Appendix Numbers: A Macro to Eliminate Hard Coding

Daphne Ewing
Bio-Pharm Clinical Service, Blue Bell, PA

Abstract

When a project requires appendix numbers to be displayed on each of many output displays, and there is a possibility of them being modified, even once, hard coding them into each program makes a change of this nature time consuming and tedious. Within the report code, be it DATA _NULL_ or PROC REPORT, a macro call can dynamically generate the appendix number making this dynamic portion of the display easy to maintain.

Identifying Report

Assuming each report generated has a unique appendix number which potentially could change, the program which generates the report has to identify this report. Linking the program name to the appendix number makes for an easy cross reference. Using a %LET statement in each program that coincides with the program name allows for decoding through a format statement to generate an appendix number. The PROC FORMAT code should reside in a separate file. Below is a sample for defining appendix numbers which should be placed in the format library:

PROC FORMAT;
  VALUE $TBLCON
    'L.DE Demo' = 'Appendix D.1'
    'L.AE' = 'Appendix D.15'
    'S.DE Demo.A' = 'Appendix F.1.1'
    'S.DE Demo.B' = 'Appendix F.1.2'
    ...
RUN;

The Macro

The macro used to display the appendix number is as follows:

%MACRO APPNUM(ID,TITLE=TITLE,
  RPT=PUTRPT,TTL=4);
  %LET RPT ; %UPCASE(&RPT);
  %IF '&RPT' EQ 'PUTRPT' %THEN %DO;
    &TITLE = TRIM(PUT("&ID",$TBLCON.));
    PUT / %CENTER %STR(&TITLE);
  %END;
  %ELSE
    %IF '&RPT' EQ 'TTLRPT' %THEN %DO;
      DATA _NULL_; 
      CALL SYMPUT(
        "&_APP",TRIM(PUT("&ID",$TBLCON.));
      RUN;
      TITLE=TITLE 
        "&_APP";
  %END;
%END;

The ID parameter is the program identifier set in each calling program. The TITLE variable defines the name of the variable created to hold the appendix number when reporting using a DATA _NULL_. The report type is defined by the RPT parameter. By default this is set to PUTRPT which is for DATA _NULL_ reporting. The other option here is TTLRPT indicating PROC REPORT reporting. The last value to be passed to this macro is the title number for PROC REPORT.

Sample Usage

If we have a program called L123DEMO.SAS which is being used to generate the Listing of Demographic information, we would place the following statement at the beginning of the program:

%LET PGM = L.DE Demo;

This &PGM variable defines the program being used to create the output. Within the DATA _NULL_ a call to the macro will issue the appropriate PUT statement to display the appendix number. For example:

...
DATA _NULL_;  
SET TEST;  
FILE PRINT HEADING=HDG PS=51 LS=132 
    NOTITLES; 
RETURN; 
HDG: 
    PUT / %CENTER "Title 1" 
        @100 "Date Time Program" 
    / %CENTER "Subtitle" ; 
%APPNUM(&PGM) 
RETURN; 
RUN; 

When preparing output using PROC REPORT, the macro is called prior to the PROC REPORT code. It is best placed where the other titles for the report are being generated. For example:

TITLE1 "Title 1 Date Time Program"; 
TITLE2 "Subtitle"; 
%APPNUM(&PGM, TTL=3) 
PROC REPORT DATA = TEST CENTER MISSING; 
   COLUMN VAR1 VAR2 VAR3 ...; 
   ... 
RUN; 

The above code will create a TITLE3 statement with the appendix number. Using the STBLCON shown above, this will produce the following centered line in the output:

Appendix D.1

Changing Appendix Numbers

In the event appendix numbers change, the process to generate the tables with the new appendix numbers is straightforward. Editing the one file containing the PROC FORMAT statement for the TBLCON format is the first step. Make the necessary changes and recreate the format library. If all the appendix numbers have changed, then all programs will have to be re-run. The process can be tedious and confusing. If only some have changed, only run the programs which have appendix numbers that have been modified.

Summary

Dynamic generation of Appendix numbers is a key to saving time and increasing productivity. The APPNUM macro provides a consistent and easy to use method for dynamic appendix number generation.

Acknowledgements

The author would like to thank the programmers at Bio-Pharm for helping in the development of many macros which provided the information for this paper.

Author Information

Daphne Ewing  
Manager, Clinical Programming  
Bio-Pharm Clinical Services  
Four Valley Square  
512 Township Line Road  
Blue Bell, PA 19422  
(215) 283-0770  

SAS is a registered trademark of SAS Institute, Inc.