

How Simple Table of Content can make CSR Table, Listings, and Figures Easily Accessible?

Yogesh K. Pande, Merck & Co., Inc., Kenilworth, NJ, USA

ABSTRACT

Clinical Study Report (CSR) contains 100+ Tables, Listings, and Figures (TLFs). If a study SAS® programmer programmatically creates Table Of Content (TOC) with titles, page numbers, and includes all displays as part of the same document, the file becomes quite large. Navigating from one display to another becomes slow. Also, the TLF folder can get crowded with lots of TLF files and to know which filename belongs to which title becomes difficult. To overcome these challenges, this paper is introducing a macro that can generate TOC having the TLF titles and it makes sure that the TLFs are easily accessible via hyperlink for each display.

Keywords: SAS®, Macro, analysis, CSR.

INTRODUCTION

The SAS® programmer generates multiple analysis tables, listings, and figures for supporting the study CSR. Before programming analysis TLFs the programmer receives details from statistician in terms of mock document having the display layout, TLF specifications, and complete list of TLFs required to support the study CSR. The only missing part sometimes study statistician does not give is the TLF filename. It is very important to identify which TLF filename contains which title. Given the importance to know this it is a challenge to identify which filename is displaying which title.

Once the programming team starts programming analysis displays (TLFs) the display folder gets crowded with multiple display files. Besides few characters, in the TLF filename, indicating which display are common and which are specific to the study protocol these generated display files does not follow specific naming conventions. In fact, it's not possible to design specific naming convention for each display file.

Below screenshot shown in Figure-1 gives an example on how many files TLF folder can have. It is hard finding which file contains what title.

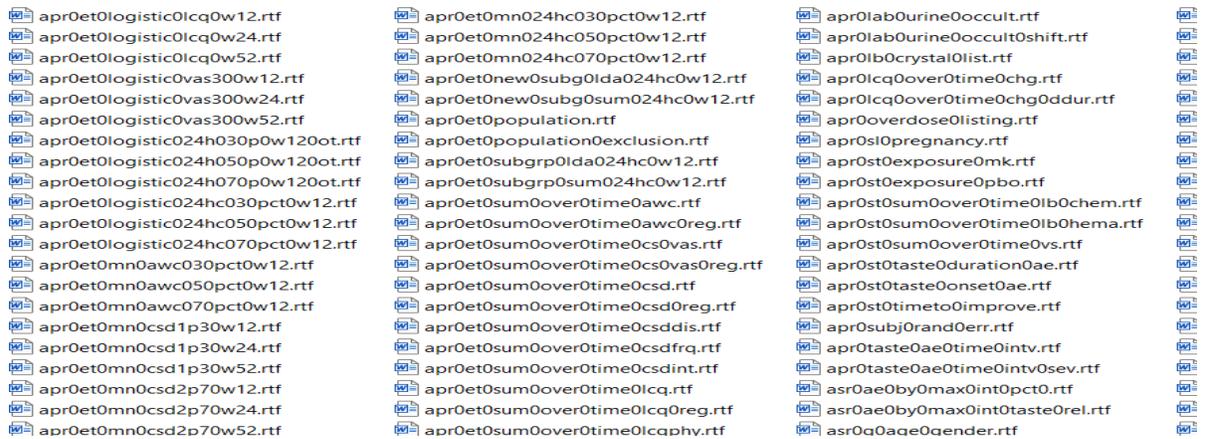


Figure-1

To overcome the identity issue of these TLFs, this paper introduces a SAS® macro which will create a TOC that will show two columns table title and hyperlink. The hyperlink will be named after the TLF

filename. In addition to the two columns as shown in Figure-2 user can also add another column on the right to show which analysis dataset was used to program the TLF. The only requirement to add this source dataset column in this TOC is that the TLF should have source information at the bottom part of the display body. More on the macro and coding in the next section.

Display Title	Display Link
<title>	<filename>.rtf

Figure-2: Table of Content

MACRO CALL AND SAS® CODE:

```
%tocfortlf(input_folder = /*input folder cannot be empty - rtf location*/
,ext = rtf /*extension always rtf for making the macro work*/
,landing_folder = /*TOC output folder link*/
,tocfile = /*TOC filename*/
,rtf = Y /*rtf y/n*/
,docx = /*docx y/n*/
,docxlink = /*docx table link*/
,pdf = /*pdf y/n*/
,acrolink = /*pdf link*/
);
```

Consider below as an example to better understand how this macro works to generate the TOC given in Figure 2. For flexibility, programmer can convert below code into above macro for creating TOC.

Step-1: Let's consider 2 display files in the analysis folder as shown below and we need to create the TOC to know which file contains what title

TLF folder: C:\example\.

Tables in above location

1. xxx0g0base0char.rtf
2. xxx0st0ae0soc05pct.rtf

Step-2: Use below code to get the folder location and the TLF filenames in the dataset.

```
filename dirlist PIPE "dir C:\example\*.rtf /B /S";
data tocfiles;
infile dirlist missover pad;
input link $2000. ;
mfold = "C:\example\";
tfold = tranwrd(link,scan(link,-1,'\'),' '); /*This will give TLF filename*/
run;
```

Step-3: Use below code to create dataset out of each TLF file. In this case 2 datasets. Make sure to select only first 100-lines from the TLF filename for making sure we get the display title in the dataset.

```

data example1;
filename rtf "C:\example\xxx0g0base0char.rtf";
infile rtf missover length = 1 lrecl = 200;
input STRING $varying2000. 1;
  if _n_ >= 100 then stop;
run;
/*Perform similar steps as above for 2nd display */

```

Step-4: Use below programming technique to extract title portion from above TLF dataset.

```

data example_1;
  set example1;
  if string = ' ' then delete;
  strc = catc(string);
  if (substr(strc,1,1)= '\ ' or substr(strc,1,1)='{') then delete;
run;

```

```

data example_1;
  set example_1;
  titno = _n_ ; /*This gives number of titles*/
  tablink = "C:\example\xxx0g0base0char.rtf" ;
  table = "xxx0g0base0char";
  strc = tranwrd(strc, "\line", "^R/RTF'{'\line}'");
run;
/*Perform similar steps as above for 2nd display */

```

Step-5: Use variable that counts number of titles from above step to separate each title line into variables.(Example: If table has 3 titles then after below step there will be three variables.)

```

proc transpose data=example_1 out=ex1t;
by tablink table;
var strc;
id titno;
run;

data ex1t;
  set ex1t;
/*Concatenate the titles using CATX function*/
titles = catx(' ',_1,_2);
/*Above step will have _3 variable if there are more then 2 titles in the
table.*/
  keep table tablink titles ;
run;
/*Perform similar steps as above for 2nd display*/

```

Step-6: Stack all dataset together. In this case 2 datasets since we have 2 TLF files in Step-1.

```

/*Stack datasets together, in this case 2 datasets as below*/
data _final;
set ex1t ex2t;
run;

```

Step-7: Use PROC REPORT to generate the output like the format shown in Figure-2. Below is the screenshot of the output shown in figure 3.

```

options nodate nonumber orientation=landscape;
/*ODS RTF statement Use style = style.rtf for the
output*/title1 f=arial justify=center h =5 "Table of
Content" ; ods escapechar='^';

```

```

proc report data=_final nowd headline headskip split="";
column titles tablink table ;
  /*use noprint for variable 'tablink' while defining the column*/
compute table ;
call define(_col_, "url", tablink);
call define (_col_, 'style',
'style=[foreground=bluetextdecoration=underline]');
endcomp;
run;
/*Close ODS RTF*/

```

Table of Content

Display Title	Display Link
Subject Baseline Characteristics All Subjects Randomized and Treated	xxx0g0base0char
Subjects With Adverse Events (Incidence >= 5% in One or More Treatment Groups) All Subjects as Treated	xxx0st0ae0soc05

Figure-3: Program Output

Further updates on the TOC output (Figure-3) can be made to add columns showing links for other file extensions (.docx or .pdf) and add a column for data source.

CONCLUSION

This paper tried to emphasize on how important it is to make sure that the TOC shows the title information embedded within each TLF file name and how important it is for having ease of access to multiple CSR tables generated by team of study programmers and statisticians. The macro and the code described in this paper will facilitate any level of SAS® programmers to quickly generate the Table of Content (TOC). I would encourage to use the programming technique given in the paper, take notes and get flexible for generating the TOC.

REFERENCE

1. SAS®/MACRO Software: version 9.4, SAS® Institute Inc., Cary NC
2. SAS®/BASE Software: Version 9.4, SAS® Institute Inc., Cary NC
3. SAS® 9.4 Macro Language Reference
4. Introduction to SAS® 9.4 PROC REPORT.

CONTACT INFORMATION

Your comment and questions are valued and encouraged.

Yogesh Pande
Principal Scientist, Statistical Programming .
Merck & Co., Inc., Kenilworth, NJ, USA
E-mail: yogesh.pande@merck.com