The 'SKIP' Statement

Paul Grant, Private Healthcare Systems, Inc.

The Problem

One of the most irritating things about modify or enhancing an existing SAS® job is trying to run just the code you've modified, so you can test it without having to execute the entire job stream. If you're a craftsman with respect to your code, you can't use bracket-style comments (/* .... */) because you've already used these thoughout your program to document what each step is doing and bracket-style comments can't be nested. We all know what will happen when we try to run this code:

```
    /*
DATA SALARY;
    /* ASSIGN COMMISSION RATES */
    more SAS statements
RUN;
    */
```

What we really need is a SKIP statement. Like a comment statement, a SKIP statement would block lines of SAS code from being compiled and executed by the SAS supervisor. But unlike statement-style comments, the SKIP statement would 'hide' multiple SAS statements from the SAS supervisor. And unlike bracket-style comments, SKIP statements could be nested.

The Solution

You'll be pleased to know that this capability already exists in the SAS language - not as an undocumented feature - just an unimagined one. And as the following example shows, you can easily code one yourself.

```
    SAS statements you want to execute.....
%macro SKIP;
    SAS statements you want to skip.....
%mend SKIP;
    SAS statements you want to execute.....
```

This technique hides any SAS statements: lines within a DATA or PROC step, entire DATA and/or PROC steps and other macro definitions.

SKIP statements can be nested:

```
    SAS statements you want to execute.....
%macro SKIP;
    SAS statements you want to skip.....
%macro SKIP;
    More SAS statements you want to skip.....
%mend SKIP;
%mend SKIP;
    SAS statements you want to execute.....
```

SKIP statements can also be repeated:

```
    SAS statements you want to execute.....
%macro SKIP;
    SAS statements you want to skip.....
%mend SKIP;
    SAS statements you want to execute.....
%macro SKIP;
    More SAS statements you want to skip.....
%mend SKIP;
    SAS statements you want to execute.....
```

Note that you don't have to assign different names (e.g., 'SKIP1,' 'SKIP2,' etc) to each SKIP macro.

How It Works

This tip works by asking the SAS macro facility to store the portion of code that you don't want to execute as a SAS macro. Since the macro 'SKIP' is never called in the program, it is never passed to the SAS Supervisor to be compiled and executed.

A Final Note

Anyone who has inadvertently omitted an %MEND statement at the conclusion of one of their macro definitions knows how effective this trick is. In fact, that's how I 'discovered' it.

In addition to learning how to code a SKIP statement, I hope you have also seen the importance of learning from your mistakes even the 'little' ones like forgetting an %MEND statement and the role of imagination in putting what you've learned to good use.

Author

The author welcomes your comments and suggestions at:

Paul Grant
Private Healthcare Systems, Inc.
1100 Winter Street
Waltham, MA 02154
Tel: 617-895-7614
E-Mail: 0005733569@mcimail.com

SAS® is a registered trademark of SAS Institute, Inc., Cary, North Carolina