

Commenting SAS Code

Presented by David Franklin, March 15, 2006

The three most common ways for commenting code are:

- Statement Comment, <*. . . .;>
- Macro Comment, <%*;>
- Block Comment, </* */>

Other ways are:

- An early syntax for a comment, the statement COMMENT. An example of this is

```
comment Set Height to SI;
if ^missing(height) then do;
  select(hght_u);
  when(2) hght_si=height*2.54;
  otherwise hght_si=height*1;
end;
end;
```

The Enhanced Editor does not recognize the COMMENT statement as a valid statement but the compiler does. This statement dates back to the early days of SAS and SAS have not made any mention of it in later releases of their documentation but it is valid.

- The use of a macro parameter. An example of this is

```
%macro p-value(test=, outvar=, qcprt=NO);
  /* Macro statements;
  if (%upcase(&qcprt)=YES) %then %do;
    proc print data=rslts;
      run;
  %end;
  /* More macro statements;
%mend;
```

- A Virtual Comment by creating a dummy macro around code, and that macro is not executed. An example of this is:

```
data alldata1;
  merge keyvars (in=a) commed (in=b);
  by pno cno patno;
  if a and b;
run;
%macro comment;
  proc print data=alldata1;
  run;
%mend comment;
data posdups;
  set alldata1;
  by pno cno patno conmedtx startda;
  if sum(first.startda,last.stopda)<2;
run;
```

When commenting code some of the most common mistakes are:

- A broken logic chain

```
%macro calcbsa(type=SI);
  %let type=%upcase(&type);
  * SI calculation, height (cm),
  weight (kg);
  %if (&type=SI) %then
    bsa=sqrt(ht*wt/3600);
  * Now for imperial calculation,
  height (in), weight (lbs);
  %else %if (&type=IM) %then
    bsa=sqrt(ht*wt/3131);
  * else error;
  %else
    %put WARN%STR(ING): <text>;
%mend calcbsa;
```

The above code will not work as the second comment breaks the logic chain since SAS treats the '*' as a statement although it is not executed. For this to work it would have been better to use the '%*' symbol instead, as the macro compiler removes any text starting with '%*' from compilation.

- The comment '%*'

The following example will show the problem:

```
  *%let tobe=ornottobe;
```

The semicolon will only close the '%let' but not the '*', hence for this statement to be truly correct the code should read

```
  *%let tobe=ornottobe;;
```

Quick Tip

The following example reads only the last observation in a SAS dataset:

```
data whatever;
  set something point=nobs nobs=nobs;
  stop;
run;
```
