PhUSE 2012

Paper PP15

Metaprogramming in SAS

Arvind Chaudhari, Cytel Statistical Software & Services Pvt. Ltd., Pune, India

ABSTRACT

In the Clinical Trial reporting field we need to write SAS® programs for many activities such as Data Extraction, Data Creation, Report Generation and so on. In many situations, both SAS program file and the output of the program are required for submission. In SAS we can do Metaprogramming using SAS macro. If a SAS macro is already present for these kinds of task then we can generate submission ready SAS code file by using some options with the macro. This paper illustrates how SAS macro options MPRINT and MFILE can be used to generate SAS code file. In addition to these options, this paper explains various techniques required for generation of efficient SAS code. We will show an illustration of SAS graph code generator utility developed using SAS and EXCEL. This paper is intended for experienced SAS programmers who have a basic understanding of the SAS macro language.

INTRODUCTION

In the clinical trial reporting field, most of the cases we use generic macro to perform SAS programming activities. We can use these macros not only for generating output (for example tables, listings, figures) but also for generating its SAS code file. Instead of submitting SAS macro code it is good to submit base SAS code. Submission of generated code is good for macro confidentiality. This paper will talk about how SAS macro language can be used in combination with options MPRINT and MFILE to create a SAS source code generator utility. Some of the benefits of this method are listed as follows.

1. Increase in productivity
2. Efficient development
3. SOP compliant, submission quality SAS code
4. Person independent
5. Require validation only once
6. Reusable for similar tasks
7. Can be used in debugging SAS macro

OPTIONS MPRINT MFILE

The MPRINT option displays the text generated by macro execution. These generated SAS statements are useful for debugging macros. Here we will use MPRINT option for different purpose. A great feature of MPRINT is that in conjunction with MFILE option we can direct executed program statements to a stand-alone output file without MPRINT prefix printed at the beginning of each line. Below mentioned code will create test_code.sas file, which contains resolved statement of test_macro macro.

```
options mfile mprint;
%inc "D:\GraphGenie\test_macro.sas" ;
Filename mprint "E:\GraphGenie\test_code.sas";
data _null_;
   file mprint;
run;
%test_macro ;
options nomfile nomprint;
```

Since the code generating macro has been validated, the generated code using above method will not require rigorous validation. In some cases, minor modifications may be required to the generated code. The generated code from options MPRINT and MFILE is not in standard format. Generated code requires modification to convert it in to the exact required format such as proper indenting. Thus apart from the standard options MPRINT and MFILE we need to use other techniques to convert generated code into submission ready (SOP/GPP compliant) SAS code. The next section of the paper will explain about the points which are required to integrate in your SAS macro or MPRINT created code to generate submission ready code

COMMENTS

Everyone understands the importance of inserting appropriate comments in the code for understanding the logic and purpose of the code. Code Generated using MPRINT and MFILE does not contain comments.
While writing macro you can do two types of commenting.

Type 1
* …comments… ;

Type 2
/*……..*/

There is a limitation about what type of comments are included in the generated code by MPRINT option.
Only comments of Type 1 are included in the generated code by MPRINT option. Thus while writing your macro you should use Type 1 comments, in case you want them to be included in the generated code. Comments which are written for understanding macro program should not use Type 1 comments as these are not desired to be present in the generated stand-alone program. Documentation blocks need to use Type 1 comments to neatly pass them through to the generated stand-alone program.

INDENTATION

Generated code using MPRINT and MFILE is not indented. There is no blank space between lines. We can write separate macro specifically for indention. This macro can be written in SAS or other languages also, like VBA, VB and C++.
For example we have written SAS macro for indenting the generated code. Parameters to this macro are passed using a CSV file. No additional license is required to read a CSV file using SAS, whereas in case of Excel, we need extra license. This is an advantage of CSV over Excel. We can handle spacing for indentation using text mentioned in csv file. As shown in table below we can give instructions to indent macro for indenting. As per table below text mentioned for Level 1 should be start from line position 1, for Level 2 one space can be added before the text and for Level 3 text and remaining text two spaces can be added.

Parameter table:

Level 1	Level 2	Level 3
LIBNAME	SET	RETAIN
GOPTIONS	MERGE	
DATA	VAR	
PROC	ID	
QUIT	FORMAT	
RUN	ATTRIB	
*	INFORMAT	
/*	SELECT	
OPTIONS	INPUT	
ODS	CREATE	
TITLE	BY	
FOOTNOTE	TABLES	
FILENAME	LENGTH	
	KEEP	
	DROP	
	LABEL	

Sample macro call can be written as follows:

```
%Indent_program (
  programpath = &FolderPath,
  programname = &Filename ,
  linelen = 90,
  space=2,
  author = ARVIND CHAUDHARI
);
```

HEADER

Generated code using MPRINT and MFILE does not contain header. Header is important in program. This will give you the information about the Project name for which program is written, study name, study number, protocol number, purpose of program, date when created, author of the program, copyright information, revision history, program algorithm and so on.
Code below can be included in indentation macro to insert the header in generated code.

```
/*Priting header in the output sas file*/
data _null_;
file "&programpath.\&programname1..sas" ;
put "/*-------------------------------------------------------------------------------------------";
put "PROGRAM:                                                                                     ";
put "SAS VERSION: &sysver.                                                                        ";
put "DESCRIPTION:                                                                                 ";
put "AUTHOR: &author.                                                                             ";
put "DATE COMPLETED: &sysdate9.                                                                   ";
put "REVISIONS:                                                                                   ";
put "PROGRAM INPUT:                                                                               ";
put "PROGRAM OUTPUT:                                                                              ";
put "PROGRAM LOG:                                                                                 ";
put "EXTERNAL CODE CALLED:                                                                        ";
put "                                                                                             ";
put "PROGRAM ALGORITHM:                                                                           ";
put "                          1.                                                                 ";
put "                          2.                                                                 ";
put "                          3.                                                                 ";
put "                          4.                                                                 ";
put "-------------------------------------------------------------------------------------------*/";
run;
```

SAS STATEMENTS

There are many SAS statements we need to use while writing macro like TITLE, FOOTNOTE, LIBNAME, FILENAME, ODS, OPTIONS GOPTIONS statements. All these statement will resolve as it is.
In the context of MPRINT and MFILE option no constraint on use of SAS statements.

DRIVER PROGRAM

This is the final program where we need to write appropriate macro call according to the requirement of the generated code. In this program we need to specify all required options, name and location for saving the generated SAS code file, Indentation macro call, code for checking the existence of file and so on.
Generated code using MPRINT and MFILE does not contain macro call since it contains all resolved macro statements. Some times it is necessary to include macro call which creates required initial set up like creation of all libraries. For example code below can be used to include macro “setup” at start of the program. You can place this type of code inside your main macro code at the start .

```
data _null_ ;
file mprint ;
put "%inc 'E:\setup.sas' ; " ;
put "%setup (Path=E:\ ) ;" ;
run;
```

SAMPLE MACRO CALL CAN BE AS FOLLOWS.

```
dm 'log;clear;';

* Specify name for SAS code file;
%let Filename = code;

* Folder location of all required and output files ;
%let FolderPath = E:\GraphGenie ;
* Including all necessary files ;
%inc "&FolderPath.\Report_Macro.sas";
%inc "&FolderPath.\Indent_program.sas";

options mprint mfile mlogic symbolgen nofmterr noquotelenmax;
Filename mprint "&FolderPath.\&Filename..sas" ;

/* Deleting SAS code file if already exists*/
data _null_;
  rc=fdelete('mprint');
run;
```

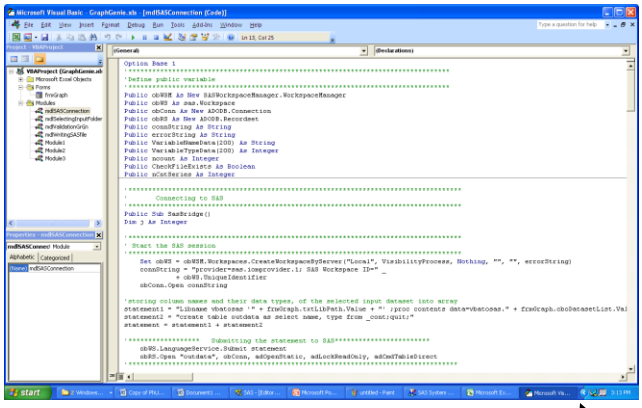
```
%Report_Macro ( Parameter1= X,
                Parameter2= Y,
                Parameter3= Z,
                           .
                           .
                           .
                           ) ;
```

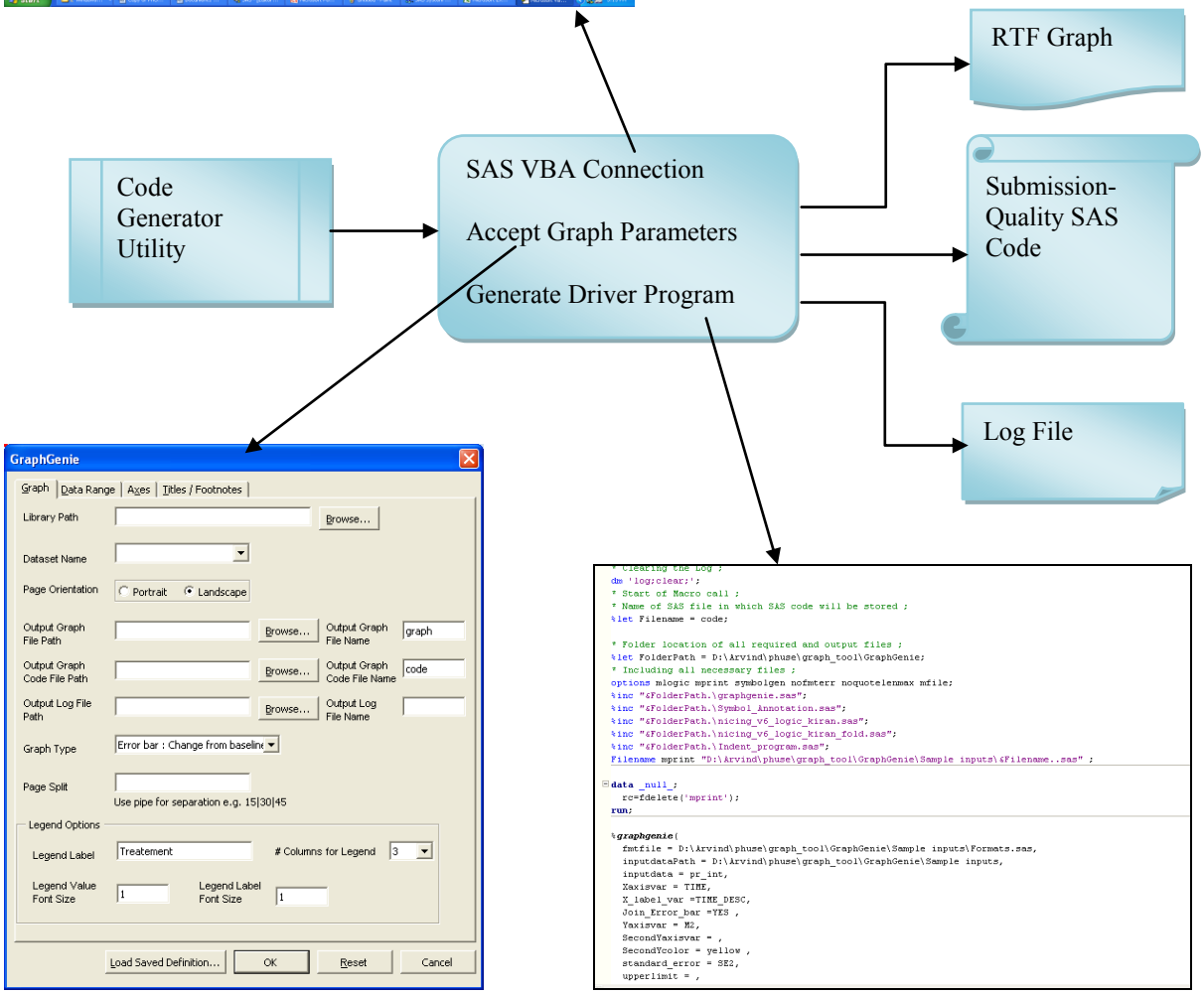
```
options nomfile nomprint;

/*macro call for indentation of generated code*/
%indentcode (
  programpath = &FolderPath ,
  programname = &Filename ,
  programname1 = &Filename ,
  filepath = &FolderPath.\keywords.csv,
  space = 2 ,
  linelen = 90
);

/*To generate log file of generated code*/
proc printto log="&OutFileLocation.\&Filename..log" new ;
run;
%inc mprint;
proc printto;
run;
```

ILLUSTRATION OF SAS GRAPH CODE GENERATOR UTILITY DEVELOPED IN SAS AND EXCEL

This utility generates SAS graph and its SOP compliant, submission quality SAS code.
The user interface is developed in Excel-VBA. The logic of generating the SAS graph output and the code resides is a SAS program. User enters all required input parameters through Excel user interface. Excel interface will create driver program and no need to open SAS for creation of graph and SAS code. User does not require having detail prior knowledge of SAS to use this utility.
UI developed in EXCEL-VBA is used to input all required parameters.
For example parameters can be location of input SAS dataset, name of the input dataset, name and location for all outputs, orientation and type of the graph, page splitting and legend options for the graph. After entering all required parameters utility will create driver program sas file. Using connection between VBA and SAS we can run the SAS driver program file. This driver program will be used to generate submission ready SOP compliant SAS code.

FLOWCHART OF THE UTILITY

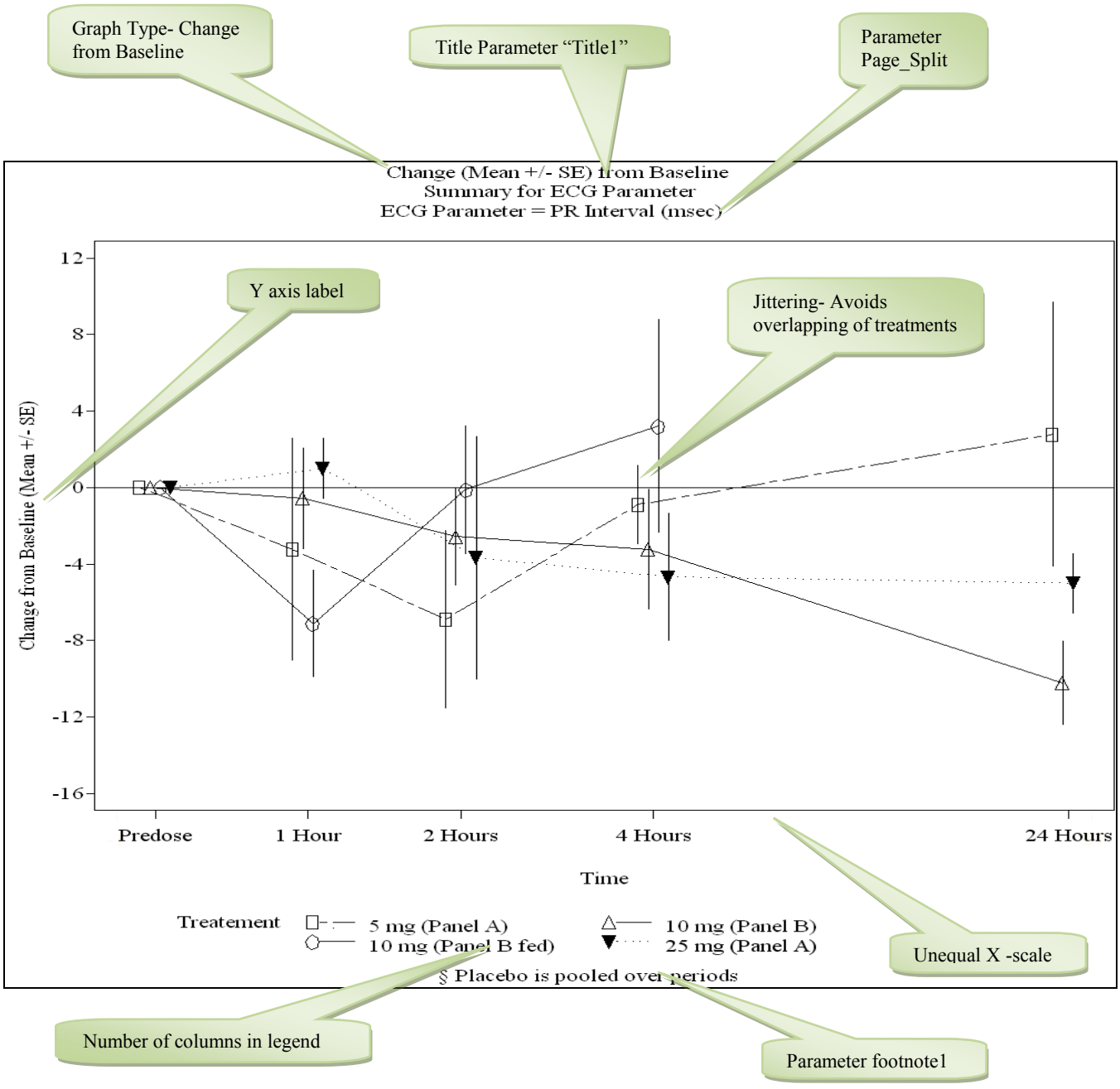




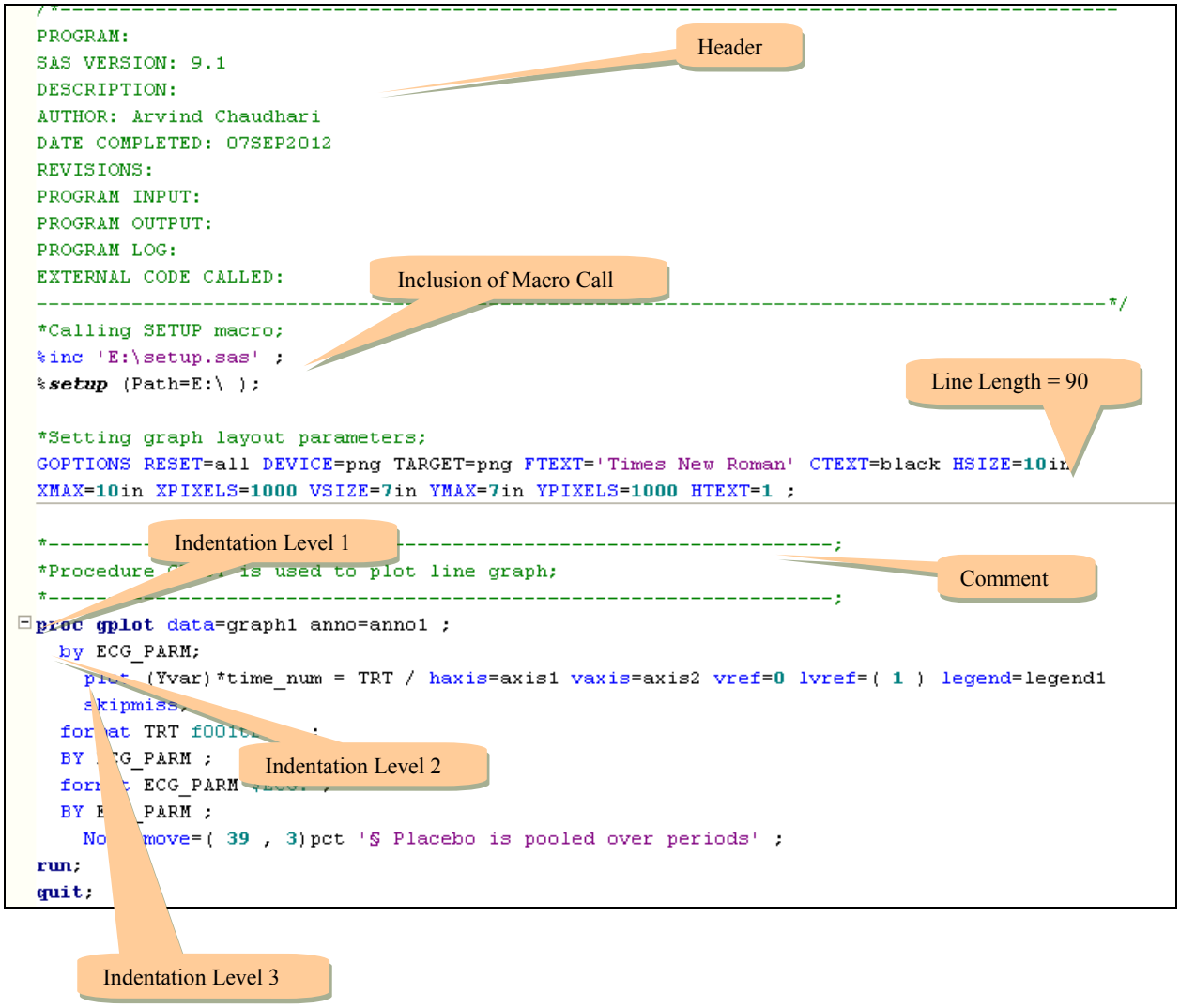

```
* Clearing the Log ;
dm 'log;clear;';
* Start of Macro call ;
* Name of SAS file in which SAS code will be stored ;
%let Filename = code;

* Folder location of all required and output files ;
%let FolderPath = D:\Arvind\phuse\graph_tool\GraphGenie;
* Including all necessary files ;
options mlogic mprint symbolgen nofmterr noquotelenmax mfile;
%inc "&FolderPath.\graphgenie.sas";
%inc "&FolderPath.\Symbol_Annotation.sas";
%inc "&FolderPath.\nicing_v6_logic_kiran.sas";
%inc "&FolderPath.\nicing_v6_logic_kiran_fold.sas";
%inc "&FolderPath.\Indent_program.sas";
Filename mprint "D:\Arvind\phuse\graph_tool\GraphGenie\Sample inputs\&Filename..sas" ;

data _null_;
  rc=fdelete('mprint');
run;

%graphgenie(
  fmtfile = D:\Arvind\phuse\graph_tool\GraphGenie\Sample inputs\Formats.sas,
  inputdataPath = D:\Arvind\phuse\graph_tool\GraphGenie\Sample inputs,
  inputdata = pr_int,
  Xaxisvar = TIME,
  X_label_var =TIME_DESC,
  Join_Error_bar =YES ,
  Yaxisvar = M2,
  SecondYaxisvar = ,
  SecondYcolor = yellow ,
  standard_error = SE2,
  upperlimit = ,
```

SAMPLE OUTPUT CREATED FROM UTILITY

Various input parameters which are required for graph generation are explained in screenshot below



SAMPLE SAS CODE CREATED FROM UTILITY




```
/*-------------------------------------------------------------------------------------
PROGRAM:
SAS VERSION: 9.1
DESCRIPTION:
AUTHOR: Arvind Chaudhari
DATE COMPLETED: 07SEP2012
REVISIONS:
PROGRAM INPUT:
PROGRAM OUTPUT:
PROGRAM LOG:
EXTERNAL CODE CALLED:
-------------------------------------------------------------------------------------*/
*Calling SETUP macro;
%inc 'E:\setup.sas' ;
%setup (Path=E:\ );

*Setting graph layout parameters;
GOPTIONS RESET=all DEVICE=png TARGET=png FTEXT='Times New Roman' CTEXT=black HSIZE=10in
XMAX=10in XPIXELS=1000 VSIZE=7in YMAX=7in YPIXELS=1000 HTEXT=1 ;

*----------------------------------------------------------------------;
*Procedure GPLOT is used to plot line graph;
*----------------------------------------------------------------------;
proc gplot data=graph1 anno=anno1 ;
  by ECG_PARM;
    plot (Yvar)*time_num = TRT / haxis=axis1 vaxis=axis2 vref=0 lvref=( 1 ) legend=legend1
    skipmiss;
  format TRT f001t.  ;
  BY ECG_PARM ;
  format ECG_PARM $ECG. ;
  BY ECG_PARM ;
    Note move=( 39 , 3)pct '§ Placebo is pooled over periods' ;
run;
quit;
```

CONCLUSION

If we have generic SAS macro to perform SAS programming activity then using options MPRINT and MFILE with SAS macro, one can generate SAS code. In addition to these options, by using some of the techniques mentioned above for adding header, comments and indentation one can generate submission quality, SOP compliant SAS Code and the output without requiring an in-depth SAS coding knowledge. This will significantly reduce the development time needed for SAS code generation.

REFERENCES

http://support.sas.com/
http://www2.sas.com/proceedings/sugi27/p034-27.pdf

ACKNOWLEDGMENTS

The author would like to thank his colleagues at Cytel, Pune for their support and assistance and special thanks to Manjusha Gode for her suggestions and review of the paper.

CONTACT INFORMATION

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

Arvind Chaudhari
Cytel Statistical Software & Services Pvt. Ltd.
(a subsidiary of Cytel Inc.)
6th Floor, Lohia-Jain IT Park – A Wing,
Survey #150, Paud Road, Kothrud,
Pune 411 038, India
Work Phone: +91 (20) 6709-0141
Fax: +91 (20) 6709-0120
Email: Arvind.Chaudhari@cytel.com
Web: www.cytel.com