Paper AD07-2009

Automate the Process of Creating and Updating Titles and Footnotes of TLG for a Clinical Study Report from a Table Shell Document

Xiangchen (Bob) Cui, Ph.D., Shan Chen, M.S., Mei Wu, M.S.
Sanofi-aventis, Bridgewater, NJ

ABSTRACT

Preparation of TLG (tables, listings, and figures) for a clinical study report typically calls for programming efforts to manually type and/or copy the titles and footnotes from a table shell document into table programs. Since table shells may undergo a lot of changes until very late in the preparation stage, it is highly desirable to automate this process to ensure technical accuracy and operational efficiency.

This paper introduces a SAS macro named %***get_headfoot*** to extract the table numbers, titles, footnotes and output file names from a table shell document and use the information to create a SAS program called ***headfoot.sas***, which contains a collection of SAS macro variables storing table titles and footnotes. ***Headfoot.sas*** will be included in each table program, with the corresponding macro variables invoked to generate titles and footnotes. As the table shell document evolves, simple re-runs of both the macro %***get_headfoot*** and table programs automatically update the titles and footnotes in the output files. The macro ***%get_headfoot*** also saves the title and footnote information into a SAS data set, which can be used not only to create a CSV file serving as a project tracking sheet, but also to generate a tracking report that reflects any deletions, additions of tables, listings, and figures, and/or any updates of table numbers, titles, footnotes and output file names in the new version of table shells. The use of this macro results in significant reduction of programming load and error-prone manual processing in creating and updating report titles and footnotes. It also helps to limit the need to manually enter information for project management activities. Another notable benefit comes from its built-in capability to output a series of reports to aid the user to check the completeness of table shells and track any updates in them, therefore it ensures complete implementation of the table shell requirements and serves as a tracking tool for table shells. The macro is easy to use and works well in the PC environment as well as on the UNIX platform.

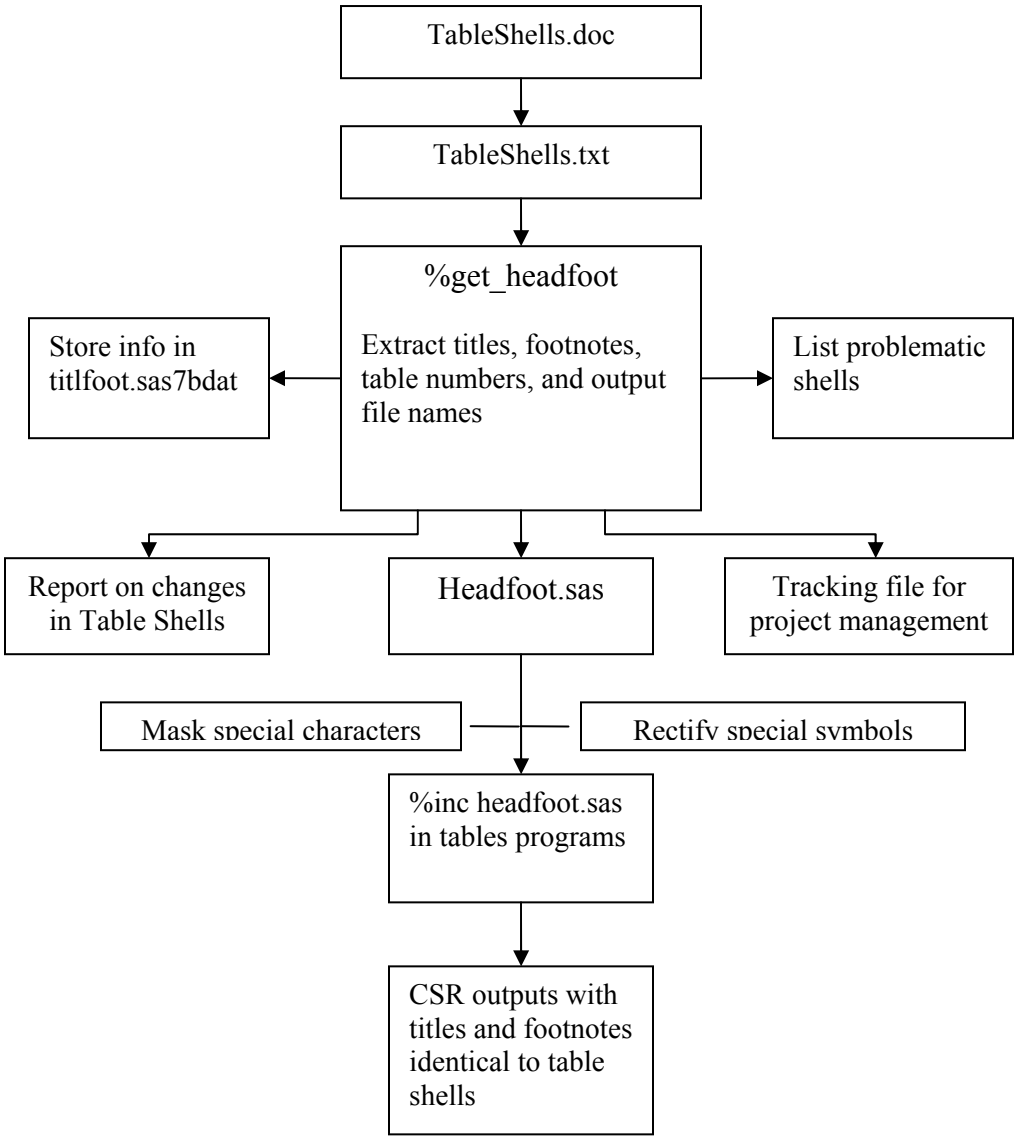
INTRODUCTION

This paper introduces a macro approach to automate the process of creating and updating the table numbers, titles, footnotes and output file names of TLG (tables, listings, and figures) for a clinical study report by extracting these key elements from a table shell document in a Microsoft Word format, storing them into SAS macro variables, and invoking these macro variables in table programs. It reduces the degree of error-prone manual processing as well as the programming load, and greatly improves both work efficiency and quality.

A SAS macro named %***get_headfoot*** extracts the table numbers, titles, footnotes and output file names from a table shell document, uses them to create a SAS program named ***headfoot.sas***, which contains a collection of SAS macro variables storing these information of TLG for a clinical study report. These SAS macro variables are subsequently invoked in table programs to generate titles and footnotes in the final outputs. The macro ***%get_headfoot*** also saves all information in a SAS data set, which can be used to create a CVS file for a project tracking sheet. Checking any duplicates or missing information in table numbers and output file names is another functionality of this macro. When the macro %***get_headfoot*** is re-run, it compares the new information with the previous version if it exists, and reports any changes from the updated table shell document.

Below is an overview of the process flow of the macro %***get_headfoot***.

Figure 1 Overview of Process Flow



SAS MACRO SYNTAX

```
%get_headfoot(      indir = %str(C:\WORK\_min)
                   ,outdir = %str(C:\WORK\_mout)
                ,shelname = template2.txt
                  ,firstcall = Y
               ,tracksheet = );
```

SAS MACRO PARAMETERS

INDIR: Specifies the full folder or directory path where the table shell document is located.

OUTDIR: Specifies the full folder or directory path where the macro's outputs are redirected. &INDIR is the default.

SHELNAME: Specifies the text format table shell document name.

FIRSTCALL: If set to 'N', it requests the macro to compare the table information in the current version with the previous one. "Y" is the default.

TRACKSHEET: Requests the macro to output a CSV file serving as a project tracking sheet. "N" is the default.

SAS MACRO ASSUMPTION

The table shell document is a Microsoft Word format. The table number, title, footnote and output file name should be specified for each distinct table, even if the layout is highly similar. That is, if there are multiple tables with same table structure and footnotes, the table shell document must have one shell for each table. As such, it eliminates the need for SAS programmers to make any assumption, interpretation or edition of each table shell when extracting the table number, titles, footnotes and output file name, hence it facilitates a "seamless" process from reading the table shell document to writing titles and footnotes in table programming.

There are two categories of table reporting programs:

- Reporting tools or macros requiring the titles and footnotes as macro parameters to be invoked in macros calls (refer to example A in section 10),
- Proc report or data _null_ procedure using title and footnote statements to generate the titles and footnotes.

The macro %***get_headfoot*** works in both cases. We will focus on the first situation in this paper, but briefly discuss the second case which is generally simpler than the first one.

BASIC REQUIREMENT FOR TABLE SHELL

- The first title starts with one of the four keywords: Table x (table number), Appendix x (appendix number), Listing x (listing number), or Figure x (figure number).
- The maximum line size of titles and footnotes is 124.
- Space(s)/Tab(s) are expected between table number and title text.
- Output file name should show up in the square brackets [ ] at the end of the first title line and not to exceed 25 characters in length. The naming convention is determined by the team.
- A blank line is expected to separate the titles and table body information. Same applies to footnotes.
- The first footnote starts with either Note or Notes.
- The programming note should follow the last footnote if applicable.

The following is a sample of a table shell.

Table 14 Summary of treatment compliance during treatment period [i_exp_comp1_s_t.rtf]
Safety population

		Drug A	
	Placebo (N=xxx)	7mg (N=xxx)	14mg (N=xxx)
Compliance (%)			
Number	xx	xx	xx
Mean (SD)	xx.x (xx.x)	xx.x (xx.x)	xx.x (xx.x)
Median	xx.x	xx.x	xx.x
Min : Max	xx.x : xx.x	xx.x : xx.x	xx.x : xx.x
<80%	xx (xx.x%)	xx (xx.x%)	xx (xx.x%)
80 - 100%	xx (xx.x%)	xx (xx.x%)	xx (xx.x%)
>100%	xx (xx.x%)	xx (xx.x%)	xx (xx.x%)

Note: % compliance: percent of actual drug taken over the exposure duration.

SAS MACRO OUTPUTS

Headfoot.sas: A SAS program consisting of 3 macros: %***headfoot***, %***titlfoot*** and %***mysymdel*.** Inside the macro %***headfoot***, there is an IF block to assign each title and footnote to a macro variable according to the output file name. At the end of the IF blocks, ***%titlfoot*** and ***%mysymdel*** are defined. %***titlfoot*** has "do loops" to be resolved into the assignments of macro parameters for titles and footnotes and it is called in table programs. %***mysymdel*** deletes these newly created SAS global macro variables right after the table program finishes running.

Titlfoot.sas7bdat:A SAS data set that stores all information about the table numbers, titles, footnotes and output file names, etc..

Bad_shell.lst: A SAS lst file that lists all shells with duplicates or missing information in table numbers and output file names

Del_tbls.lst: A SAS lst file that lists TLG deleted in the new version of a table shell document when macro variable FIRSTCALL is set to N.

New_tbls.lst: A SAS lst file that lists TLG added in the new version of a table shell document when macro variable FIRSTCALL is set to N.

Changed_tlft.lst: A SAS lst file that lists TLG with modified titles and/or footnotes when macro variable FIRSTCALL is set to N.

New_del_tlft.lst: A SAS lst file that lists TLG with added and/or deleted titles and/or footnotes when macro variable FIRSTCALL is set to N.

Diff_tbid.lst: A SAS lst file that lists TLG with changed table numbers and/or output file names when macro variable FIRSTCALL is set to N.

Tracksheet.csv: A CSV file which serves as a project tracking sheet when macro variable TRACKSHEEL is set to "Y".

SAS MACRO METHOD

1. SAVE THE TABLE SHELL IN MICROSOFT WORD AS A FLAT TEXT FILE

Since a SAS program can read a flat text file, but not a Microsoft word file, the table shell document in Microsoft word format should be converted into a flat text file by saving the document as a *.txt file with the 'save line breakers' option checked.

When the table shell is saved as a text file, each line of titles and footnotes may be wrapped by the word processor. The location to be wrapped is determined by both user's computer system and the orientation of the file: Portrait or Landscape. Setting the default line size to 124 and the orientation of the file to landscape can prevent the occurrences of text wrapping. Due to the addition of an output file name to the first title, the problem may still occur for the first title. The macro %***get_headfoot*** can detect it by searching the right square bracket "]"in the first title. If the right square bracket "]" is not found, the wrapping occurs and the macro restores the two wrapped lines into the original one if total length of these two wrapped titles is not greater than 124 (default line size).

Special attention should be paid to special characters in titles and footnotes, for example, ±, μ, †, α, β. These special characters will be incorrectly converted into the symbol '**?**' in text file. Refer to section 5 for the solution to correct these errors.

2. EXTRACT THE TITLES, FOOTNOTES, AND OUTPUT FILE NAMES

The macro %***get_headfoot*** reads the text file line-by-line and generates a SAS data set called ***titlfoot***. It flags the records as titles or footnotes and extracts the output file name for each table shell by identifying the keywords and the square bracket [], respectively. The first title is identified by the keywords, Table/Appendix /Listing/Figure, and the last word inside the square bracket [] on the line in the title section is extracted as the output file name by the assumption that there should not be any square bracket [] in the contents of titles except the one for output file name. The first blank line determines the end of title section. The keywords, Note/Notes, indicate the start of the footnote section, which ends with another blank line, the standard SAS program path, or the first title of the next table shell. It excludes the programming notes if they are detected.

Each record in ***titlfoot*** dataset contains the following columns:

1. Table order from the table shells
2. Index (1=title/2=footnote)
3. Part (titles/footnotes) to indicate it is a title or a footnote
4. Order in titles/footnotes, for example, title1, title2, etc.
5. Table number
6. Output file name
7. Table ID, combining table number and output file name
8. The first title
9. The second title
10. The third title
11. Titles/footnotes from shells
12. Titles/footnotes for reporting

The column “Titles/footnotes for reporting” (item 12) is derived from the column “Titles/footnotes from shells” (item 11) by either directly copying the contents from item 11 or masking special characters in item11 with a percent sign (%) if it has special characters, refer to section 6 for details. Item 12 is used for creating macro variables in ***headfoot.sas***.

The first three titles combined are used as a key for each table shell to compare the new version with the previous one if it exists. Refer to section 8 for details. The SAS data set named ***titlfoot*** is saved into the given folder/directory.

3. DETECT ANY DUPLICATES OR MISSING INFORMATION IN TABLE NUMBERS, AND OUTPUT FILE NAMES

The macro %***get_headfoot*** can detect any duplicates or missing information in table numbers and output file names. The macro reports the findings to the user if anything is detected. Therefore it ensures complete implementation of the table shell requirements.

4. CREATE A SAS PROGRAM HEADFOOT.SAS

Using the output file names, titles, and footnotes from the SAS data set ***titlfoot***, the macro %***get_headfoot*** creates a SAS program named ***headfoot.sas***, which has a macro named %***headfoot***. Inside this program, there is an IF block corresponding to each table, inside each block, each title and footnote is assigned to a macro variable.

%***headfoot*** also contains 2 other small macros: 1) %***titlfoot,*** uses "do loops" to create a series of macro assignments of keyword parameters defined in reporting macros for titles and footnotes, refer to example A in section 10.; 2) %***mysymdel***, deletes the SAS global macro variables newly created inside %***headfoot*** right after each table program is run. %***mysymdel*** ensures a robust SAS program and a good programming practice. Refer to section 10 for the sample call of these two macros and ***headfoot.sas***.

Below is a sample of ***headfoot.sas***, excluding the program header.

```
%macro headfoot;
   %global tit1 tit2 tit3 tit4 tit5 tit6 foot1 foot2 foot3 foot4 foot5 foot6;
   %do i=1 %to 6;
        %let tit&i=%str();
        %let foot&i=%str();
   %end;
   %if &tblname=i_dis_scr1_sr_t %then %do;
        %let tit1=Table 1 Summary of screening failure;
        %let tit2=Screened population;
        %let foot1=%str(% calculated using the number of screened subjects as
                   the denominator.);
   %goto eof;%end;
    ..............................................

   %put WARNING: !!!! Table output name provided in the macro parameter
   &tblname is not in headfoot.sas !!!!;
   %eof:

   %global _i _j;
   %let _i=1;
   %let _j=1;
   %macro titlfoot;
        %do %while(%qcmpres(&&tit&_i)^= %str());
            title&_i=%nrbquote(&&tit&_i), %let _i = %eval(&_i+1);%end;
        %do %while(%qcmpres(&&foot&_j)^= %str());
            footnot&_j=%nrbquote(&foot&_j), %let _j = %eval(&_j+1);%end;
   %mend titlfoot;

   %macro mysymdel;
     %symdel _i _j tit1 tit2 tit3 tit4 tit5 tit6 foot1 foot2 foot3 foot4 foot5
             foot6;
   %mend mysymdel;
%mend headfoot;
```

The statements inside the macro %***titlfoot***: `title&_i=%nrbquote(&&tit&_i)` and `footnot&_j=%nrbquote(&foot&_j)` are macro keyword parameter assignments in the reporting macro for generating titles and footnotes. The title1 to title6, and footnot1 to footnot6 are keyword parameters defined in the reporting macros (refer to example A in section 10). The resolutions of the macro variables: `tit&_i and foot&_j` will be populated into titles and footnotes in the output table by the reporting macro. If the table output name provided in the reporting macro is not found in ***headfoot.sas***, the macro outputs a warning message to the user even though the table program still generates an output table without titles or footnotes. The method above is applicable to the situation where the titles and footnotes are passed as macro parameters as required by the reporting tool and/or macro.

5. DETECT SPECIAL SYMBOLS AND CHARACTER FORMATS AND RESTORE THEM IN HEADFOOT.SAS

Special characters, for example, ±, μ, †, α, β, are incorrectly converted into symbol '?' from a MS word file to a text file. In the SAS data set ***titlfoot***, the column of Titles/footnotes from shells contains symbol '?' if titles and/or footnotes have any special characters in the table shells.

The macro %***get_headfoot*** detects symbol '?' in each title and footnote and outputs the corresponding table information for the user to easily locate the exact special symbols in the table shells and correct the symbol '?' in ***headfoot.sas***. If the SAS output file is in Rich Text Format, the correction is to replace the symbol '?' with the corresponding hexadecimal codes, therefore the table programs can generate these special symbols in the titles and/or footnotes correctly. The method to find the hexadecimal codes for the special symbols was discussed by Cindy Song in her paper titled "Producing Special Characters in SAS Output in RTF" [1].

For example, suppose that %***get_headfoot*** detects the symbol '?' in the title of table 6 and outputs the information shown below.

"Table 6 Number (%) of subjects experiencing TEAEs at a frequency **?**5% presented by primary system organ class and preferred term"

The title of table 6 in table shell is as follows.

"Table 6 Number (%) of subjects experiencing TEAEs at a frequency ≥5% presented by primary system organ class and preferred term"

The hexadecimal code of special character "≥" is 'B3'. The symbol '**?**' is replaced with **'\f1\'B3\f0'**. The assignment statement of macro variable tit1 for table 6 in ***headfoot.sas*** is manually modified as follows.

```
%let tit1=%nrstr(Table 6 Number (%) of subjects experiencing TEAEs at a
          frequency \f1\'B3\f05% presented);
%let tit2= by primary system organ class and preferred term;
```

Below is an example of hexadecimal codes of some special symbols.

≤	\f1\'**A3**\f0
≥	\f1\'**B3**\f0
α	\f1\'**61**\f0
β	\f1\'**62**\f0
≠	\f1\'**B9**\f0
∞	\f1\'**A5**\f0

TimesNewRoman font and the symbol font are defined by \f0 and \f1, respectively. When the hexadecimal code is preceded with a string \', it becomes an RFT text that a word processor can translate it into a character.

In addition to special symbols that need to be rectified in ***headfoot.sas,*** special character formats which are lost during the conversion from Microsoft word document to the text file, e.g., superscript, subscript, and etc, also need to be applied using rtf language, e.g., {\super} for superscript, to the text strings in ***headfoot.sas***.

6. DETECT ANY SPECIAL CHARACTERS AND MASK THEM IN HEADFOOT.SAS

The titles and/or footnotes might contain one or more of the following special characters:

- Unmatched quotation marks and/or parentheses { ‘ “ ( )}
- Macro triggers {& %}
- Operators and Mnemonics {+ - * / < > = ^ | ~ AND OR NOT EQ NE LE LT GE GT IN}
- Semicolon {;}

When these titles and/or footnotes are assigned to the macro variables, these special characters must be masked by macro quoting function during compilation.

Unmatched quotation marks and parentheses must be preceded with a percent sign (%). To be simple, the SAS function TRANWRD is used to add ‘%’ before any quotation marks and parentheses when the macro %***get_headfoot*** detects them. The SAS code is as follows.

```
lines3=tranwrd(lines2,"‘","%'");** quote ‘;
lines3=tranwrd(lines2,'“','%"');** quote “;
lines3=tranwrd(lines2,'(','%(');** (;
lines3=tranwrd(lines2,')','%)');** );
```

It is worthwhile to point out that the single quote and double quotes in the SAS data set ***titlfoot*** from the text file are different from keyboards, i.e, ‘ vs. ', and “ vs. ", which is due to the conversion from Microsoft word document to the text file.

%NRSTR macro quoting function is used to mask special characters above. Below is an example of masking the special character '%' in ***headfoot.sas***.

```
%let foot1=%nrstr(% calculated using the number of safety subjects as the
denominator.);
```

To be robust, macro quoting function %NRBQUOTE is used in the macro %***titlfoot***. Below are the SAS codes to define macro %***titlfoot***.

```
%macro titlfoot;
     %do %while(%qcmpres(&&tit&_i)^= %str());
         title&_i=%nrbquote(&&tit&_i), %let _i = %eval(&_i+1);%end;
     %do %while(%qcmpres(&&foot&_j)^= %str());
         footnot&_j=%nrbquote(&foot&_j), %let _j = %eval(&_j+1);%end;
%mend titlfoot;
```

7. MACRO USED IN PROC REPORT AND DATA _NULL_ PROCEDURE

The methodology and the macros %***headfoot***, %***titlfoot*** and %***mysymdel*** are also applicable to the situation where the proc report and/or data _null_ procedures are used for table reporting. Since they only need title and footnote statements to generate the titles and footnotes, the simplest way is to define title and footnote statements in each IF block directly inside ***headfoot.sas***.

Below is a sample of *headfoot.sas*, excluding the program header.

```
%macro headfoot;
   %if &tblname=i_dis_eot_r_t %then %do;
       title1 "Table 2 Summary of subject disposition";
       title2 "Randomized population";
       footnote1 "% calculated using the number of randomized subjects as
                  the denominator.";
   %goto eof;%end;
   ..........................................
   %eof:
%mend headfoot;
```

Below is a sample SAS program, which is invoking the macro %***headfoot*** before the proc report procedure is used.

```
%inc "C:\work\headfoot.sas";
..............................
%let tblname=i_dis_eot_r_t;
%headfoot;

proc report data=table split='*' headskip headline missing nowd nocenter;
     column (pagecnt index col1
            ("Number and percentage of patients " "__" N_1) );
     define pagecnt  / order order=internal noprint;
```

```
     define index  / order order=data group noprint;
     define col1   / "Factor" width=45  left group flow;
     define N_1    / "80 μg/day* (N=%eval(&tot_1))" width=11 left;
run;
```

8. TRACK ALL CHANGES IN TABLE SHELL

The first call of the macro %***get_headfoot*** generates a SAS data set ***titlfoot*** stored in a given folder. When the table shell document is updated, the macro %***get_headfoot*** should be re-run. It automatically updates the SAS dataset ***titlfoot*** and ***headfoot.sas***. In the meantime, upon the user's request, it can compare the new data set with previous version using the first three titles combined as the key for each table shell and reports its findings of the differences.

If the new version and the old version have a different key (the first three titles), the following information is reported when applicable.

1. **DEL_TBLS.LST**: TLG deleted in the new version of a table shell document
2. **NEW_TBLS.LST**: TLG added in the new version of a table shell document

If the new version and the old version have the same first three titles, the macro will compare their other titles, footnotes, table numbers, and output file names. If it detects any differences between them, the macro reports the following.

3. **CHANGED_TBLS.LST**: TLG with modified titles and/or footnotes

4. **NEW_DEL_TLFT.LST**: TLG with added and/or deleted titles and/or footnotes

5. **DIFF_TBID.LST**: TLG with changed table numbers and/or output file names

Sample output files from tracking the table shells can be referenced in appendix A to E.

9. OUTPUT A CSV FILE AS A PROJECT TRACKING SHEET

Finally, the macro %***get_headfoot*** is able to use the SAS data set ***titlfoot*** to generate a CSV file as a project tracking sheet. As such, there again eliminates the need to manually enter table information for project management activities.

The SAS codes for creating a tracking sheet are as follows.

```
%macro trackingsheet(      din= titlfoot
                     ,prger_nm=
                     ,target_com_dt=
                     ,qc_lvel=
                     ,validator=
                     ,ready_dt=
                     ,v_com_dt=
                     ,outpath=);
```

```
    proc sort data=&din(keep=tblorder lines suborder index outputnm
                            where=(index=1)) out=tf;
      by tblorder suborder;
    run;

    data forxls;
       set tf;
      ...................
    run;

    ods csv file="&outpath.trackingsheet.csv";
    proc report data=forxls nowindows split="^" ls=200;
         columns output_nm tbltit prg_nm prger_nm target_com_dt qc_lvel
                 ready_dt validator v_prg_nm v_output_nm v_com_dt status;
         define output_nm    /display "Output ID " flow;
         define tbltit       /display "Title of Output " width=200 flow;
         define prg_nm       /display "Program Name " flow;
         define prger_nm     /display "Programmer Name " flow;
         define target_com_dt/display "Target Completion Date" flow;
         define qc_lvel      /display "QC Level " flow;
         define ready_dt     /display "Ready for QC Date " flow;
         define validator    /display "Validator Name " flow;
         define v_prg_nm     /display "Validation Program " flow;
         define v_output_nm  /display "Validation output Name " flow;
         define v_com_dt     /display "Validation Completion Date " flow;
         define status       /display "Status/Comments " flow;
    run;
    ods csv close;
%mend;
```

A sample project tracking sheet can be referenced in Appendix F.

10. TWO DIFFERENT SAMPLE CALLS IN TABLE PROGRAMS

Below are two different examples of SAS table reporting programs using the reporting macro %reporttool. %reporttool defines the macro keyword parameters: from title1 to title6 for titles, and from footnot1 to footnot6 for footnotes. The assignments of these parameters are resolved into titles and footnotes by the reporting macro %reporttool. Refer to example A. If any updates of the titles and/or footnotes occur, these macro keyword parameters have to be updated. Hence the call from example A has to be updated. By contrast, example B is a call invoking the macro %***headfoot***. By using the "symbol" ***%titlfoot***, the new call does not need any changes once the macro %***headfoot*** is updated by simply re-running the macro %***get_headfoot***. For a clinical study report, there are a lot of table reporting programs and constant changes of a table shell document until the final stage. Hence the new method results in significant reduction of programming load and error-prone manual processing.

A. Original call:

```
%macro GetTable(tblname=);

   %reporttool(dataset=data_a.addm,
            selpop=%str(random='Y'),
            filename=&tblname,
            varcol=armn[fmt=armnf pool=pooled=7-14(14.1) all=last],
            varlin=sexn# [fmt=sexs label=%nrstr(Gender (n(%)))
                   totalrow=first] racen# [fmt=races label=%nrstr(Race
                   (n(%))) totalrow=first] age [decimal=0 N=n
                   label=%str(Age (year))],

            Title1=Summary of Demographics and Patient Characteristics
                   at Baseline,
           Title2=Safety Population,
           Footnot1=Number corresponds to the count of patients with non
                    missing data,
            options=dest=csr colsize=1.2 1)
    ;

 %mend GetTable;

%GetTable(tblname=i_dem_demo1_s_t);
```

B. Sample call using headfoot.sas

```
%inc "C:\work\headfoot.sas";
%macro GetTable(tblname=);

     %headfoot;
     %reporttool(dataset=data_a.addm,
                 selpop=%str(random='Y'),
                 filename=&tblname,
                 varcol=armn[fmt=armnf pool=pooled=7-14(14.1) all=last],
                 varlin=sexn# [fmt=sexs label=%nrstr(Gender (n(%)))]
                        racen# [fmt=races label=%nrstr(Race (n(%))]
                        age [decimal=0 N=n label=%str(Age (year))],
                 %titlfoot
                 options=dest=csr colsize=1.2 1);

     %mysymdel;

%mend GetTable;

%GetTable(tblname=i_dem_demo1_s_t);
```

CONCLUSION

In summary, the SAS macro %***get_headfoot*** can be used to automate the process of creating and updating titles and footnotes of TLG (tables, listings, and figures) for a clinical study report from a table shell document. For any updates of a table shell, the macro %***get_headfoot*** can be re-run and automatically update ***headfoot.sas*** which contains title and footnote information. Once the table shell document is setup, without the presence of special characters and formats, this process requires no human intervention, hence facilitates a seamless approach. The use of this macro results in significant reduction of programming work load and error-prone manual processing to obtain the titles and footnotes, especially following the modifications to table shells. It also helps to limit the need to manually enter information for project management activities. Another notable benefit comes from its built-in capability to output a series of reports to help the user to check the completeness of table shells and track any updates in them, therefore ensure complete implementation of the table shell requirements. The macro is easy to use, and improves both work efficiency and quality.

REFERENCES

[1] Cindy Song, "Producing Special Characters in SAS Output in RTF." Pharmaceutical SAS Users Group 2000 Conference meeting.

[2] Carpenter, Art. 2003. "Macro Quoting Functions, Other Special Character Masking Tools, and How to Use Them." Proceedings of the Twelfth Annual Northeast SAS User Group Conference. Washington, DC.

[3] O'Connor, Susan. 2003."Secrets of Macro Quoting Functions – How and Why." Proceedings of the Twelfth Annual Northeast SAS user Group Conference. Washington, DC.

ACKNOWLEDGEMENTS

Appreciation goes to Cindy Song for her review and comments.

CONTACT INFORMATION

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

Xinagchen (Bob) Cui, Ph.D.
Sanofi-aventis
908-231-3051
BX2-400B, 200 Crossing Boulevard
Bridgewater, NJ 08807-0890
Xiangchenbob.cui@sanofi-aventis.com

Shan Chen, M.S.
908-304-6653
BX2-409A, 200 Crossing Boulevard
Bridgewater, NJ 08807-0890
Shan.Chen@sanofi-aventis.com

Mei Wu, M.S.
908-304-6655
BX2-409A, 200 Crossing Boulevard
Bridgewater, NJ 08807-0890
Mei.Wu@sanofi-aventis.com

APPENDIX A: Tables, listings, and figures deleted in the new version of table shells

Table order in shells	Table number	The First two titles of table or listing
35	Table_32	Table 32 White blood cells count - Descriptive statistics by visit and at endpoint value ITT population
36	Table_33	Table 33 Red blood cells and platelets count - Descriptive statistics by visit and at endpoint value ITT population

APPENDIX B: Tables, listings, and figures added in the new version of table shells

Table order	Table number	The First two titles of table or listing
26	Table_22	Table 22 Summary of post-baseline PCSA labs Safety population
43	Table_43	Table 43 Summary of post-baseline PCSA vital signs Safety population

APPENDIX C: Tables, listings, and figures with Modified Titles and Footnotes

Table number	Titles/ Footnotes
Table 3	old footnote: Note: Patients are considered in the group of treatment they received at randomization. new footnote: Note: Patients are considered in the group of treatment they **actually** received at randomization.
Table 6	old footnote: LOCF: last observation carried forward. new footnote: LOCF: last observation carried forward. MRI: Magnetic resonance imaging

APPENDIX D: Tables, listings, and figures with added and/or deleted titles and footnotes

Table number	Titles/ added/deleted	Titles/ Footnotes
Table 3	The second footnote was added.	MRI: Magnetic resonance imaging
Table 5	The third footnote was added.	Baseline status is defined according to PCSA criteria;
Table 2	The 4th footnote was deleted.	Number= number of patients with assessment.

APPENDIX E: Tables, listings, and figures with changed table numbers and/or output file names

Table order in shells	Old table number output file name	New table number output file name	The First titles
1	Table **1** i_exc_pop1_r_t	Table **4** i_exc_pop1_r_t	Enumeration of the number (%) of patients included in the analysis populations
2	Table 2 i_disp_**r**_t	Table 2 i_disp_**s**_t	Summary of patients disposition - Number (%)

APPENDIX F:

Output ID	Title of Output	Program Name	Programmer Name	Target Completion Date	QC Level	Ready for QC Date	Validator Name	Validation Program	Validation output Name	Validation Completion Date	Status/ Comments
i_disp.rtf	Table 2 Summary of patients disposition - Number (%)	i_disp.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_disp.sas	v_i_disp.rtf		
i_exp.rtf	Table 3 Summary of extent of exposure	i_exp.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_exp.sas	v_i_exp.rtf		
i_dem_demo_s_t.rtf	Table 4 Summary of demographics and patient characteristics at baseline	i_dem_demo_s_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_dem_demo_s_t.sas	v_i_dem_demo_s_t.rtf		
i_dem_mshist_s_t.rtf	Table 5 Summary of baseline disease characteristics	i_dem_mshist_s_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_dem_mshist_s_t.sas	v_i_dem_mshist_s_t.rtf		
i_eff_bodsumm_i_t.rtf	Table 6 Summary of burden of disease (ml)	i_eff_bodsumm_i_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_eff_bodsumm_i_t.sas	v_i_eff_bodsumm_i_t.rtf		
i_eff_relap1_i_t.rtf	Table 9 Summary of MS relapse	i_eff_relap1_i_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_eff_relap1_i_t.sas	v_i_eff_relap1_i_t.rtf		
i_eff_edss1_i_t.rtf	Table 11 Summary of Expanded Disability Status Scale	i_eff_edss1_i_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_eff_edss1_i_t.sas	v_i_eff_edss1_i_t.rtf		
i_ae_overview_s_t.rtf	Table 12 Overview of adverse events: Number (%) of patients with any adverse event	i_ae_overview_s_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_ae_overview_s_t.sas	v_i_ae_overview_s_t.rtf		
i_pk_pksum1_s_t.rtf	Table 23 Summary of PK concentration by visit	i_pk_pksum1_s_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_pk_pksum1_s_t.sas	v_i_pk_pksum1_s_t.rtf		
i_eff_bodmmrm_i_t.rtf	Table 24 MMRM analysis of BOD using cubic root transformed volume data	i_eff_bodmmrm_i_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_eff_bodmmrm_i_t.sas	v_i_eff_bodmmrm_i_t.rtf		
i_lab_renalpcsa1_s_t.rtf	Table 25 Renal function - Summary of patients with at least one post-baseline PCSA in the on-	i_lab_renalpcsa1_s_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_lab_renalpcsa1_s_t.sa	v_i_lab_renalpcsa1_s_t.rt		
i_lab_electropcsa1_s_t.rtf	Table 26 Electrolytes - Summary of patients with at least one post-baseline PCSA in the on-	i_lab_electropcsa1_s_t.sa	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_lab_electropcsa1_s_t.	v_i_lab_electropcsa1_s_t.		
i_lab_rbcpcsa1_s_t.rtf	Table 27 Red blood cells and platelets count - Summary of patients with at least one post-baseline	i_lab_rbcpcsa1_s_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_lab_rbcpcsa1_s_t.sas	v_i_lab_rbcpcsa1_s_t.rtf		
i_lab_liverdesc_s_t.rtf	Table 28 Liver function - Descriptive statistics by visit and at endpoint value	i_lab_liverdesc_s_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_lab_liverdesc_s_t.sas	v_i_lab_liverdesc_s_t.rtf		
i_lab_metabdesc_s_t.rtf	Table 29 Metabolic function - Descriptive statistics by visit and at endpoint value	i_lab_metabdesc_s_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_lab_metabdesc_s_t.sas	v_i_lab_metabdesc_s_t.rtf		
i_lab_renaldesc_s_t.rtf	Table 30 Renal function - Descriptive statistics by visit and at endpoint value	i_lab_renaldesc_s_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_lab_renaldesc_s_t.sas	v_i_lab_renaldesc_s_t.rtf		
i_lab_electrodesc_s_t.rtf	Table 31 Electrolytes - Descriptive statistics by visit and at endpoint value	i_lab_electrodesc_s_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_lab_electrodesc_s_t.s	v_i_lab_electrodesc_s_t.r		
i_lab_wbcdesc_s_t.rtf	Table 32 White blood cells count - Descriptive statistics by visit and at endpoint value (	i_lab_wbcdesc_s_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_lab_wbcdesc_s_t.sas	v_i_lab_wbcdesc_s_t.rtf		
i_lab_rbcdesc_s_t.rtf	Table 33 Red blood cells and platelets count - Descriptive statistics by visit and at endpoint value	i_lab_rbcdesc_s_t.sas	John Smith	15-Dec-08	3	12-Dec-08	Jane Doe	v_i_lab_rbcdesc_s_t.sas	v_i_lab_rbcdesc_s_t.rtf		