

## Paper 97-27

**A Macro to Help with Accurate Output Documentation**

Bruce Spotts, Eli Lilly and Company, Indianapolis, IN

**ABSTRACT:**

Accurately documenting the program and data library on output as well as the mode of operation is extremely important in order to manage programs and output. To assist with this task I have developed a macro that provides 4 macro variables to be used in titles or footnotes to accurately describe the program, data library, and the mode of operation for SAS programs running on MVS SAS® versions 6.09-8.1. This macro is designed to work on MVS but can be modified to run on PC SAS. The macro is provided and the benefits and limitations are discussed.

**PURPOSE:**

For efficient handling of SAS output it is extremely helpful to have the source of the output documented on the SAS output either in the title or in a footnote. In this case, the source consists of the SAS program and the data library being used by the program. It is also helpful (essential in some organizations) to document the status (TEST or PRODUCTION) of the program library and data library being used. All of this information can be included on the output in the form of a title or footnote and should be accurate.

To prevent inaccurate documentation, I imagined a macro that could capture this information into macro variables for use within a title or footnote. The macro I currently use provides 4 macro variables (&PGM, &DATNAME, &PGMMODE, &DATAMODE). The present macro works on MVS but will work on PC SAS with some modifications. The macro is included at the end of this paper along with an example of its use.

**SPECIFICATIONS:**

The value for the macro variable &PGM is provided by the macro parameter pgmname=. Unfortunately, the user still has to hard code the program name information in the program itself. The value for &DATNAME will contain up to 2 library paths that have been allocated with libname statements prior to calling the macro. If more than 2 libnames are allocated prior to calling the macro the macro variable &DATNAME will resolve with the values of the first 2 pathnames and the words '+ others'. This section of code can be easily modified to accept as many pathnames as you wish.

The values for the macro variables &PGMMODE and &DATAMODE are evaluated based on the path names of the program and the data libraries respectively. For our purposes, we classify reports as being in TEST or PRODUCTION. This status is reflected in the pathname of our program and data libraries. For example – if the program is in test then the pathname will contain the letters RMT as the first node of the MVS library. If the program is in production then the pathname will contain the letters RMP. If the pathname does not contain these letters then the value is set to UNKNOWN MODE. If *any* of the allocated libnames reflect a test library then the value for &DATMODE will display TEST MODE. Some modifications may be necessary to reflect your organization's library and program naming conventions.

Modifications will be needed for use on other platforms as well.

The following bit of code will provide you with the program name automatically if you run on PC SAS version 6.12 eliminating the need to provide the program name in the macro call.

```
%if (&sysver eq 6.12) %then %do;
proc sql;
  reset noprint;
  select xpath into :pgmname
    from dictionary.extfiles
    where fileref='TMP1';
quit;
%end;
```

**MACRO USE:**

In order to use this macro properly the macro must be included in your code via a %include, SAS macro autocall library, or have the macro coded in the program itself. Once the macro is ready for use, call the macro %MODE *after* you allocate the data libnames you will be using in your program. In MVS you will need to specify the program name with the parameter pgmname=. You may also specify the data library with the dataname= parameter but this is not required. After the macro call, you may use the 4 macro variables in your titles, footnotes, or any other place you wish to document your output.

There are some obvious limitations with this macro. First there is no 'easy' way to get SAS to recognize the program name and path you are running on MVS. Therefore, the user must still specify the program name in the macro call. However, you only need to specify the program name once in the program. After calling the macro you can reference the macro variable any time you wish to display the program name. I have provided a solution to this problem for PC SAS users with the bit of code preceding this section. The other obvious limitation is the naming conventions that are required for the macro to recognize the 'mode' you are running in. Coding for your specific naming conventions can solve this problem.

The benefits of using this macro have been a reduction in the 'corrections' we have needed due to inaccurate titles or footnotes. Additionally, You can accurately identify the data library that the program is pointing to without worrying whether or not the footnote was accurately hard coded. By identifying the mode of the program and data there is no question about the status of the report in hand or the status of the data that created the report. This approach also reduces the number of changes needed to move programs to different libraries. For example, when moving a program from a test to a production library there is no need to change titles or footnotes, as the macro variables will reflect these changes on the output for you. The following provides an example use of the macro and the naming conventions required for the macro as it is written.

**EXAMPLE PROGRAM:**

```

libname in 'RMT.STUDY.SASDATA' DISP=SHR;
libname in2 'RMP.STUDY.SASDATA' DISP=SHR;

%mode (pgmname=RMT.STUDY.SASPGM(EXAMPLE));

title1 "&pgmmode - &datamode";
title3 "pgm: &pgm";
title4 "data: &datname";

data _null_;
file print;
put // 'THIS IS AN EXAMPLE OUTPUT'//
      'TO SHOW THE VALUES OF THE 4 MACRO
VARIABLES'//
      'CREATED BY %MODE'//;
run;

```

**EXAMPLE OUTPUT:**

```

TEST MODE - TEST DATA

pgm: RMT.STUDY.SASPGM(EXAMPLE)
data: RMT.STUDY.SASDATA, RMP.STUDY.SASDATA

THIS IS AN EXAMPLE OF OUTPUT
TO SHOW THE VALUES OF THE 4 MACRO VARIABLES
CREATED BY %MODE

```

**%MODE MACRO**

```

*****
PROGRAM NAME : %MODE
PURPOSE      : SAS Macro to determine the
program mode (test/production)based on the
library and data mode of the libname
specified in the program. Provides macro
variables with the mode and the names of the
program and data libraries being used

MACRO CALL
  %mode(dataname= , pgmname= )
  dataname(optional) = the library of the SAS
  datasets used in the program
  pgmname(required) = the library and member
  name of the program
*****;

%macro mode(dataname=, pgmname= );

%global datamode data;

  %if &dataname= %then %do;

    proc sort data=sashelp.vslib out=_mem;
      by libname;

    data _mem; set _mem(keep=libname path);
      by libname;
      if libname not
in('SASHELP','USER','SASUSER','WORK','MAPS');
      if first.libname;
      seq+1;
      call symput('nlib',seq);
    run;

```

```

%put &nlib;

  %do i=1 %to &nlib;
    data _null_;
      set _mem;
      where seq=&i;
      call symput("dat&i ",compress(upcase(path)));
    run;

%if (&datamode ne TEST DATA) %then %do;

  *** FOR MVS ****;
  %if %substr(&dat1,1,3)=RMP %then %let
datamode=PRODUCTION DATA;
  %else %if %substr(&dat1,1,3)=RMT %then %let
datamode=TEST DATA;
  %else %let datamode=UNKNOWN DATA MODE;
  *****;
%end; %end;

  %if &nlib=1 %then %let datname=&dat1;
  %else %if &nlib=2 %then %let datname=&dat1,
&dat2;
  %else %if &nlib ge 3 %then %let datname=&dat1,
&dat2, + others;

%global pgm pgmmode;
%let pgm=%upcase(&pgmname);

  *** FOR MVS ***;
  %if %substr(&pgm,1,3)=RMP %then %let
pgmmode=PRODUCTION MODE;
  %else %if %substr(&pgm,1,3)=RMT %then %let
pgmmode=TEST MODE;
  %else %let pgmmode=UNKNOWN MODE;

%end;
%mend mode;

```

**CONTACT INFORMATION**

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

Bruce Spotts  
Eli Lilly and Company  
Indianapolis IN 46285  
(317) 277-0710  
bspotts@lilly.com