the Process Flow for continued development much more efficiently.

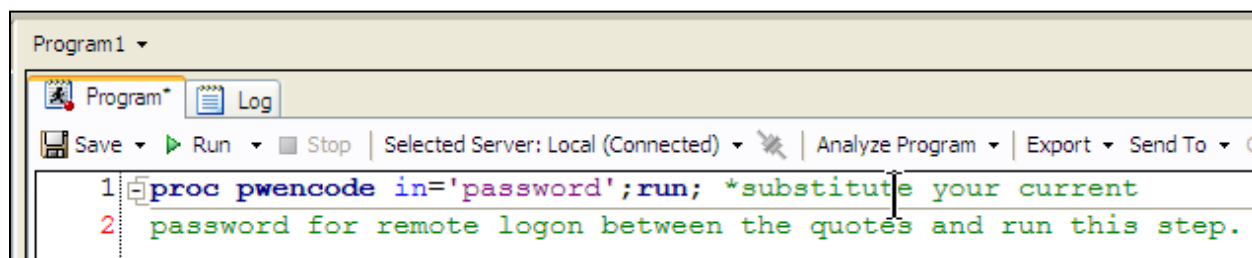
## 8.1 Using RSUBMIT to Send Code to Unix for Processing

In this section of the paper, we will explore using RSUBMIT to run code residing on EG on the PC to a Unix server with a larger processor that also is source of the SAS datafiles to be processed. There are several considerations:

1. Security must be addressed as the PC session will need to be logged on to the Unix server, not totally unlike a user sitting at a terminal accessing that server. Direct hard coding of passwords in scripting files would be extremely risky. Thus SAS Institute has developed an encoding system for passwords that will generate codes totally useless to anything other than SAS Connect and comply with all security methodologies in place.
2. SAS must reside on the distant server as well as the SAS Connect product.
3. Libnames must be issued in the code for the folders containing the datasets on the server.
4. Work space must be allocated for the SAS work datasets on the server
5. If you wish to have datasets passed back to the EG session on the PC, you will need to use proc download or some other means to achieve this.

## 8.2 SAS Security Provisions in Using Connect with Unix Servers

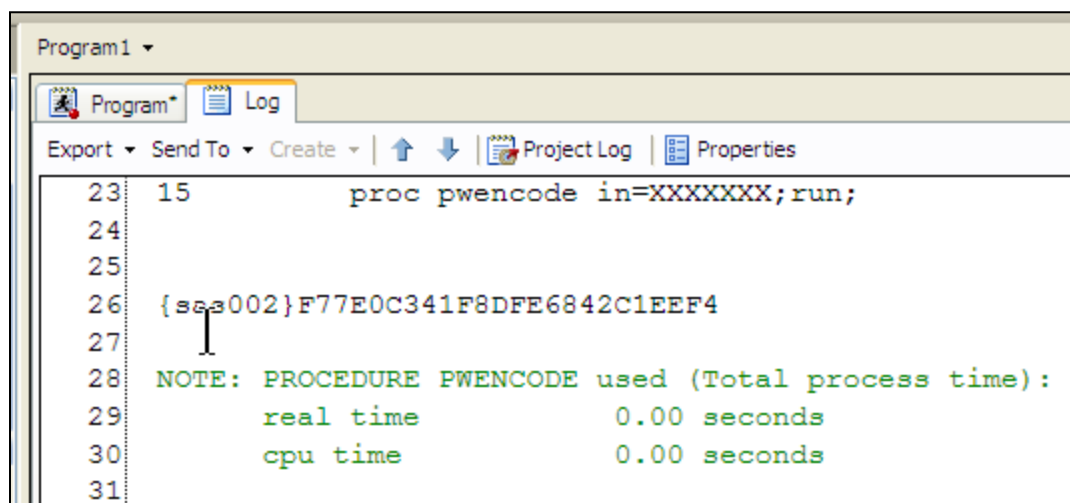
1. Users will use Proc Pwencode in a SAS session to convert the logon password for the account in Unix over to an encoded value that is useful for no other purpose. This conversion is taking place here in EG in this example, but could be done in interactive SAS or other types of SAS installations. In this example I am not using a real password, simply a bunch of X's to show the technique.



The screenshot shows the SAS Enterprise Guide interface. The 'Program' window is active, displaying the following code:

```
1 proc pwencode in='password';run; *substitute your current
2 password for remote logon between the quotes and run this step.
```

2. SAS will push an encoded password useful only to SAS Connect to the log.



The screenshot shows the SAS Enterprise Guide interface with the 'Log' window active. The output of the PROC PWENCODE command is displayed as follows:

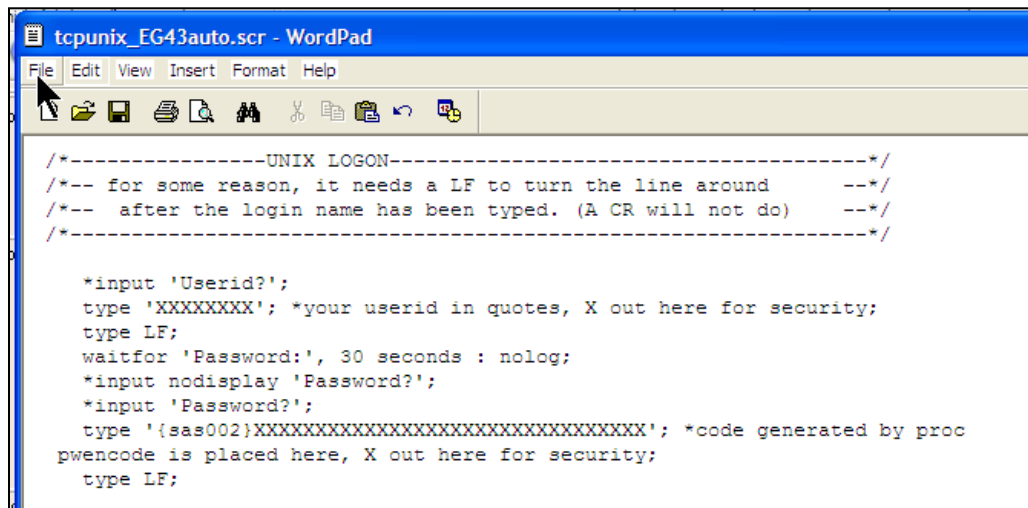
```
23 15          proc pwencode in=XXXXXXX;run;
24
25
26 {sas002}F77E0C341F8DFE6842C1EEF4
27
28 NOTE: PROCEDURE PWENCODE used (Total process time):
29      real time          0.00 seconds
30      cpu time           0.00 seconds
31
```

3. Place this password in a script file.

If you have both EG 4.3 and SAS Connect software on your machine, the installation will have placed the following script file on your machine. This will need to be slightly modified as shown below with an editor such as Wordpad. The file location and name is:

C:\Program Files\SAS\SASFoundation\9.2\connect\saslink\tcpunix\_EG43auto.scr.

The portion of the file containing reference to Unix logon needs to be modified as to userid and password as shown in the following screen snapshot. Place your inputs after the type command in quotes substituting you values for the XXX's.



```

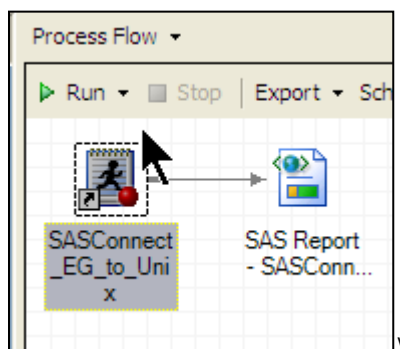
/*-----UNIX LOGON-----*/
/*-- for some reason, it needs a LF to turn the line around  --*/
/*-- after the login name has been typed. (A CR will not do)  --*/
/*-----*/

*input 'Userid?';
type 'XXXXXXXX'; *your userid in quotes, X out here for security;
type LF;
waitfor 'Password:', 30 seconds : nolog;
*input nodisplay 'Password?';
*input 'Password?';
type '{sas002)XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'; *code generated by proc
pwencode is placed here, X out here for security;
type LF;

```

### 8.3 Coding the program for RSUBMIT program in EG

The Process Flow being used is as shown in the following:



The contents of the program highlighted above are shown in the following figure and are fairly self-explanatory as to the steps involved:



```

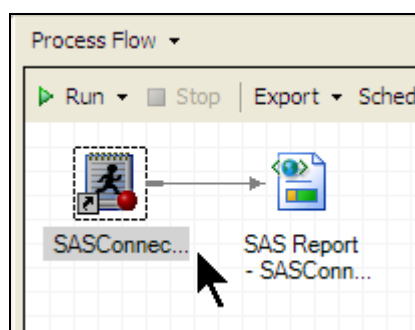
17 FILENAME RLINK '!sasroot\connect\saslink\tcpunix_EG43auto.scr';
18 %let rmtnode= ;
19 options remote=rmtnode comamid=tcp;
20 signon ;
21 %*** to turn off endsas when an error is encountered ;
22 options noerrorabend ;
23
24 *MAKE WORK LIBRARY FOR THE UNIX SESSION;
25 libname unixwork slibref=work server=rmtnode; *local submit;
26
27 rsubmit rmtnode;
28 options date ;
29 title 'Content of Dataset sashelp.class on Unix Server';
30 proc contents data=sashelp.class;
31 options obs=5;
32 proc print data=sashelp.class;
33 run;
34
35 options obs=max;
36 data work.class_unix;set sashelp.class;run;
37 proc contents data=work.class_unix;
38 run;
39 proc download data=work.class_unix out=test_class_trans_from_unix;run;
40
41 endrsubmit;
42 signoff ;

```

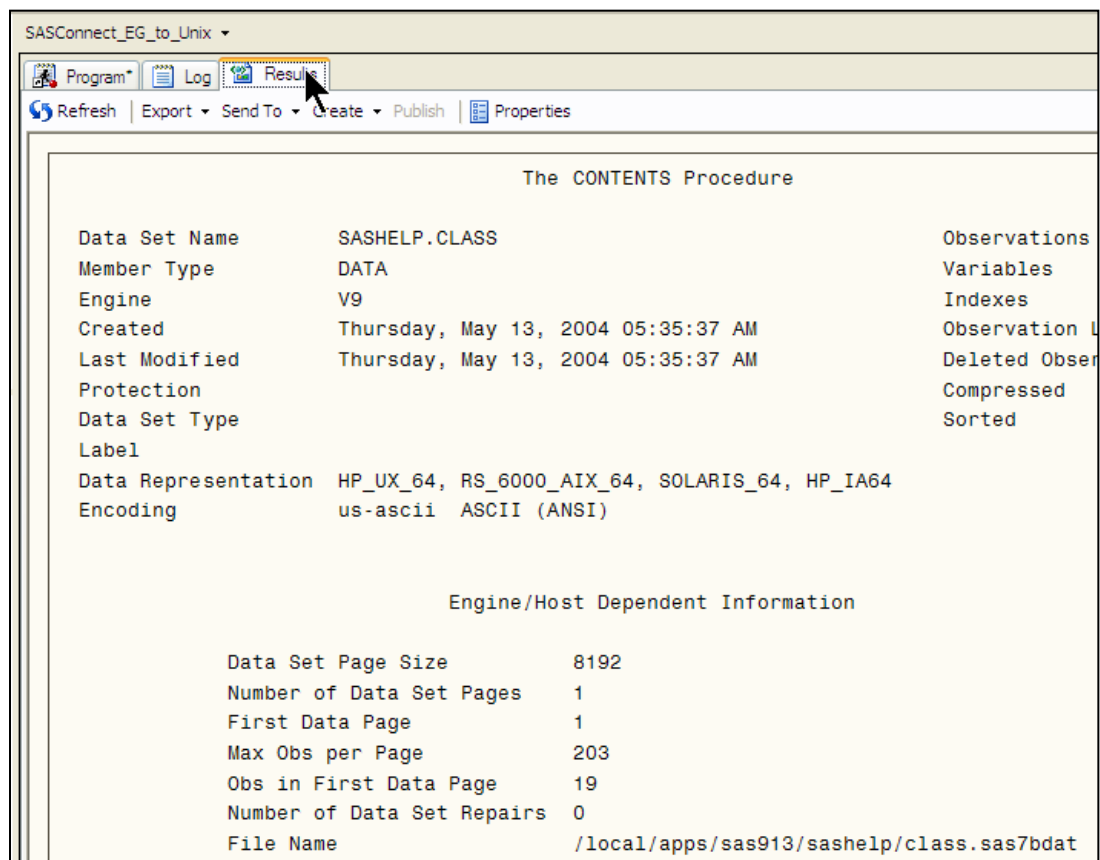
← Name removed for security. It would be something like: servertest.company.com

You will note in the program that necessary connections to a server that has SAS are being made. The dataset we will work with is on the remote server, not on the local PC that has EG on it. We will do a few simple procs (contents and print) on the server. We will then make a copy of the sashelp.class dataset on the server and download it to our EG session on the PC with Proc Download.

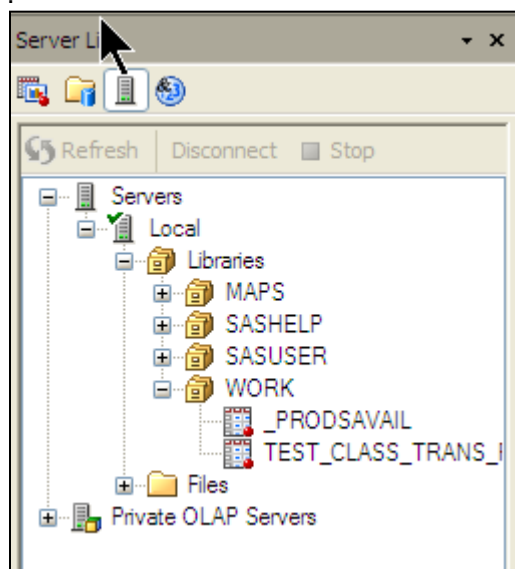
Looking at the Process Flow in EG in the following figure we have one program and the resulting output that was produced on the Server Side PC residing on the EG Process Flow on the PC



If we double click on the program node, we get the following screen with tabs for the code (program), log, and the output. All of these reside on the PC, even though the actual code was run on the Unix server. This figure shows a portion of the output (Results, the SAS print file).



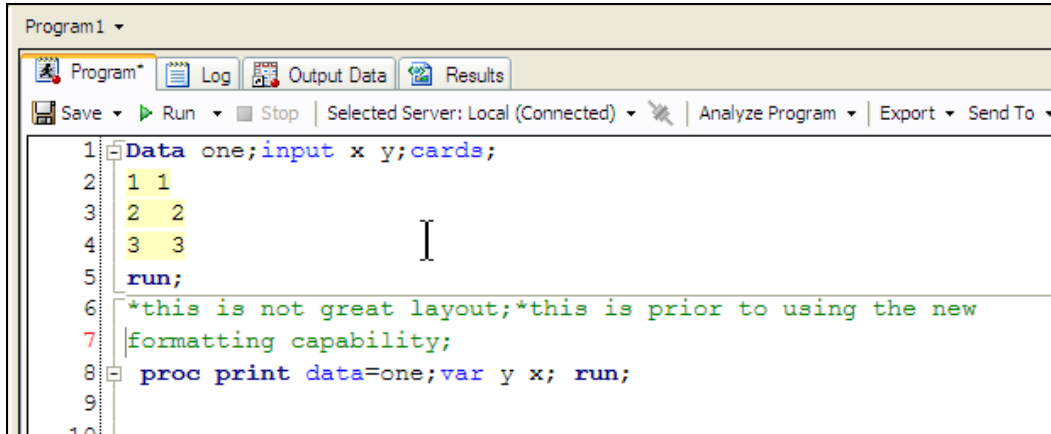
One last figure for this effort: The EG server list window when expanded shows that the work datasets for the Local EG SAS session now contain the dataset named work.test\_class\_trans\_unix. This was placed here by proc download (see code several frames higher). This dataset could be dragged to the Process Flow window to start yet another branch on the Process Flow.



## 9.1 The Program Editor

While many programming and data analysis operations are now icon driven on the Process Flow, there is still a need to do traditional SAS programming. The new enhanced editor provides several exciting new capabilities, several which are highlighted here. This is yet another reason for the traditional SAS programmer to start using EG.

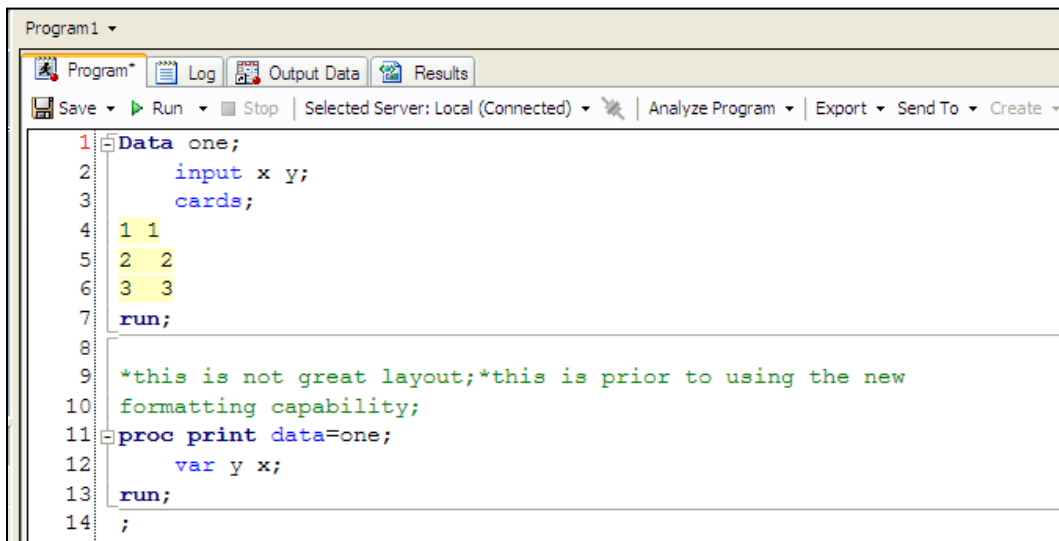
1. Formatting Code. Take a look at the following code which is not laid out in such a fashion to make the structure of what is happening very easy to follow:

A screenshot of the SAS Program Editor window titled 'Program1'. The window has a menu bar with 'Program\*', 'Log', 'Output Data', and 'Results'. Below the menu bar is a toolbar with icons for 'Save', 'Run', 'Stop', 'Selected Server: Local (Connected)', 'Analyze Program', 'Export', and 'Send To'. The main text area contains the following SAS code:

```
1 Data one;input x y;cards;
2 1 1
3 2 2
4 3 3
5 run;
6 *this is not great layout;*this is prior to using the new
7 formatting capability;
8 proc print data=one;var y x; run;
9
10
```

The code is not formatted, with the data values and the PROC statement not indented. A vertical cursor is positioned at the end of line 4.

Highlight the code and select Edit Format from the top menu and the following changes are made to make the code much easier to read (note: this example is not really in bad shape to begin with, it just shows the capabilities).

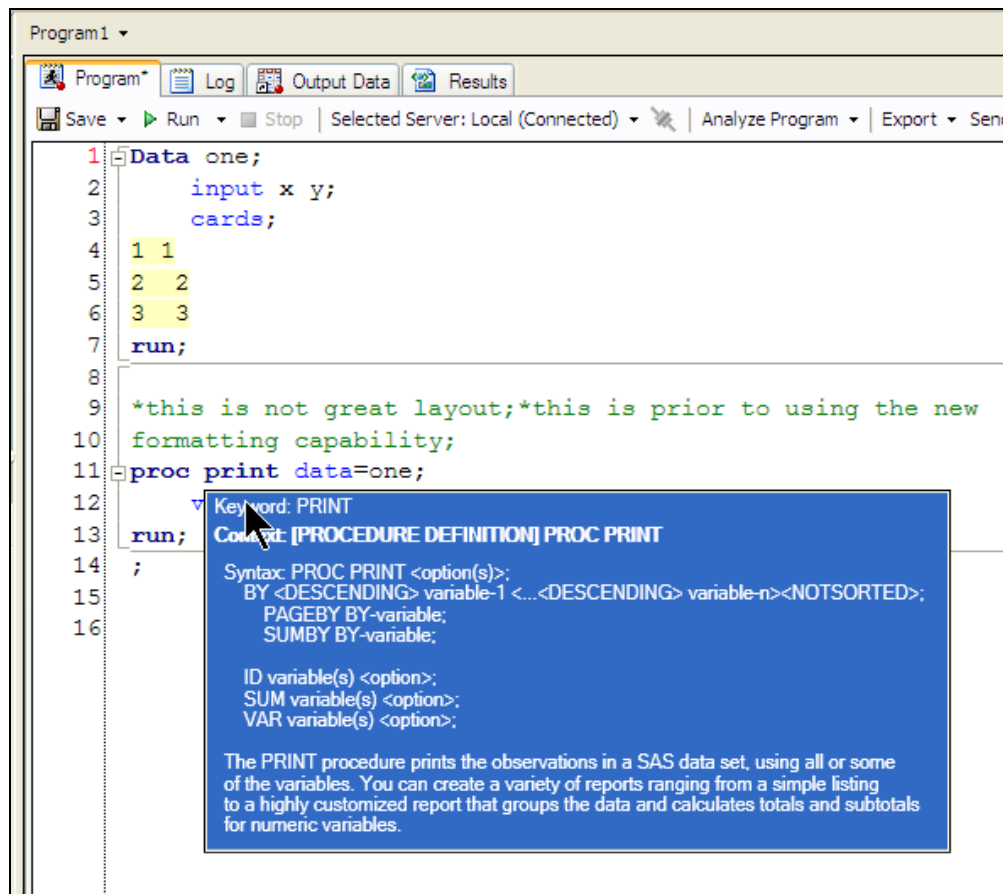
A screenshot of the SAS Program Editor window titled 'Program1', showing the same code as the previous screenshot but with improved formatting. The code is now indented to clearly show the structure of the DATA and PROC statements. The window's menu bar and toolbar are identical to the previous screenshot.

```
1 Data one;
2     input x y;
3     cards;
4 1 1
5 2 2
6 3 3
7 run;
8
9 *this is not great layout;*this is prior to using the new
10 formatting capability;
11 proc print data=one;
12     var y x;
13 run;
14 ;
```

2. Context Sensitive (Hovering) Assist (help) for Syntax Options

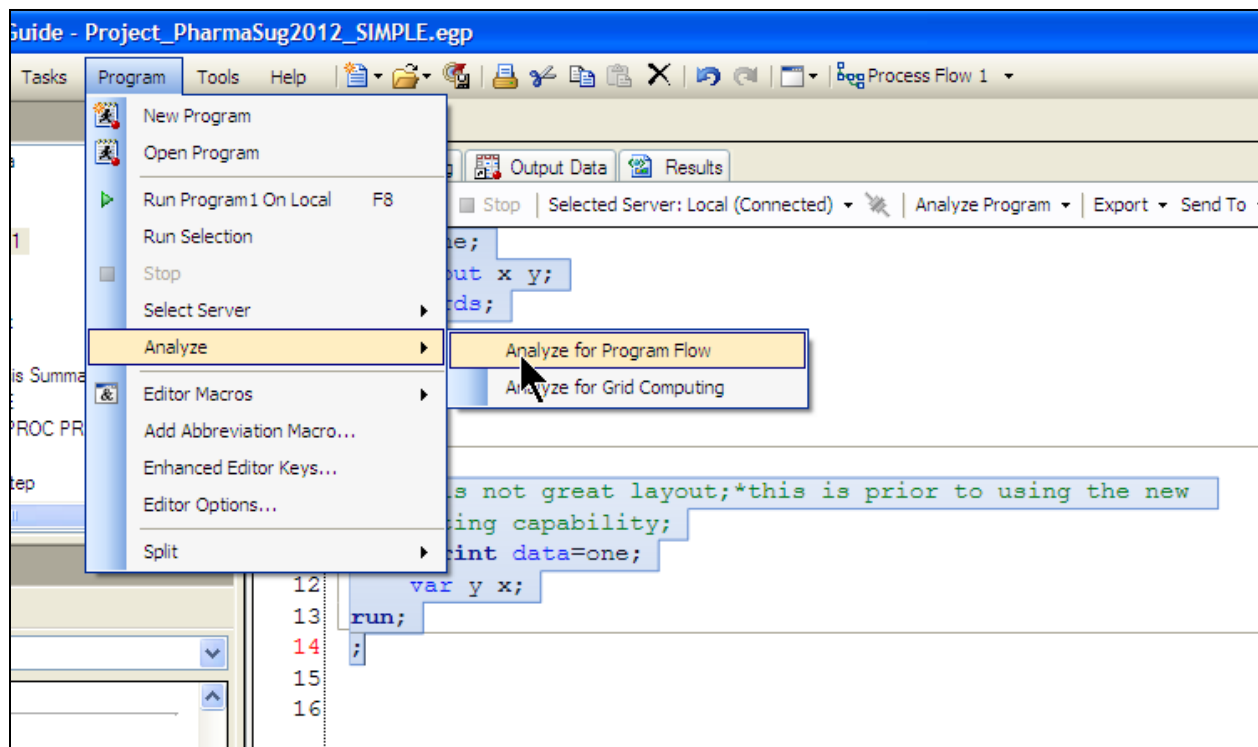
When passing the cursor over the text being typed, help on syntax appears. This can be highlighted and

copied by Windows (Cntl-C) and pasted into the file being edited (Cntl-V). This is then available for editing.

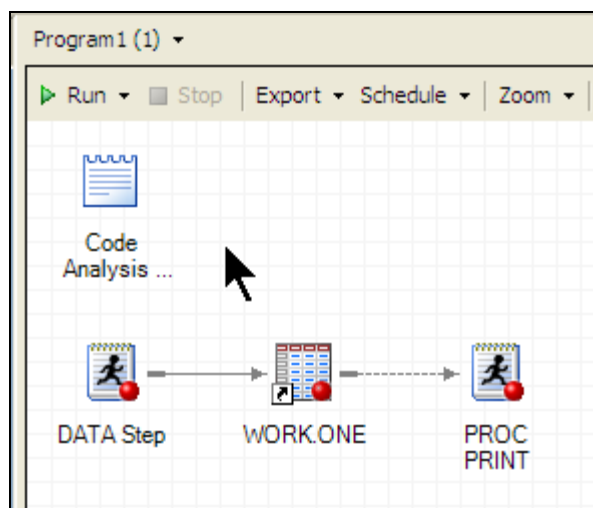


### 3. Converting programs to Process Flow.

By highlighting the program code above under step 1 and selecting Program - Analyze - Analyze for Program Flow --

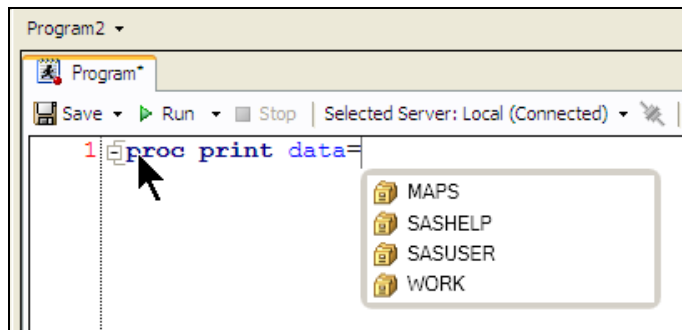


Results in a diagram in the Process Flow of this code:

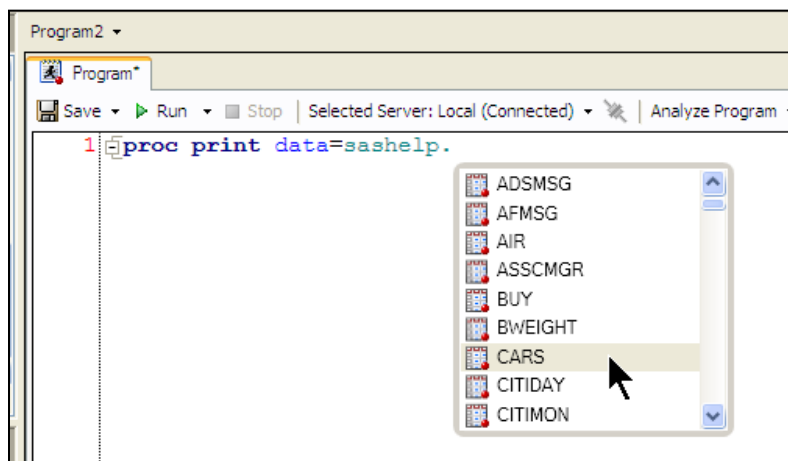


4. Assistance in selecting libnames, dataset names and variable names.

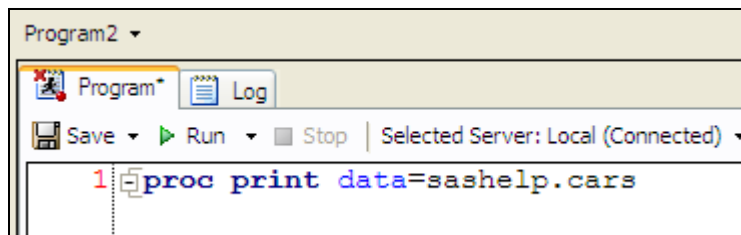
When working in the editor, first pick off the libname from the list of those available (in this case, SASHELP):



Then pick the dataset - Select cars

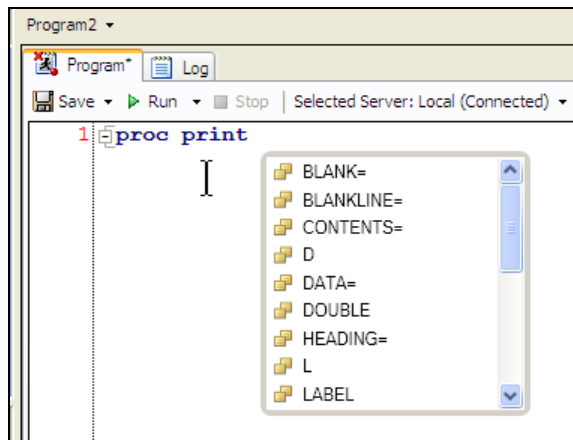


And the data set name is completed.



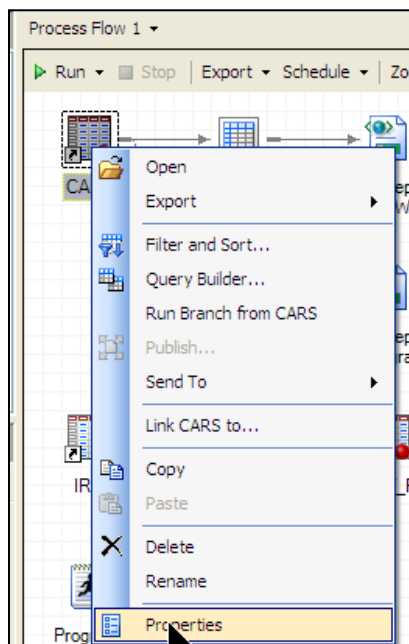
5. Help with other options available for the task at hand --- take our above example, after you type "proc print", what are some of the options available?

Over 20 options are available, a few of which are shown in the following:

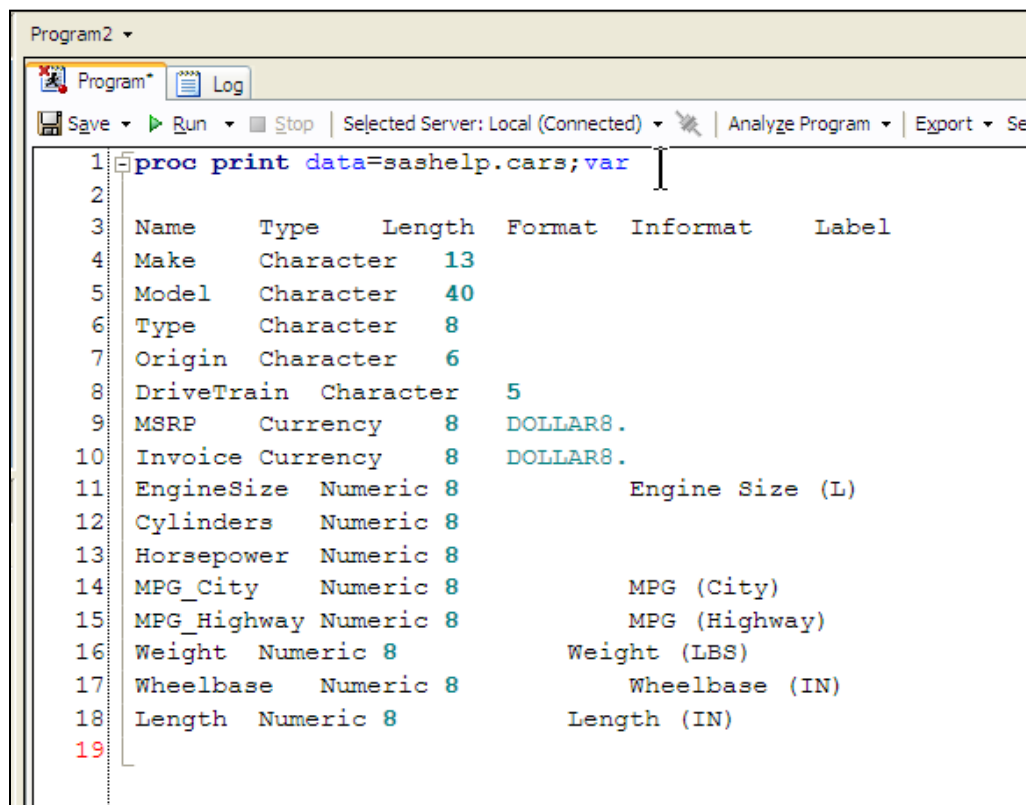
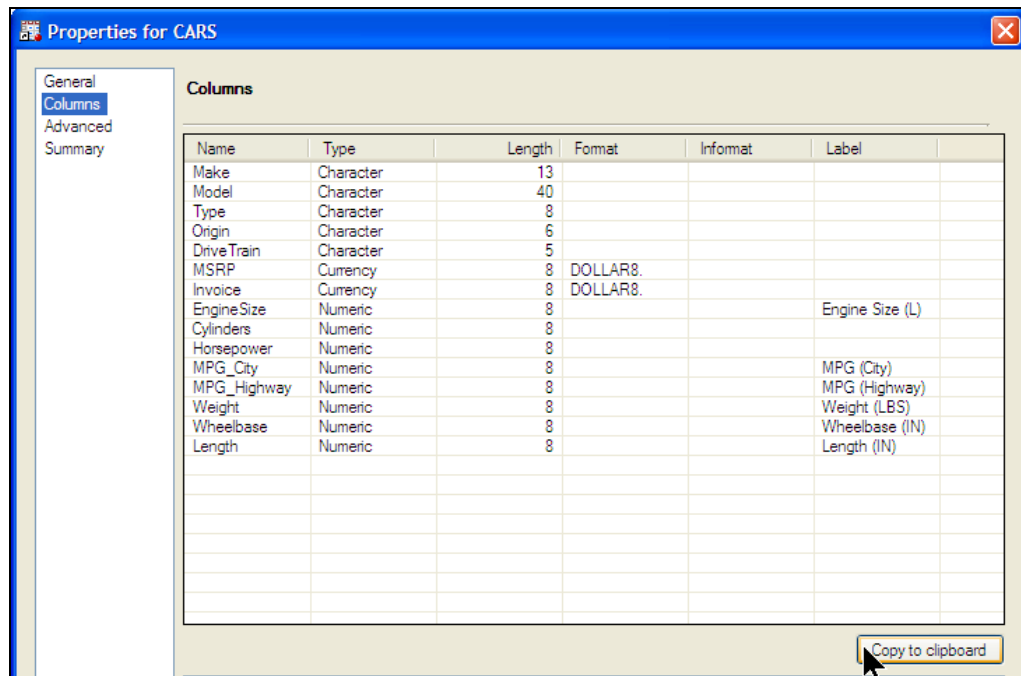


6. Bring variable names into the editor for programming:

While a direct tool is not available, there is a convenient work-around furnished by the environment of EG. Drag a dataset onto a Process Flow. Follow the steps in the next series of filters. Highlight and click on it with the right mouse, Properties - Columns - Copy to Clipboard (lower right). Paste into editor and clean it up.. Pretty?? No ---- but everything is here for windows operations such as copy, paste, drag and drop. Don't forget about holding the Alt key while dragging the left mouse button to "box off or lasso" areas for selection for copy and delete operations in the editor.







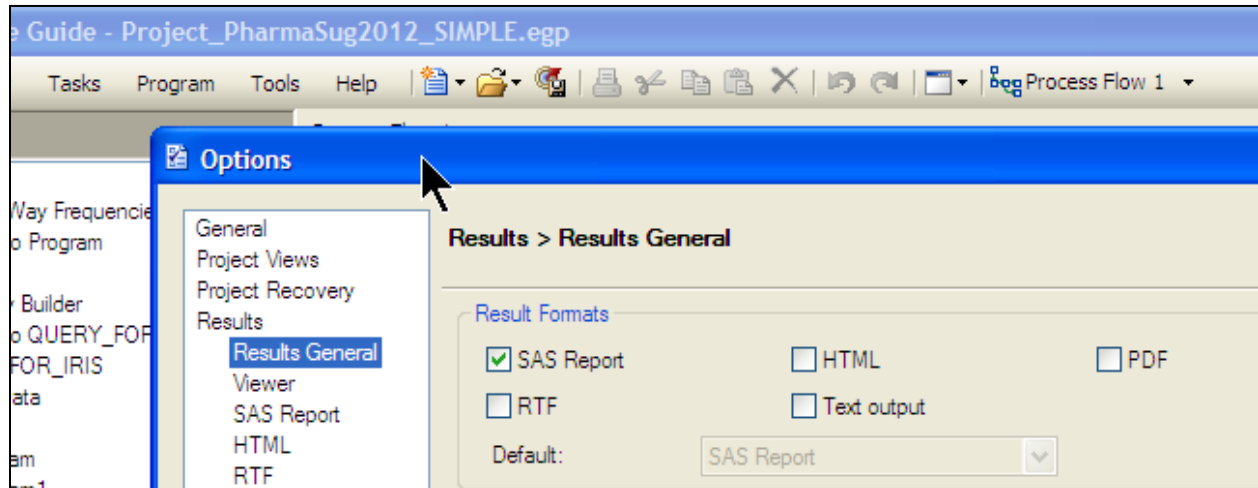
7. A host of other options for the editor, including line numbering, appearance, autocompletion (which can be deactivated if desired), colors, macros etc. are available.

## 9.1 Exporting Code

Sas code may be exported from the File - Export menu selection. Projects or Process Flows may be exported as SAS code to files. For general use, deselect the "include generic wrapper" option in the export dialogue that comes up. Most programming efforts will not need these.

## 10.1 Output

Output from reports, procs etc can be generated in a wide variety of formats with numerous options. See Tools - Options - Results.



## 11. A Few Useful Hits on Setting Options and Using EG

Individual preferences vary, but here are a few option and configuration settings that can greatly enhance the user's experience:

- Line numbers in editor: Program – Editor Options – General – check on "Show line numbers"
- Comments, in SAS programs, make all comments have a yellow background: Program – Editor Options – Appearance – Select Comments – Change Background to Solid Yellow
- Split the screen: View – Workspace Layout – Choose "Stacked" or "Side by Side". Side by Side is great on dual monitor systems.
- Display Variable Names (Columns) alphabetically (rather than position in the program data vector); Tools – Options – Data –Data General – check on Display columns in alphabetical order
- Storage of SAS Program (\*.sas) files – By default, any programming developed on a program node on the Enterprise Guide Process Flow is saved with the project, which means the code is available for viewing, editing etc. only when the project is open. If the code is to be viewed from outside the project, save the programs to individual SAS program files. For most programmers, this is the preferred method.

## 12. Summary

This paper discussed a number of reasons for traditional SAS programmers to convert to EG. Techniques demonstrated in this paper illustrate getting started with EG and using it efficiently with datasets on Unix. Remember --- the EG tutorial is good, and F4 is your favorite key if you ever get lost and need to return to the Process Flow!!!.

### 13. Additional Information

Muller, R.D., 2002. Optimizing SAS® Version 8 in a Windows Environment – from the User Interface to Automated Document Production, <http://www2.sas.com/proceedings/sugi27/p024-27.pdf>

Muller, R.D. and J.M. Horstman, 2010. Custom Google Searches, PDF Sticky Notes, and Other Tips for Organizing and Accessing SAS® Help Resources.  
<http://www.mwsug.org/proceedings/2010/resources/MWSUG-2010-120.pdf>

Horstman, J.M. and R.D. Muller, 2011. Dealing with Duplicates in Your Data.  
<http://www.mwsug.org/proceedings/2011/sas101/MWSUG-2011-S111.pdf>

Muller, R.D. and J.M. Horstman, 2011. Taming the Interactive SAS® Environment: Tips and Tricks for VIEWTABLE, The Enhanced Editor and More.  
<http://www.mwsug.org/proceedings/2011/tutorials/MWSUG-2011-TS07.pdf>

Schacherer, 2012. Take a Fresh Look at SAS® Enterprise Guide®: From point-and-click ad hocs to robust enterprise solutions. <http://support.sas.com/resources/papers/proceedings12/294-2012.pdf>

SAS Institute, SAS® 9.2 Companion for Windows Second Edition,  
<http://support.sas.com/documentation/cdl/en/hostwin/63285/PDF/default/hostwin.pdf>

SAS Institute, Enterprise Guide Documentation:

<https://support.sas.com/pubscat/booklist.jsp?attr=product&val=SAS+Enterprise+Guide>

<http://support.sas.com/software/products/guide/index.html#documentation>

SAS Integration Technologies

<http://support.sas.com/documentation/online/doc/inttech/>

### 13 Contact Information

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

Name: Roger D. Muller, Ph.D.  
Enterprise: First Phase Consulting, Inc.  
Address: 14475 Stephanie St.  
City, State ZIP: Carmel, IN 46033  
Work Phone: 317/846-5782  
E-mail: [rdmuller@hotmail.com](mailto:rdmuller@hotmail.com)

Name: Donald L. Penix, Jr. (D.J.)  
Enterprise: Pinnacle Solutions, Inc.

Address: 426 East New York Street  
City, State ZIP: Indianapolis, IN 46202  
Work Phone: 317-423-9143 x801  
E-mail: [dj.penix @ psiconsultants.com](mailto:dj.penix@psiconsultants.com)  
Web: <http://www.psiconsultants.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.