Paper 050-31

Dynamic Table Numbering – A Straightforward Approach

Diane Hall, Averion Inc., Framingham, Massachusetts

ABSTRACT

Do your clients constantly renumber their tables, listings, or graphs? Do you get tired of going into individual programs to update the numbering on the tables, as well as the output file names?

This paper will describe a simple, straightforward way to number your output displays and files using PROC FORMAT, and a small MACRO. The numbers are driven by a unique text identifier for each display, rather than being hard coded into each program.

THE SETUP

You will need a few things to get this set up.

(1) A central format to define the identifier/number relationship.
(2) A macro to return the number from the format given the identifier.

THE FORMAT

This format should be located in a central place. Do not put it in each program, or you have gained nothing.

```
proc format ;
   value $tnumfmt
                 "DISP"     = 1
                 "DEMO"     = 2
                 "AEALL"    = 3.1
                 "AEREL"    = 3.2
                 "AESAE"    = 3.3
                 ;
end ;
```

DISP is the program name for a Disposition table, DEMO for the Demographics table, and AE for the 3 Adverse Events tables. The 3 AE tables are differentiated by their suffix after the program name in the format. Any naming convention can be used, as long as it makes sense to the project team.

THE MACRO

This macro should be located in a central place. The project macro area is ideal.

```
%MACRO tnum ;
   %global repnum tabnum ;

   %let tabnum = %sysfunc(putc(&ident.,tnumfmt)) ;

   *** Numbers with 1 decimal point ***;

   %if %sysfunc(indexc(&tabnum,".")) > 0 %then %do ;
          %if %sysfunc(putn(%sysfunc(scan(&tabnum.,1,".")),z2)) < 10
              %then %let repnuma = %str(0)& tabnum. ;
          %else %let repnuma = & tabnum;

          %if %sysfunc(putn(%sysfunc(scan(&repnuma.,2,".")),z2)) < 10
              %then %let repnum =
          %sysfunc(scan(&repnuma.,1,"."))%str(.0)%sysfunc(scan(&repnuma.,2,".")) ;
          %else %let repnum = &repnuma. ;

          %let repnum = %sysfunc(compress(&repnum.,%str(.))) ;
   %end ;
```

```
    *** Numbers with NO decimal point ***;

    %else %do ;
           %if %sysfunc(length(&tabnum)) = 1 %then %let repnum = %str(0)&tabnum. ;
           %else %let repnum = &tabnum. ;
    %end ;

%MEND tnum ;
```

This macro takes the table identifier, &ident., and returns &repnum. and &tabnum..

- &repnum is a character formatted number, padded with zeros, to allow the user to name their output files as t01.rep, t02.rep, etc., allowing the files to appear in the proper order when printed using a wildcard character.
- &tabnum. Is the numeric formatted number for use in table titles, i.e. "Table 2 Demographics".

THE PROGRAM STATEMENTS

Prior to calling the TNUM macro, assign a value to &ident.

```
%let IDENT = DEMO ;
```

Call the TNUM macro and PUT the resulting &tabnum and &repnum values to the log, for reference.

```
%tnum ;

%put TABNUM = &tabnum ;
%put REPNUM = &repnum ;
```

Use the created macro variables in a later title statement and/or output file specification.

```
title3 "Table &tabnum.  Demographics" ;
filename _rptout "t&repnum..rep";
```

THE RESULT

In the case of the programming statements above for DEMO, the output is a file named `t02.rep`. The title for this table is "`Table 2  Demographics`".

In the case of the AE tables, one would need to add a suffix to the %let IDENT = ; statement identifying the specific table being actively run. Assuming that the program is organized as a macro, with 3 different macro calls for the 3 subsets (all AEs, related AEs, and Serious AEs), then this is as simple as adding a subset variable to your macro call and tacking it onto your IDENT definition.

```
%let IDENT = AE&subset. ;
```

This program would result in 3 output files; `t0301.rep`, `t0302.rep`, and `t0303.rep`, and the titles would be "`Table 3.1  All Adverse Events`", "`Table 3.2  Related Adverse Events`", and "`Table 3.3 Serious Adverse Events`".

QUESTIONS

As a lead on a project, how do I keep track of my identifiers? I'm bound to forget what I used, and the other programmers are bound to use something I can't decipher.

I keep an excel spreadsheet containing the identifiers, their currently assigned numbers, and the table/listing/graph titles. This makes it relatively easy to go through and figure out what's been used and how to renumber.

Won't it be confusing to have Table 1 and Listing 1 and Graph 1 defined in the same format. I'd like to keep these separate for ease of use. How can I do this?

I myself prefer to have them separate. There is less chance for a change in the listing numbering to inadvertently effect the table numbering. In this case, I set up 3 macros, **tnum**, **lnum**, and **gnum**, and use 3 formats **tnumfmt**, **lnumfmt**, and **gnumfmt**. The only change that needs to be made to the macro other than the name is in the first %let statement. "`%let tabnum = %sysfunc(putc(&ident.,`**`tnumfmt`**`)) ;`" The **tnumfmt** needs to change to either **lnumfmt** or **gnumfmt**. The only change needed to the programs producing the output is to call **%lnum** or **%gnum**, instead of **%tnum**.

CONCLUSION

This is a quick no-hassle way to control table, listing, and graph numbering, as well as the corresponding output file names. The advantages are

(1) Time spent to maintain table, listing, and graph numbering is reduced. (given a client who changes their mind)
(2) Changes can be made in the format. Programs need never be touched, thus preventing inadvertent changes to verified code.

ACKNOWLEDGEMENTS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

CONTACT INFORMATION

Diane Hall
Averion, Inc.
4 California Avenue
Framingham, MA 01701
(508) 416-2659
DHall@averioninc.com