

Using Dictionary Tables to Create Meaningful Titles

Frank C. Dilorio, ASG, Inc., Cary, NC

ABSTRACT

"The SAS System" is not what anyone would consider a descriptive default title for SAS output. This paper briefly describes a macro that creates more meaningful titles. It uses information from the dictionary tables and, optionally, user-supplied titles. The paper presents the macro source as well as examples of its usage.

THE PROBLEM, AND A SOLUTION

How many times have you run a SAS procedure without a title? The output may be correct and useful, but the default title, "The SAS System," is less than informative. The macro PRT, shown in Exhibit 1, creates more meaningful titles. It retrieves select variables from the dictionary tables (see "Data About Data: An Introduction to Dictionary Tables," elsewhere in this volume, for a discussion of the tables). This information is used to create two title lines displaying, among other items, the dataset name, dataset label, number of observations and variables, and the number of character and numeric variables. The title also identifies: dataset compression being used, RESUSE option turned on, indexes present, number of deleted observations, and dataset protection.

By default, the macro invokes the PRINT procedure. The user can specify BY and ID variables for the procedure or bypass it entirely, using PRT simply as a title generator. The user also has the ability to add up to two lines to the default title.

THE MACRO

The complete source for PRT is presented in Exhibit 1. The macro takes up to eight keyword parameters:

1. **DATA=dataset_name.** Identifies the one- or two-level name of the dataset. It is required. If omitted, PRT will print an error message in the Log and stop executing. Exhibit 7 demonstrates the macro's reaction to a missing dataset name.
2. **STATS=stat_request.** If STATS is not entered or if *stat_request* is "Y", PRT generates titles containing the information described earlier. Any other value of *stat_request* results in a title which is simply the name of the dataset (not very descriptive, but still an improvement over "The SAS System.")
3. **ID=id_vars.** Specifies one or more variables to use in PRT's invocation of the PRINT procedure. This parameter is optional. PRT does not check to see if *id_vars* actually exist in the dataset. See Exhibit 6 for an example of ID's use.
4. **BY=by_vars.** Specifies one or more BY-variables to use in PRT's invocation of the PRINT procedure. This parameter is optional. PRT does not check to see if *id_vars* actually exist in the dataset or if the dataset is sorted by *id_vars*.
5. **ONLYTITL=title_switch.** If ONLYTITL is not entered or if *title_switch* is N, PRT will create titles and run the PRINT procedure. If *title_switch* is anything other than "N," the PRINT procedure is bypassed — the titles generated by PRT are ready to use with subsequent procedures. See Exhibit 4 for an example of ONLYTITL's use.

6. **OBS=n_obs.** Requests that *n_obs* observations be displayed by PRT's invocation of the PRINT procedure. The default is all observations.
7. **ADDT1=user_supplied_title_1.** The user may supply additional titles. Surround the title with the %STR function if it contains commas. See Exhibit 5 for an example of ADDT1's use.
8. **ADDT2=user_supplied_title_2.** See the description of ADDT1, above.

Several items in Exhibit 1 are discussed below. The items are keyed to the bracketed, boldfaced numbers in the exhibit.

1. The NOSOURCE option suppresses listing of the macro's statements. This is helpful when the macro is in-stream rather than auto-called.
2. We check for the presence of a DATA parameter. If it is null, we print a message and branch to the bottom of the macro. Note that we do not check to see if the dataset actually exists.
3. We use the dictionary tables and SQL to create macro variables holding the values of several SAS System options. The options will be set to the values required by PRT. Then, just prior to macro termination, they will be reset to their original values.
4. If full titles were requested we begin the SQL and DATA step code contained in this %DO group.
5. Create a WORK dataset containing all information available for the dataset specified by the DATA parameter. Notice that we restrict the search of the TABLES table to member types of "DATA." Views could be included here as well, although information such as number of observations would, by definition, be missing.
6. Use SQL to count the number of character and numeric variables in the DATA parameter's dataset. Both of these items appear in the title.
7. Process the dataset created in step 5. Create macro variables TITLEA and TITLEB.
8. Assign the titles created in item 7. Also assign the optional titles identified by the ADDT1 and ADDT2 parameters.
9. If ONLYTITL doesn't say otherwise, run the PRINT procedure.
10. Reset the system options examined in item 3 to their original values.

EXAMPLES

Exhibit 2 presents the program fragment used to create the datasets used in the examples. Both datasets ONE and TWO have two variables, one numeric and one character. TWO has reusable space and dataset compression turned on.

Exhibit 3 demonstrates the simplest use of the macro. Only the dataset name is specified. All other parameters take their default values. The title contains the dataset name, number of observa-

tions, number of variables, and character/numeric variable counts.

Exhibit 4 shows PRT being used to generate titles. PROC FREQ is run without TITLE statements and so inherits the titles created by PRT.

Exhibit 5 uses the ADDT1 parameter. The default, PRT-generated title is supplemented with text from the user.

In Exhibit 6 we see the additional title line generated when special options and features are in effect. The second title line identifies dataset compression and reusable space.

Finally, Exhibit 7 shows PRT's reaction to omission of the DATA parameter. PRT displays a message in the Log and terminates execution.

EXTENSIONS

This macro was written as an in-house utility in response to specific, narrowly-defined requirements. As such, there are many extensions and modifications which may be useful in other users'

systems. Items such as creation date and sort order may also be gleaned from the dictionary tables. For some users, such information would be a valuable addition to the title lines.

The macro would also benefit from more error-checking. As it is currently written, it only checks for the presence of the required parameter DATA. Checks for the actual existence of the dataset and variable names would be helpful.

QUESTIONS? COMMENTS?

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

ASG, Inc.
2000 Regency Parkway, Suite 355
Cary, NC 27511

919-467-0505 (work)
919-467-2469 (fax)
919-942-2028 (message)

Internet: 0007216987@mcimail.com

Exhibit 1: PRT Macro Source

```
%macro prt(data=, stats=y, id=, by=, onlytitl=N, obs=MAX,
          addt1=, addt2=);
options nosource; [1]

%if &data. = %str() %then %do; [2]
  %put %str(*****);
  %put %str(ERROR! Dataset name not specified by DATA= parameter.);
  %put %str(      Macro execution terminating );
  %put %str(*****);
  %goto bottom;
%end;

[3]
*-- First, get current values of some options we want to turn off. The
  saved values will be restored later, on exit. ;
proc sql noprint;
select setting into :sgen
  from dictionary.options where optname = 'SYMBOLGEN';
select setting into :mprint
  from dictionary.options where optname = 'MPRINT';
select setting into :mlogic
  from dictionary.options where optname = 'MLOGIC';
select setting into :dateopt
  from dictionary.options where optname = 'DATE';
quit;
options nodate nosymbolgen nomprint nomlogic pageno=1;

%let onlytitl = %upcase(%&onlytitl.);
%let stats    = %upcase(%&stats.);
%let data     = %upcase(%&data.);
%if %index(%&data., .) > 0 %then %do;
  %let libname = %scan(%&data., 1, '.');
  %let memname = %scan(%&data., 2, '.');
%end;
%else %do;
  %let libname = WORK;
  %let memname = %&data.;
%end;

/* Asking for title only overrides any value they may have entered for
  STATS parameter. ;
%if %&onlytitl. ^= N %then %let stats = Y;
```

Exhibit 1: PRT Macro Source (continued)

```

title1 " "; /* Blank because first printer lines on HP4M tend to shift
            and/or wrap */
%let titlea = Dataset &libname..&memname.;
%let titleb = %str( );

[4]
%if &stats. = Y %then %do;
proc sql noprint;
create table ____templ as [5]
select * from dictionary.tables
where libname="&libname." & memname="&memname." & memtype = "DATA" ;

*--- Notice TYPE is stored in lower case (but LIBNAME and MEMNAME are
upper case) ---*;

select count(*) into :nchar [6]
from dictionary.columns
where libname="&libname." & memname="&memname." & type = "char";

select count(*) into :nnum [6]
from dictionary.columns
where libname="&libname." & memname="&memname." & type = "num";

quit;

[7]
data _null_;
set ____templ;
length titlea titleb $200;
%if &nobs. = MAX %then titlea = "All Obs. from " %str( );
%else titlea = "First &nobs. Obs. from " %str( );
titlea = trim(titlea) || " Dataset &libname..&memname. " ;
%if memlabel ^= ' ' then
titlea = trim(titlea) || ' (' || trim(left(memlabel)) || ')';
titlea = trim(titlea) ||
": obs=" || trim(left(put(nobs,best.))) ||
", vars=" || trim(left(put(nvar,best.))) ||
" (# char vars= " || trim(left("&nchar. ")) ||
" # numeric vars= " || trim(left("&nnum. ")) || " )";

compress = upcase(compress);
reuse = upcase(reuse);
%if compress ^= "NO" | reuse ^= "NO" | protect ^= "----" |
indxtype ^= " " | delobs > 0 then do;
titleb = "Features, options: ";
%if compress ^= "NO" then titleb = trim(titleb) || " Compressed /";
%if reuse ^= "NO" then titleb = trim(titleb) || " Reuse Space /";
%if protect ^= "----" then titleb = trim(titleb) || " Protect: " ||
protect || " /";
%if indxtype ^= " " then titleb = trim(titleb) || " Index: " ||
indxtype || " /";
%if delobs > 0 then titleb = trim(titleb) || " # Del Obs: " ||
left(put(delobs,comma8.)) || " /";
substr(titleb,length(titleb),1) = ""; * Remove trailing / ;
end;
call symput('titlea', titlea);
call symput('titleb', titleb);
run;
%end;

[8]
title2 "&titlea.";
%if "&titleb." ^= "" %then title3 "&titleb." %str( );
%if "&addt1." ^= "" %then title4 "&addt1." %str( );
%if "&addt2." ^= "" %then title5 "&addt2." %str( );

%if &onlytitl. = N %then %do; [9]
proc print data=&data. %if &nobs. ^= MAX %then (obs=&nobs.); ;
%if &id. ^= %str( ) %then id &id. %str( );

```

Exhibit 1: PRT Macro Source (continued)

```

    if &by. ^= %str() %then by &by. %str( ) ;
    run;
    %end;

* Clean up after ourselves ;
proc datasets library=work nolist;
delete ____templ;
quit;

%bottom: options source &sgen. &mprint. &mlogic. &dateopt.; [10]

%mend;

```

Exhibit 2: Program to Set Up Sample Datasets

```

data one;
char='x';
num = 1;
run;

* Change some system settings. This creates a non-blank second title
  line. ;
options compress=yes reuse=yes;
data two;
set one;
run;

```

Exhibit 3: Minimal Call to PRT

```

----- Program -----
filename macros '[]';

options sasautos=(macros) nocenter;

* Minimal call - just supply dataset name ;
%prt(data=one);

----- Output -----

All Obs. from Dataset WORK.ONE: obs=1, vars=2 (# char vars= 1 # numeric vars= 1)

OBS    CHAR    NUM
  1      X      1

```

Exhibit 4: Use ONLYTITL Parameter

```

----- Program -----
filename macros '[]';
options sasautos=(macros) nocenter;

%prt(data=one, onlytitl=Y);

proc freq data=one;
tables char;
run;

----- Output -----

All Obs. from Dataset WORK.ONE: obs=1, vars=2 (# char vars= 1 # numeric vars= 1)

CHAR    Frequency    Percent    Cumulative Frequency    Cumulative Percent
-----
x              1      100.0              1      100.0

```

Exhibit 5: Use ADDT1 Parameter

Program

```
filename macros '[]';

options sasautos=(macros) nocenter;

* Same as previous example, but add a user-defined title ;
%prt(data=one, addt1=This is another title);
run;
```

Output

All Obs. from Dataset WORK.ONE: obs=1, vars=2 (# char vars= 1 # numeric vars= 1)

This is another title

OBS	CHAR	NUM
1	x	1

Exhibit 6: Enhance PRINT Output by Using the ID Parameter

Program

```
filename macros '[]';

options sasautos=(macros) nocenter;

%prt(data=two, id=char);
```

Output

All Obs. from Dataset WORK.TWO: obs=1, vars=2 (# char vars= 1 # numeric vars= 1)

Features, options: Compressed / Reuse Space

CHAR	NUM
x	1

Exhibit 7: Fail to Specify Required Parameter DATA

Log

```
35          %prt;
*****
ERROR! Dataset name not specified by DATA= parameter.
      Macro execution terminating
*****
```