

Automating the Footnote Generation

Sonali Garg, Alexion Pharmaceuticals, Cheshire, CT
Catherine DeVerter, Novella Clinical, Morrisville, NC

ABSTRACT

Footnotes are always needed in outputs for clinical trial analysis. Due to this need programs often get littered with too many footnote statements, especially the programs which produce multiple outputs. What if a standard footnote could be included in every output produced without having to include a FOOTNOTE statement in each program? What if you could renumber standard footnotes without having to edit your programs?

This paper will show how SAS® programmers can automate the numbering of footnotes or titles without changing their programs. The method discussed in this paper will make use of the SQL procedure and SASHELP views.

INTRODUCTION

In the pharmaceutical/CRO industry SAS programmers frequently output a standard footnote at the bottom of each page of a table, listing or figure (TLF) output file which looks something like this:

```
E:/Projxxx/sascode/prog/table/txxx.sas
```

```
DATA CUT: 28DEC2011 12:22
```

For each TLF, a FOOTNOTE statement hardcoding the standard footnote text will have to be written in each new program after all other defined footnotes. A macro variable for the date of the data cut can probably be made to be dynamically read in, but the rest of it still has to be coded in each program.

What if footnote requirements change so that a blank line (or 2) must be added between the output-specific footnotes and the standard footnote, or before the first footnote? Now the programmer will have to go back and renumber all the FOOTNOTE statements. Further, if the requirements continue to change then the programmer will have to go back and fix each program yet again.

The macro introduced here describes another way of tackling this problem without having to change the programs repeatedly.

There are basically 4 steps:

1. Specify the text of your standard footnote.
2. Write or dynamically generate the specific FOOTNOTE statements for your output file. Do not include the standard footnote with the path and filename of the program; instead we will call the %stdfootr macro.
3. Call the %stdfootr macro after all the output-specific FOOTNOTE statements, but prior to the SAS output procedure (such as PROC REPORT).
4. Execute the program.

DEFINE THE STANDARD FOOTNOTE

Our standard footer consists of two pieces: the left side with the program pathname and the right side with the date of the data cut. As such, we define a macro variable for each side: &lfooter and &rfooter.

Ideally, these macro variables will be defined in a setup or initialization program or macro that is called at the beginning of the table program.

CALLING THE % STDFOTR MACRO

Step 2 can be achieved in many ways, the most straightforward being to write the FOOTNOTE statements in the program. Other options include reading in an Excel spreadsheet or Access database that contains all the footnotes required for a specific study and linking them by the table number. For our example, we will just define the footnotes manually in our program.

```

data tables.xyz;
  set abc;
run;

footnote1 "This is the first table footnote.";
footnote2 "This is the second table footnote.";

* Global macro variables lfootr and rfootr have already been defined.;

%stdfootr;

proc report data=tables.xyz nowd;
  column a b c;
  define statements;
run;

```

Notice that the %stdfootr macro is called *after* the output-specific FOOTNOTE statements. The macro uses the SASHELP.VTITLE view to see how many footnotes have been defined to ensure the standard footer is the last footnote on the page.

INSIDE THE %STDFOOTR MACRO

```

%*****
%* Required Parameter: BLANK Indicates how many lines to insert between
%* last defined footnote and standard footnote. Must be >= 0.
%* Default is to insert 1 blank line.
%***** ;
%macro stdfootr(BLANK=1);

%*****
%* Determine how many non-blank FOOTNOTE statements have been defined.
%* This number will be limited to 9 or less if any blank lines are
%* required before the standard footnote.
%***** ;
proc sql noprint;
  create table _fn as
  select *
  from sashelp.vtitle
  where type='F' and text not like "&lfooter%" and text is not missing;

  select left(put(coalesce(max(number),0),2.)) into :lastfn
  from _fn;
quit;

%if %eval(&lastfn+&blank)>9 %then %do;
  %put NOTE (STDFOOTR): You have too many FOOTNOTE statements to allow for the
  standard footnote and number of blank lines specified.;
  %abort cancel;
%end;

%*****
%* If there are defined footnotes, place standard footer below the last
%* defined footnote, adding blank lines in between if BLANK>0.
%***** ;
%if &lastfn>0 %then %do;
  %* Put defined footnote text into macro variables;
  %do h=1 %to %eval(&lastfn);
  proc sql noprint;
    select text into :footnote&h
    from _fn
    where number=&h;
  quit;
  %end;

```

```

%* Output defined footnotes;
%do h=1 %to %eval(&lastfn);
  %if %length(&&footnote&h)>0 %then footnote&h j=1 "&&footnote&h";;
%end;

%* Determine FOOTNOTE number for standard footer;
%let stdfn=%eval(&lastfn+&blank+1) ;
%end;

%*****
%* If there are no footnotes defined in the table program then only the
%* standard footer will be present, preceded by blank lines if BLANK>0.
%***** ;
%else %do;
  %let stdfn=%eval(&blank+1);
  footnote ' ';
%end;

%* Output final FOOTNOTE statement with standard footer text;
footnote&stdfn j=1 "&lfooter" j=r "&rfooter";

%mend stdfootr;

```

The BLANK parameter allows you to easily change the number of blank lines that are inserted prior to your standard footnote.

SAMPLE OUTPUT

Display 1 is produced with ODS RTF using the FOOTNOTE statements in the SAS code above. The %stdfootr macro determined that 2 footnotes were previously defined in the program and then assigned the standard footnote to FOOTNOTE4 so it would be preceded by a single blank line.

The screenshot shows the output of a SAS program. At the top left, it says "Protocol: ABC-123". At the top right, it says "Page 1". In the center, there is a table titled "Table 1: Patient Age". The table has three columns: "Age (years)", "Statistic", and "ABC 50mg (N=5)". The rows are "N", "Mean", and "Min, Max". Below the table, there are two footnotes: "This is the first table footnote." and "This is the second table footnote." At the bottom left, the path "D:/Proj123/sascode/prog/table/Table1.sas" is shown. At the bottom right, it says "DATA CUT: 28DEC2011 12:22".

Age (years)	Statistic	ABC 50mg (N=5)
	N	5
	Mean	37.6
	Min, Max	19, 67

This is the first table footnote.
This is the second table footnote.

D:/Proj123/sascode/prog/table/Table1.sas DATA CUT: 28DEC2011 12:22

Display 1. Output of %stdfootr macro

LIMITATIONS

- Since the %stdfootr macro uses the SASHELP.VTITLE view to determine the position of the standard footer, all output-specific footnotes should be defined prior to the call to %stdfootr.
- The total number of FOOTNOTE statements allowed by SAS is 10. Since our version of %stdfootr inserts a blank line between the last footnote and the standard footer, the number of footnotes defined in the table program cannot exceed 10 minus the value of the BLANK parameter. If a table requires more footnotes, there are some workarounds for this depending on whether you are using ODS (inserting line breaks in the footnote text) or .lst output (using COMPUTE blocks in PROC REPORT).

CONCLUSION

This paper demonstrates how we can automate the footnote numbering and spacing without having to touch the specific program repeatedly. The text of the footnotes can also be modified by using the approach of metadata. This allows for shorter programs, more elegant and portable code and better ways to utilize the power of SAS.

CONTACT INFORMATION

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

Name: Sonali Garg
Enterprise: Alexion Pharmaceuticals
Address: 352 Knotter Drive
City, State ZIP: Cheshire, CT 06410
Email: gargs@alxn.com

Name: Catherine DeVerter
Enterprise : Novella Clinical
Address: 1700 Perimeter Park Drive
City, State ZIP: Morrisville, NC 27560
E-mail: cdeverter@novellaclinical.com

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.

Other brand and product names are trademarks of their respective companies.