PharmaSUG 2014 - Paper BB14

A SAS® Macro Utility to Modify and Validate RTF Outputs for Regional Analyses

Jagan Mohan Achi, PPD, Austin, TX
Joshua N. Winters, PPD, Rochester, NY

ABSTRACT

Clinical Study Reports (CSRs) of clinical trials require the development of the tables, listings, and figures (TLFs) in Rich Text Format (RTF) outputs. The analyses are usually complex and the number of RTF outputs is fairly large depending on the magnitude of the studies. It is not uncommon to see requests to repeat the analyses by region and/or country for multicenter trials to understand the differences in geographical locations. For regional analyses, changes to an RTF output can be but not limited to changing the filename, adding additional information in the title, and modifying the hidden bookmark which is used to ensure that the report refers to the specific TLF from CSRs. Such changes have to be made to the original TLF programs which are time consuming. Therefore, a macro was developed to systematically modify the RTF outputs, save the processed RTF outputs with new names, and validate the original and modified RTF outputs to confirm that the only intended changes were made to the RTF outputs.

INTRODUCTION

This paper demonstrates how to read RTF outputs into SAS and make intended changes to the contents of the outputs such as bookmarks, titles, footnotes, column labels, etc., and save the modified outputs with new names. Then, the macro reads both original and modified versions of the RTF outputs into SAS and compares the two versions line by line, producing a report displaying the lines where changes were made.

METHOD

Process 1: **Read RTF outputs into SAS and make intended changes to the contents of the outputs such as bookmarks, titles, footnotes, column labels, etc.**

This is a macro call to process all the rtf files from a directory- path.

1.1 This step creates a list of rtf files from a user define path and stores them in file macro variables.

```
filename dirlist pipe "dir /-c /q /t:c ""&path""" ;

data fnums1(keep=filename);
      infile dirlist missover pad length=len;
      input @01 line $varying200. len eof;
      if index(upcase(line),'.RTF');
      fname=line;
      array temp_ $200 date time timef sizec owner filename;
      do over temp_;
            temp_=scan(fname,_i_,"");
      end;
      filename=substr(filename,1,length(filename)-4);
run;

data _null_;
      set fnums1;
      call symputx(catt("file",_n_),strip(filename));
      call symputx("totfiles",strip(put(_n_,best.)));
run;
```

Loop over the number of RTF outputs. Each iteration will read the file and does the following actions:

1.2 Rename rtf file with specific region. It uses rename functionality in a data step to rename a file with region id.

```
rc=rename
("&path.&infile..rtf","&path.\&infile._&regionid..rtf",'file')
```

1.3 Read the rtf file and store the title text into a macro variable

```
infile "&path.&infile..rtf" length=len end=eof lrecl=2000;
%if %substr(&infile.,1,1)=T %then %do;
            if index(upcase(linetxt),"TABLE") then do;
                  tittxt=scan(scan(linetxt,2,"{"),1,"\");
            end;
%end;
data _null_;
      set null_;
      tittxt=substr(tittxt,1,length(tittxt)-1);
      call symputx("title3",strip(tittxt));
run;
```

1.4 Update Title, Bookmarks and References and create a .tmp file from rtf output to store the processed changes.

```
if index(linetxt,"{\header\pard\plain\qc{") then header=1;
if header=1 then do;
      linetxt=tranwrd(linetxt,"&title3.","%trim(%bquote(&title
      3._&regionid))");
      linetxt=tranwrd(linetxt,"\cell}","_&regionid. \cell}");
end;
if index(linetxt,'{\*\bkmkstart}{\*\bkmkend}') and titl=1 then
do;
      linetxt=tranwrd(linetxt,"&title3","%trim(%bquote(&title3
      ._&regionid))");
end;
```

1.5 Save the modified outputs

Put the updated file into a .tmp file and read into the modified output. After the file is modified, the .tmp file is deleted. Replace the existing file with the modified output .tmp file using the steps below.

```
data _null_;
    infile "&infile..tmp " LENGTH=len end=eof lrecl=2000;
    file "&PATH.&infile..RTF " lrecl=2000;
    input @1 linetxt $VARYING2000. len;
    put linetxt $varying2000. len;
run;
%let rc = %sysfunc(filename(fname,&infile..tmp)) ;
   %if &rc eq 0 %then %do ;
   %let rc = %sysfunc(fdelete(&fname)) ;
   %let rc = %sysfunc(filename(&fname)) ;
%end ;
```

Process 3: Read the original and modified RTF outputs into SAS and compare the two versions

3.1 Output original and modified RTF file names from their respective folders into two datasets.

```
filename ori pipe "dir ""\\TLF\Output\&region\Original"" /b";
filename mod pipe "dir ""\\TLF\Output\&region\Modified"" /b";

data dirlist_ori;
     length txt_line $256 ;
     infile ori length=reclen ;
     input txt_line $varying256. reclen ;
run ;

data dirlist_mod;
     length txt_line $256 ;
     infile mod length=reclen ;
     input txt_line $varying256. reclen ;
run ;

proc sort data= dirlist_ori;
      by txt_line;
run;

proc sort data= dirlist_mod;
      by txt_line;
run;
```

3.2 Put the total number of RTF outputs into a macro variable.

```
%let dsid=%sysfunc(open(dirlist_ori, is));
%let cnt= %sysfunc(attrn(&dsid.,nobs));
%let rc=  %sysfunc(close(&dsid));
%put Total number of RTF outputs= &cnt;
```

3.3 Loop over the number of RTF outputs. Each loop will read an original and its respective modified RTF outputs into SAS. Each pair of RTF outputs will be compared line by line.

```
libname &region "\\TLF\Output\&region";
proc datasets lib=&region memtype=data nolist kill;
quit;

data &region._report;
      set dirlist_ori;
run;

%do i=1 %to &cnt;
   data _null_;
      set dirlist_ori;
      if _n_ eq &i then call symput('rtf_ori',strip(buffer));
   run;

   data _null_;
      set dirlist_mod;
      if _n_ eq &i then call symput('rtf_mod',strip(buffer));
   run;
```

```
/* Read the pair of RTF outputs into SAS*/
data rtf_ori;
   infile "\\TLF\Output\&region\Original\&rtf_ori"
   LENGTH=len end=eof lrecl=2000;
   input @1 linetxt_ori $VARYING2000. len;
   index= _n_;
run;

data rtf_mod;
   infile "\\TLF\Output\&region\Modified\&rtf_mod"
   LENGTH=len end=eof lrecl=2000;
   input @1 linetxt_mod $VARYING2000. len;
   index= _n_;
run;

/* Merge the pair of RTF outputs together.  Flag rows with
   any differences other than the intended changes. Output
   rows which are different from each other due to the
   intended changes.  Output other rows as well. */
data temp  (drop= linetxt_mod1)
     temp1 (drop= linetxt_mod1);
   length linetxt_mod1 $2000;
   merge rtf_ori rtf_mod end= eof;
   by index;
   linetxt_mod1= tranwrd(linetxt_mod,"_&region",'');
   if strip(compress(linetxt_ori)) ne
      strip(compress(linetxt_mod1)) then flag= 1;
   else flag= 0;
   if compress(linetxt_mod) ne
      compress(linetxt_ori) then output temp1;
   else output temp;
run;

/* Report whether the pair of RTF outputs contains
   differences other than the intended changes.*/
%let flag=;
proc sql noprint;
   select      sum(flag)
   into        :flag
   from        temp;
quit;
%let flag= &flag;

data &region._report;
   set &region._report;
   if strip(buffer) eq "&rtf_ori" then flag= &flag;
run;

%if &i lt 10 %then %let dset_name= 00&i;
   %else %if &i ge 10 & &i lt 100 %then %let dset_name= 0&i;
   %else %let dset_name= &i;

data &region..&region._&dset_name. ;
   set temp;
run;
```

```
      /* Append rows which are different due to the intended
         changes into the final report. */
      data temp1;
         length deliverable_id $200 rtf_num 8;
         set temp1;
         deliverable_id= "&rtf_ori";
         rtf_num= &i;
         drop flag;
      run;

      proc append base=&region._report data= temp1 force;
      run;
   %end;
```

Process 4: **Create a report including the modified text where the intended changes are made in comparison to the original text**

```
   proc copy in= work out= &region;
         select &region._rtf_modification_report &region._report;
   run;

   ods listing close;
   ods html file=
   "\\TLF\Output\&region\&region. rtf modification report.xls";
   proc print data= &region..&region._rtf_modification_report;
   run;
   ods html close;

   ods html file= "\\TLF\Output\&region\&region._report.xls";
   proc print data= &region..&region._report;
   run;
   ods html close;
```

CONCLUSION

The above process demonstrates how to systematically modify RTF outputs and validate the modified outputs to confirm that the only changes are the intended changes. In summary, this application efficiently modifies RTF outputs in order to save resources especially when time is a constraint.

CONTACT INFORMATION

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

Name: Jagan Mohan Achi
Enterprise: PPD
Address: 7551 Metro Center Drive
City, State ZIP: Austin, Texas
Work Phone: (512) 747-5718
E-mail: jagan.achi@ppdi.com

Name: Joshua N. Winters
Enterprise: PPD
City, State ZIP: Rochester, New York
Work Phone: (910) 558-4143
E-mail: joshua.winters@ppdi.com