

## **Mixing the Use of the Three Types of Comments for Ease of Use of SAS® Enterprise Guide® Code**

Mark Menzie, Assurant Health, Milwaukee, WI

### **ABSTRACT**

A frequent criticism of code is that it is not documented thoroughly enough. It is possible, however, for extensive comments to interfere with use, further development or troubleshooting of the code. This paper suggests thoughtful use of the three classes of comment statements to make use and maintenance of code easier:

1. Statement comment: \* [comment text] ;
2. Slash comment: /\* [comment text] \*/
3. Macro comment: %MACRO macroname; [comment text] %MEND  
macroname;

The advantage of mixing the three styles of comments is in the way that they overlay each other. Using the statement comment style where feasible allows easy overlay with the Slash comment style.

### **INTRODUCTION**

Three forms of comments, you say? The #3 above isn't really a comment at all. We'll see, however, that #3 Macro comments are the most powerful comments for use in troubleshooting our code.

It's nearly universally accepted that more and better comments in our code are a good thing. When someone new comes along to work with the code or even if we ourselves come back to work with old code that we have not touched recently the comments help explain what is going on and speed the process.

There are times, however, when comments can impede modifying and testing the code. The purpose of this paper is to propose ways that we can put comments in the code and still keep the code workable. The key element that this paper works with is an order of hierarchy that SAS® comments use.

#### **The Statement Comment**

The Statement comment is often overlooked. It is unfamiliar to many programmers coming from other languages to whom other forms of comments are more familiar. The statement comment begins with an asterisk and ends with a semi-colon.

\* [comment text] ;

The value of this comment is that it is a weak comment. The other two forms of comment easily overlay the statement comment. This means that we can freely sprinkle statement comments through our code without worrying about making it hard to block comment sections out later. The statement comment is also a very quick way to stop 1 statement from executing, we simply put the asterisk at the beginning of the statement and the statement is commented out. That is one keystroke vs at least 4 for other comment forms.

The disadvantage of the statement comment is that it needs to be a whole statement and it cannot be inserted in the middle of another statement as a result. Also there is no way to use the statement comment on multiple statements at once.

The statement comment is ideally suited for a note explaining the purpose of a section of code.

### **The Slash comment**

This is the most widely used and familiar form of comment. It's robust and comments exactly what you enclose in the `/* comment-text */`, unless there is another comment in the way.

```
/* [comment text] */
```

The slash comment is the way to go if you need to cut out only part of an existing statement, like temporarily turning off one of your options or taking a variable out of the input statement.

### **The Macro Comment**

Normally this is not thought of as a comment at all, but it will act as a comment if we don't call it. In the example below, the code between the `MACRO` and `MEND` statements is only executed when `macroname` is invoked.

```
%MACRO macroname; [comment text] %MEND macroname;
```

How is this useful? If the comment text included slash comments we could easily overlay all of them with a macro comment if we want to cut out a section of code for troubleshooting or development.

### **Illustrations – How the Comments Work Together**

We've got this code (what's shown is actually a small excerpt) and imagine that we want to run a larger program and cut out this section:

```
DATA CAPLIST;  
SET INPUT1;  
/*criteria1*/  
if [criteria1] then [result1];  
/*criteria2*/
```

```

if [criteria2] then [result2];
/*criteria3*/
if [criteria3] then [result3];
run;

```

We will need to do it like so using slash comments, 4 left hand slash comment marks and one right hand comment mark are needed to isolate the section:

```

/* DATA CAPLIST;           (1)
   SET INPUT1;
/*criteria1*/
/* if [criteria1] then [result1]; (2)
/*criteria2*/
/* if [criteria2] then [result2]; (3)
/*criteria3*/
/* if [criteria3] then [result3]; (4)
run;*/                       (5)

```

Below I have replaced the original slash comments with statement comments. Now, only one set of slash comments removes the whole section:

```

/* DATA CAPLIST;           (1)
   SET INPUT1;
   *criteria1;
   if [criteria1] then [result1];
   *criteria2;
   if [criteria2] then [result2];
   *criteria3;
   if [criteria3] then [result3];
run; */                       (2)

```

The MACRO comment removes the whole section from the executable stream, overlaying the slash style comments:

```

%MACRO MACROSTOP;       (1)
   DATA CAPLIST;
   SET INPUT1;
   /*criteria1*/
   if [criteria1] then [result1];
   /*criteria2*/
   if [criteria2] then [result2];
   /*criteria3*/
   if [criteria3] then [result3];
   run;
%MEND MACROSTOP;       (2)

```

Slash and statement comments work with SAS® Enterprise Guide® to turn commented sections of the program green. The MACRO comment text color does not change.

As an aside, you can also use the “RUN SELECTED” option in SAS® Enterprise Guide® to run only a portion of your program without changing the code.

### **Resources**

This paper is based on use of SAS® Enterprise Guide® on a Windows Personal Computer. Examples may work differently on another computing platform.

### **Contact Information**

Mark Menzie  
Assurant Health  
501 W Michigan St.  
Milwaukee WI 53203  
Work: 414-299-6156  
Cell: 414-517-8663  
mamenzie@hotmail.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 registered trademarks or trademarks of their respective companies.