

## Code Generators: Friend or Foe

Janet Stuelpner, SAS Institute

### ABSTRACT

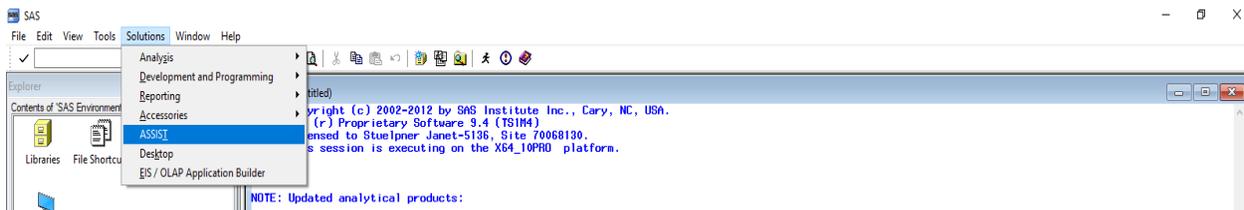
Good code generators are invaluable tools. Or are they? SAS® is constantly changing; adding new features and functions, adding new tools to the tool box while making the manipulation of data and the creation of tables more efficient. Many code generators exist in SAS. Some are embraced quickly, Others are not as people cling to the old methodologies. This presentation will show how code generators have grown over time and what is available now to make the task of programming easier, quicker and more efficient.

### INTRODUCTION

Have you been around programming in SAS for a very long time? When the display manager became available, there were features in there to generate code and capture that code. There are many SAS coders who have been writing code for decades. They know the DATA step or PROC SQL intimately and are very comfortable writing detailed code to transform data, write tables and produce graphs. However, over time, in the clinical research and development sector, we have become highly specialized and in doing so, have become less knowledgeable about every aspect of the language. So, what is a programmer to do? Embracing the impact that a code generator can have will enhance a programmer's ability to widen the scope of what they can do without looking up the syntax every 5 minutes. Let's look at some of the tools and solutions that generate code and how we can use that in our jobs.

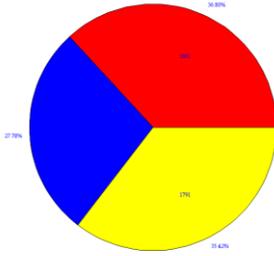
We will start with the earliest tools and work our way forward in time. When the display manager first came out, most people rolled up their sleeves and began to write code. There was a DATA step and there were many procedures. Several people wrote simple programs to provide listings to their data managers to review the data for a clinical trial. You could add a bunch of IF statements to subset the data and look for outliers or other anomalies. To create a listing that has a subset of data, you needed to know how to spell the names of all of the variables that you wanted to include in the listing. My solution was to run a PROC CONTENTS, print it off so that I would have all the names and along with the spelling of the name of the variables, (It was less complicated then because all of the variables were only 8 characters in length. It became more complicated in SAS version 8.2 when the names of dataset and variables could be 32 characters.) Let's explore how the code generators have evolved and some of the things that they offer.

### SAS/ASSIST®



The first code generator for simple tasks used with the display manager. It still exists today. It is a menu driven, task-oriented interface. This allows anyone to start programming without needing to learn the syntax. With SAS/ASSIST, you can perform tasks by using templates and menus. Since all the tasks are similar, once you learn one, it is easy to learn many. In the background, SAS/ASSIST software automatically generates SAS code with descriptive comments as it performs many of your tasks. You can save, edit and re-use the code that is generated. Some of

the tasks include: end-user reporting, presentation graphics, query and reporting, and decision support.



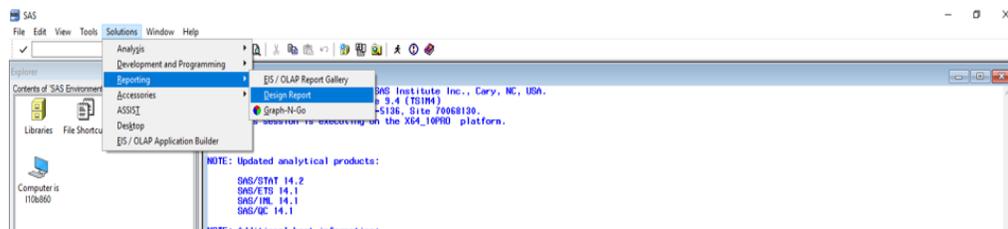
For the pie chart above, the code that was generated is:

```
View SAS Program
/*
 * Creating a simple pie chart using the table WORK.HEART
 * and showing the column Cholesterol...
 * Generated: 2006/02/17 17:51:04
 */
The GPTITLE statement allows you to have more control over the
final appearance of your output such as font, color, text
height and so on. The output device and destination is also
specified in the options statement.
-----
patterns react={axis, legend, pattern, symbol, title, fontsize} reactdate
{axis=0 upper=0 height= font= text= target= process= getmode=
};
patterns device=WIN chart=Pie
graphic interplot=join;
-----
/*
 * PATTERN statements allow you to define colors and patterns in
 * the chart. Any of the plot that you are creating. SAS/GRAPH uses
 * any pattern statements that you specify. If none are needed,
 * default PATTERN statements are used.
 */
pattern1 color=RED value=PSID.ID;
pattern2 color=BLUE value=PSID.ID;
pattern3 color=YELLOW value=PSID.ID;
pattern4 color=CYAN value=PSID.ID;
pattern5 color=GRAY value=PSID.ID;
pattern6 color=ORANGE value=PSID.ID;
pattern7 color=WHITE value=PSID.ID;
pattern8 color=GREEN value=PSID.ID;
pattern9 color=PURPLE value=PSID.ID;
pattern10 color=BLACK value=PSID.ID;
-----
/*
 * This section produces the actual pie chart and contains the
 * options that directly relate to the data.
 */
proc gchart data=WORK.HEART;
Pie Chol_Status /
relabeling

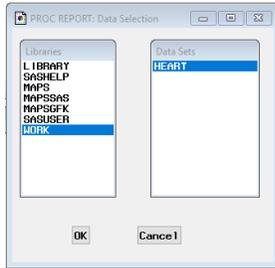
percent=OUTSIDE
fill=NAME
value=INSIDE
type=PIES;
run; quit;
```

### SAS REPORT WINDOW®

Wouldn't it be great to be able to start your PROC REPORT by opening all of the columns in a dataset, choosing which columns need to be in a report, dragging the columns so that they are in the order you need for the report and have the code for all of the DEFINE statements generated for you? This is something that the Report Window can do for you.



The Report Editor window generates a report from the selected dataset and displays it in the window. You can choose from all the LIBNAMES that you have available in your session. When you choose a LIBNAME, it will then list all the datasets that have been defined to that LIBNAME. Just click on the dataset for which you need a report and then click OK and you will have a new window pop up with the columns from that dataset.



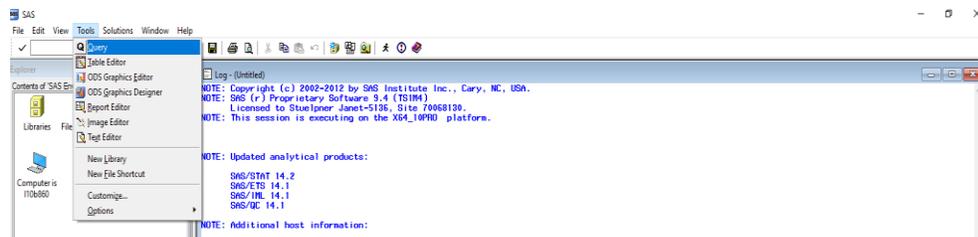
Once you have the dataset chosen, now you can start defining your columns, choosing formats, changing labels, etc. You are able to add data items, computed columns, statistics or a header line. There is quite a bit that you can do. Once you are satisfied with the report, you can save your code. Then you can open the code into the program editor and make more changes if you like. But the great thing here is that all your DEFINE statements have been started for you.

```

PROC REPORT DATA=WORK.HEART LS=98 PS=55 SPLIT="/" CENTER ;
COLUMN Status Sex AgeAtStart Weight Systolic Diastolic BP_Status Smoking_Status Chol_Status
Weight_Status Chol_Status_Al;

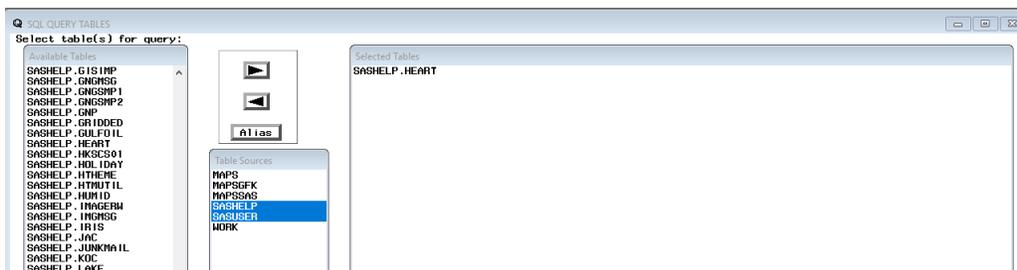
DEFINE Status / DISPLAY FORMAT= $5. WIDTH=6 SPACING=2 LEFT "Status" ;
DEFINE Sex / DISPLAY FORMAT= $6. WIDTH=6 SPACING=2 LEFT "Sex" ;
DEFINE AgeAtStart / SUM FORMAT= best9. WIDTH=5 SPACING=2 RIGHT "Age at Start" ;
DEFINE Weight / SUM FORMAT= BEST9. WIDTH=9 SPACING=2 RIGHT "Weight" ;
DEFINE Systolic / SUM FORMAT= BEST9. WIDTH=9 SPACING=2 RIGHT "Systolic" ;
DEFINE Diastolic / SUM FORMAT= BEST9. WIDTH=9 SPACING=2 RIGHT "Diastolic" ;
DEFINE BP_Status / DISPLAY FORMAT= $7. WIDTH=8 SPACING=2 LEFT "Blood Pressure Status" ;
DEFINE Smoking_Status / DISPLAY FORMAT= $17. WIDTH=17 SPACING=2 LEFT "Smoking Status" ;
DEFINE Chol_Status / DISPLAY FORMAT= $10. WIDTH=11 SPACING=2 LEFT "Cholesterol Status" ;
DEFINE Weight_Status / DISPLAY FORMAT= $11. WIDTH=11 SPACING=2 LEFT "Weight Status" ;
DEFINE Al / DISPLAY FORMAT= $10. WIDTH=11 SPACING=2 LEFT "Cholesterol Status" ;
RUN;
    
```

## SAS QUERY WINDOW®

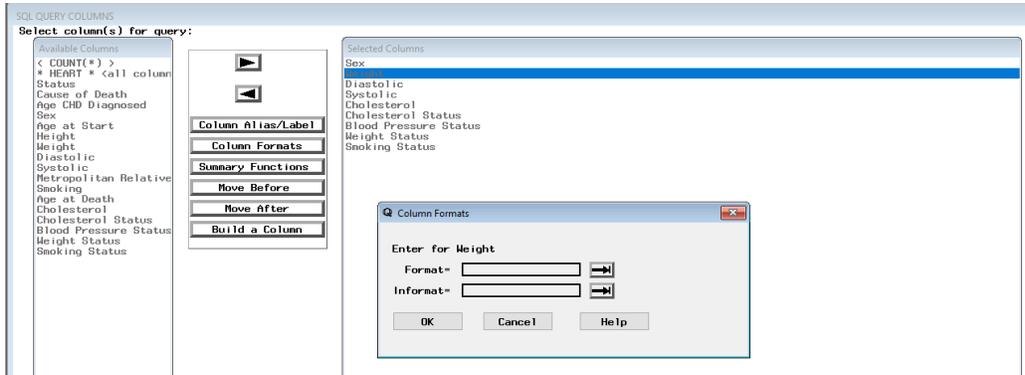


The query window creates SQL code. It is an interactive application that makes it easier to build and process an SQL query. You can access data that is stored in a SAS data set, SAS data view, VSAM files or database management systems that are supported by SAS/ACCESS software. Basically, you can access any data that SAS can access.

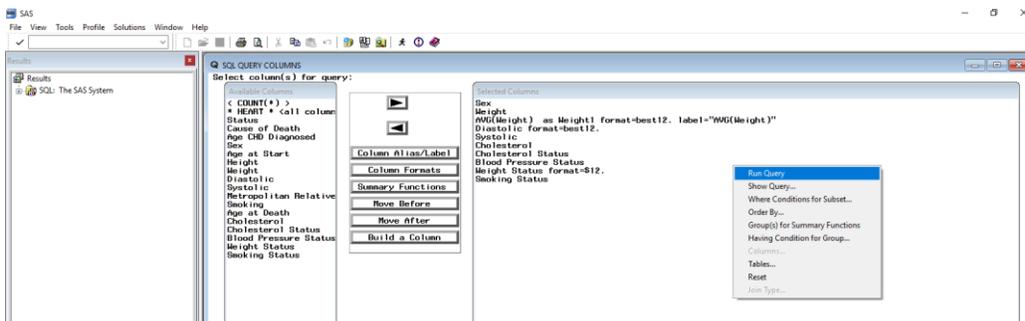
To build a query, you need to select the actions that will be performed on the data. You can customize a report by applying SAS formats and labels to individual columns, build new columns (expressions), set WHERE conditions to subset your data, or request distinct values. You can summarize, group, and order the data. From the actions that you select, the SQL Query Window builds an SQL query.



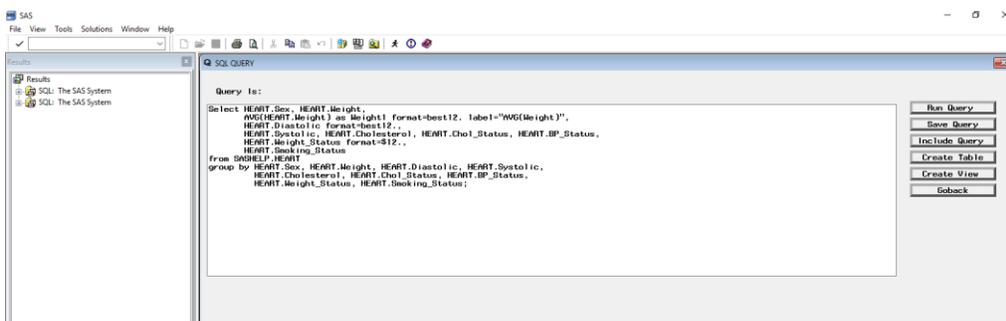
The SQL query can be composed of many components. However, only a table and its columns are required to build a complete query. Once you choose your table, a window pops up with all of the columns in that table and you can start to build your query. You have the opportunity to choose columns, change the labels or give a column an alias, add or change formats, create a new column or move your columns.



Once you have added all the pieces to your query, you can choose to run the query, view the code, put in WHERE conditions to subset the data and so much more.



Here is an example of a query that was built using the SASHELP.HEART dataset. You can save this code to use later. If you save the code to an external file, you can edit the code or use pieces of it to use in another program.

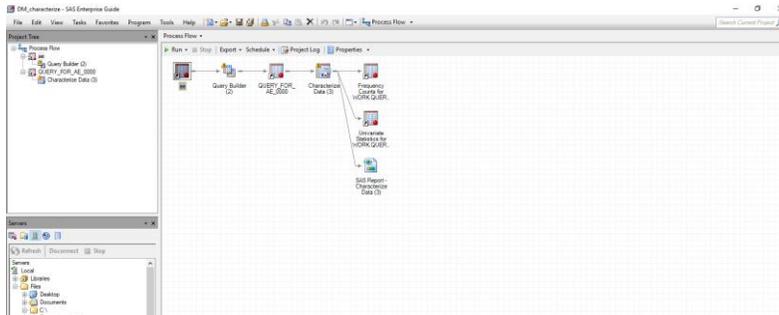


## SAS ENTERPRISE GUIDE®

SAS Enterprise Guide is a point-and-click, menu- and wizard-driven tool that empowers users to analyze data and publish their results. Interactive dialog boxes guide users through analytical and reporting tasks that range from simple to complex. Behind the scenes, SAS Enterprise Guide

builds the code for you. Either through the wizards which guide you step by step to choose options, or through the menus where you can choose your options in a less guided way. With SAS Enterprise Guide, you can see all the options for a task and choose what you need without having to know all the syntax. You can take advantage of the statistical procedures as well as graphics quickly and easily. You don't need to write any code.

Once you have chosen the options, titles, labels, etc., you can save the code and make changes to it if you need to do that. Below is an example of a project flow that uses an Adverse Event SDTM data set to characterize the data. As you can see, there is a graphical user interface.



Each time you drag a data set onto the palette to create your project, you can have the data set open so that you can review the data. In the example that I have created here, through the Query Builder, I have subset the data to just a few columns. I then dragged over the Characterize Data task that can be found in the Describe menu to create some tables and bar charts to get a better view of the types of variables and the values of the variables in my data.

Characterize Data (3)

	AEDECOD	AEBODSYS	AEBDSYCD	AESEV	AESER	AEREL
1	HYPERTENS	Vasc		moderate	N	Not Related
2	MASS EFFECT	Nerv		moderate	N	Not Related
3	VASOSPASM	Vasc		moderate	N	Not Related
4	ANEMIA	Blood		moderate	N	Unlikely Related
5	COAG DIS	Blood		severe	Y	Not Related
6	EDEMA BRAIN	Nerv		moderate	N	Not Related
7	EDEMA LUNG	Resp		moderate	N	Unlikely Related
8	EDEMA LUNG	Resp		moderate	N	Not Related
9	EDGMA PERIP	Vasc		moderate	N	Not Related
10	EFFUS PERIC	Card		mild	N	Not Related
11	FEVER	Genr		moderate	N	Not Related
12	FIBRILLAT ATR	Card		moderate	N	Not Related
13	HEART FAIL R	Card		moderate	N	Possibly Relat.
14	HEM GI	Gastro		mild	N	Not Related
15	HEM SUBARA...	Nerv		severe	Y	Not Related

You can look at the code that was generated. As you can see, it is color coded just like the Enhanced Program Editor. In this section of code, the comments are colored in green, the actual code is black, and the titles and footnotes are in magenta. This should be very familiar if you have been using the Program Editor in SAS Display Manager.

```

Code generated by SAS Task
-----
Generated on: Tuesday, April 30, 2019 at 5:03:50 PM
By task: Characterize Data (3)
Input Data: Local:WORK_QUERY_FOR_AE_0000
Server: Local
-----
%_mg_conditional_dropds(WORK.FREQCharFrequencyForQUERY_FOR_AE,
WORK.UNICharUnivariateForQUERY_FOR_AE,
WORK.STATTempTableAccumFreq,
WORK.TTATempTableUnivarUni,
WORK.TCONTempTableContests,
WORK.TFFTempTableFrequencies2,
WORK.TPONTempTableUnivariate2,
WORK.TSTTempTableUnivariate1,
WORK.TFFRTempFormatFreq);
%MACRO REPORTS;
%IF &charFreqFlag = 1 %THEN
%DO;
OPTIONS MISSING= ' ' PAGES=1;
TITLE;
TITLE1 "Summary of Categorical Variables for &L1B.&LMS.";
TITLE2 "Limited to the &CATOBS. Most Frequent Distinct Values per Variable";
FOOTNOTE;
FOOTNOTE1 "Generated by the SAS System (&SASSEVERNAME. &SYSJOBID) on &STRM(&QTSFUNC(&DATE1, &SLATE20.)) at &STRM(&SYSFUNC(TIME()), TIME)";
PROC PRINT DATA=WORK.STATTempTableAccumFreq LABEL;
BY Variable Label;
ID Variable Label;
VAR Value Count Percent;
FORMAT Label;
;
RUN;
    
```

You can also look at the results of the run of the program on the Results tab. The first set of tables are frequency counts of several variables. Then there is a horizontal bar chart of one of the categorical variables in this data set.

Character Data (1) -

Input Data | Code | Log | Output Data (0) | Results | Properties

Refresh | Modify Task | Export | Send To | Create | Publish | Properties

Summary of Categorical Variables for WORK\_QUERY\_FOR\_AE\_0000  
Limited to the 15 Most Frequent Distinct Values per Variable

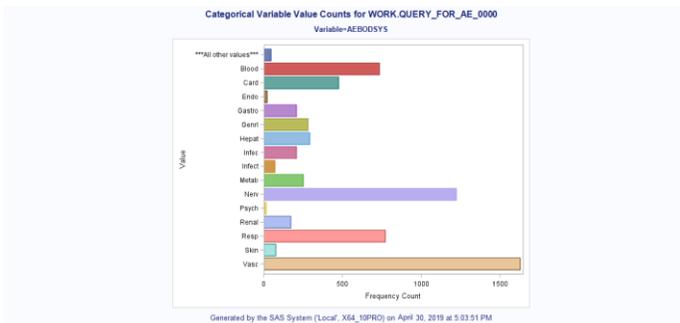
Variable	Label	Value	Frequency Count	Percent of Total Frequency
AEBODSYS	Body System or Organ Class	Vasc	1628	25.1584
		Nerv	1222	18.9843
		Resp	772	11.9301
		Blood	734	11.3429
		Card	475	7.3684
		Muscl	292	4.5124
		Genit	280	4.3270
		Metab	251	3.8798
		Gastro	208	3.2143
		Infect	208	3.2143
		Renal	172	2.6580
		Skin	76	1.1745
		Infect	70	1.0817
		Endo	21	0.3245
		Psych	16	0.2473
***All other values***		46	0.7109	

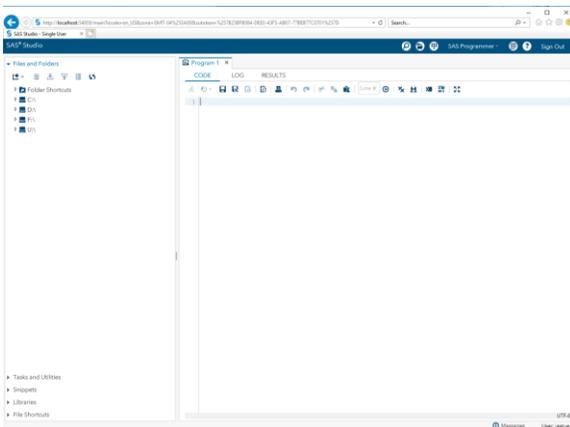
Variable	Label	Value	Frequency Count	Percent of Total Frequency
AEDECOD	Dictionary-Derived Term	VASOSPASM	693	10.7393
		ANEMIA	376	5.8323
		HYPOTENS	271	4.1879
		HYPERTENS(S)	268	4.1416
		HYDROCEPHALUS	263	4.0643
		FEVER	255	3.9407
		EDEMA LUNG	249	3.8479
		LIVER FUNC ABNORM	238	3.6779
		HYPERLYCEM	216	3.3225
		PNEUMONIA	200	3.0907
		INTRACRAN HYPERTENS	190	2.9382
		EDEMA PERIPH	187	2.8988
		EDEMA BRAIN	184	2.8435
		ATELECTASIS	152	2.3380
		PHLEB (ST)	158	2.4417
***All other values***		2572	40.3647	

Variable	Label	Value	Frequency Count	Percent of Total Frequency
AREL	Causality	Not Related	5021	77.9923
		Unlikely Related	799	12.3474
		Possibly Related	496	7.6650
		Probably Related	150	2.3653



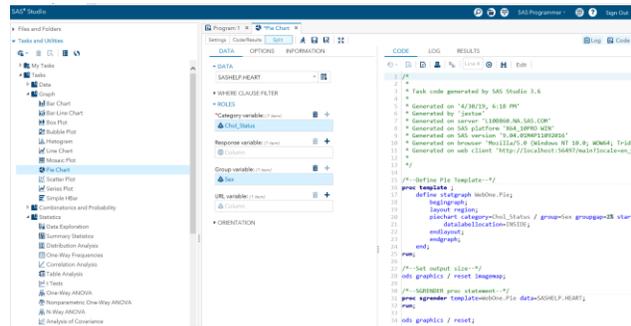
## SAS STUDIO®



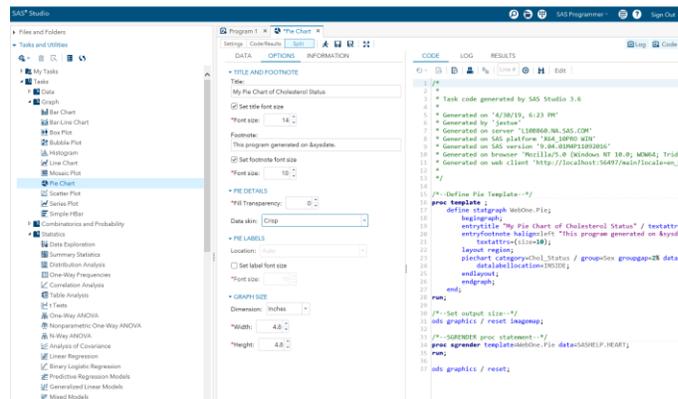
SAS Studio is a developmental web application for SAS that you access through your web browser. With SAS Studio, you can access your data files, libraries, and existing programs, and you can write new programs. You can also use the predefined tasks in SAS Studio to generate SAS code for you. It is very similar to SAS Enterprise Guide. The big difference is that there isn't a Graphical User Interface. You need to have a web browser to interface with the application. You can write your own programs where some assistance is offered (you can turn this off in the

options). Or you can use the tasks that are the same as in SAS Enterprise Guide, however, they are accessed and used in a slightly difference way.

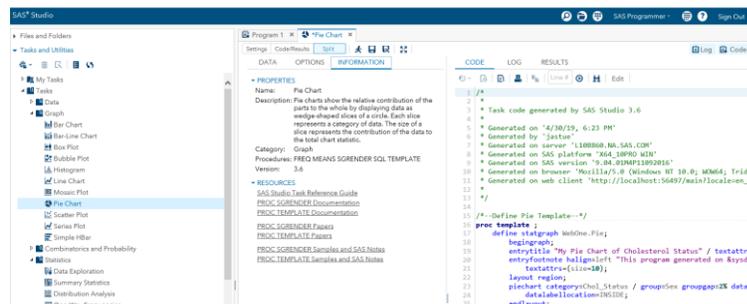
In the example below, I have dragged over the Pie Chart Task from the drop-down Graph list of tasks. The goal is to create a Pie Chart from the SASHELP.HEART data set that I have chosen from my data. The variable that I am using is the CHOL\_STATUS variable and I am grouping the data by SEX. You can see o the right hand side, that my program is being written as I am choosing my options.



I am now switching over to the Options tab so that I can choose the features of the graph from the list. I can change the font size, add in titles and footnotes as well as decide how big my graph can be. On the right side, my program continues to grow based on the options that were chosen.



If I need help along the way, there is information (properties and syntax) on the Information tab. There are also links to web sites that can provide in-depth information about the graph that you are creating. All this time, for each change that you make, the code is generated.



## SUMMARY

As you can see, there are many opportunities within SAS software where you can click, drag and drop tasks, columns, etc. to create a program that you can save. In all the options that I have shown, the common theme is the generation of code for you.

## CONCLUSION

We come back to the original premise of this topic, is a code generator a friend or a foe. There are those that believe that the only way to write a program is the code line after line. These are usually people who learned SAS just as the Display Manager was released. There have been many new features that have been added since that time. Can you imagine a time when there was no PROC SQL or even the fact that the WHERE statement didn't exist? But now, with options and tasks, the SAS program is generated completely or with a bit of help from the options that you choose. In any case, the guidance and help make programming easier. The programmer can get a whole bunch more done in less time because these aides exist. I embrace the new features of the various programming interfaces that make my job easier and make me more efficient.

## REFERENCES

SAS/Assist for Beginners, Sally Muller, Antonio Barrios, University of North Carolina at Chapel Hill, [http://nue.okstate.edu/Research\\_Methods/PAPER56.PDF](http://nue.okstate.edu/Research_Methods/PAPER56.PDF)

## ACKNOWLEDGMENTS

This paper would not have been written if it hadn't been for the support given to me by my husband, Robert Stuelpner. Bob diligently read this paper to correct obvious errors and keep me on the right track. His criticisms were constructive and his support never ending.

## RECOMMENDED READING

- *SAS For Dummies®*
- *The Little SAS Enterprise Guide Book*
- *SAS Programming for Enterprise Guide Users, Second Edition*

## CONTACT INFORMATION

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

Janet Stuelpner  
SAS Institute  
919-531-9758  
[Janet.Stuelpner@sas.com](mailto:Janet.Stuelpner@sas.com)  
[www.sas.com](http://www.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.