

Automate Compiling Clinical Appendix Tables and Listings into a Review Package with a Hyper-Linked Table of Contents

**Xiangchen (Bob) Cui, Ph.D., Wei Zhang, M.S.
Sanofi-aventis, Bridgewater, NJ**

ABSTRACT

Compilation of a large number of tables and listings in clinical studies from SAS outputs into a user-friendly review package with a hyper-linked table of contents (TOC) is a labor-intensive process in clinical study reporting. A complete TOC is not only a key for publication purpose, but also serve as a key element to both medical and statistical review at every stage of a clinical trial.

This paper introduces a SAS macro and Microsoft VBA macros that are executed under SAS environment to automatically generate such a review package with a hyper-linked TOC. Specifically in our case, each of TOC entries contains leading dots before a given page number for a clinical report. This format also applies to a table of all footnotes and titles as separate files. Furthermore, the selections to compile such a review package from clinical tables and listing and their ordering in the package will be determined by a user-specified external file, which lists the file names of SAS outputs for packing and whose ordering in the file will be the one for the package. By default, all files will be packed in an alphabetical order of file names. For debugging and Q/A purposes, the program sends a series of appropriate messages to help the user to track how the SAS list files were generated. The macro works both PC environment and UNIX platform while its VBA macros, however, have to be executed in Windows environment.

INTRODUCTION

In pharmaceuticals, clinical study reports will include a large number of tables and listings. Compiling them into a user-friendly review package and generating a table of contents for easy reference are a labor-intensive process. A SAS macro reads each SAS list file in a given folder or directory, appends one to another in a given order by issuing DOS or UNIX commands using X statement within SAS, calculates the total page number, retrieves the titles and footnotes from the first page of each file, generates a TOC, combines the TOC and the concatenated list files into a review package, and outputs a SAS data set containing titles and footnotes for later use. A dynamically generated VB program is called in SAS environment for both document formatting and creating hyperlinks within a review package. These VBA macros automatically convert each of SAS macro output list files into a Microsoft word file, format the MS Word document, and add hyper-links to the package. The SAS macro automates the process quickly and accurately. This process hence improves work efficiency and quality.

SAS MACRO SYNTAX

```
%mgettoc (   indir=%str(C:\WORK\_minputs),
            orderfile=%str(C:\WORK\_minputs\orderfile.txt),
            stdyname=%str(Study101),
            outdir=%str(C:\WORK\_mout),
            titlstrt=2,
            titlend=4,
            titlcut=130,
            tocls=30,
            alftnote=yes,
            tottbls=yes,
            alltits=yes);

run;
```

SAS MACRO PARAMETERS

INDIR: Specifies the full folder or directory pathname where SAS lst files are located.

ORDERFILE: Specifies external filename and its location used to determine the ordering in the final outputs. It is a text file and lists all lst file names with a single column. By default, no file is specified and the macro will sort the SAS lst files alphabetically.

STDYNAME: Specifies the study name in the title of TOC.

OUTDIR: Specifies the full folder or directory pathname where the macro's outputs will be redirected. &INDIR is default.

TITLSTRT: Specifies the starting line number in the title part of each file. 1 is the default.

TITLEND: Specifies the ending line number in the title part of each file.

TITLCUT: Specifies the maximum of line size of TOC. 130 is default.

TOCLS: Specifies the maximum of line number of TOC. 30 is default.

ALFTNOTE: Specifies the request to output all footnotes into a file, named _fotnots_.lst. "no" is default.

TOTTTBLS: Specifies the request to output the concatenated files into a package, named _alltbl_.lst. "no" is default.

ALLTITLS: Specifies the request to output a SAS data set, named *_altitl_.sas7bdat* , to the given output folder or directory (&outdir). “no” is default. It contains all titles and footnotes with full folder or directory pathname and file names, total page numbers of files, and the sizes of titles and footnotes.

SAS MACRO ASSUMPTION

First, there is a long line with at least 120 characters “_” before footnote section on each page. The number can be easily modified inside the SAS macro for users. The SAS macro does not create an extra parameter to specify the condition for simplicity. The paper just provides the general concepts. Secondly, in PC platform, Microsoft VBA macros have to be stored in Microsoft macros for they will be executed inside the execution of the SAS macro to convert SAS lst files into Microsoft word files, format them, and add hyper-links into the package.

SAS MACRO METHOD

1. DETERMINE THE PLATFORM

The SAS macro works both in PC platform and UNIX platform. At beginning, the macro detects the platform and assigns the macro variable OS the value “UNIX” or “WINDOW”, which will be used through out the macro. The following SAS code can be used to accomplish it.

```
%if "&sysscp" = "HP 800" %then %do;
  %let os=UNIX;
  %if %index(&indir,\\) %then %do;
    %put NOTE: ERROR: The directory in unix should have "/", not "\\".;
    %goto bye;%end;
%end;
%else %if "&sysscp" = "WIN" %then %do;
  %let os=WINDOW;
  %if %index(&indir,/) %then %do;
    %put NOTE: ERROR: The directory in window should have "\", not "/".;
    %goto bye;%end;
%end;
```

2. DETECT IF THE GIVEN INPUT DIRCTORY OR FOLDER EXISTS

The input directory or folder is required. The macro detects if it exists. If it was not provided or it was wrongly specified, the macro would stop and give the error message.

```
%if &indir= %str() %then %do;
  %put NOTE: ERROR: The macro parameter indir is required!;
  %goto bye;
%end;
%else %do;
```

```

%let filrf=testdir;
%let rc=%sysfunc(filename(filrf,&indir));
%let did=%sysfunc(dopen(&filrf));
%if &did=0 %then %do;
    %put NOTE:ERROR:The given folder/directory does not exist.
        The SAS program stops here!;
    %goto bye;
%end;
%else %do;
    %let rc=%sysfunc(dclose(&did));
    *** remove the last charater if it is '/' or '\';
    %let indir1=%sysfunc(trim(&indir));
    %let try=%substr(%sysfunc(reverse(&indir1)),1,1);
    %if %index(&try,\) or %index(&try,/) %then %do;
%let indir=%sysfunc(reverse(%substr(%sysfunc(reverse(&indir1)),2)));
    %end;
%end;
%end;

```

3. DETECT IF THE GIVEN OUTPUT DIRCTORY OR FOLDER EXISTS

By default, the output files have the same location as the given input one. The macro detects if the given output directory or folder exists. If it was not provided or it was wrongly specified, the output files would be redirected to the given input directory or folder.

```

%let wrongout=NO;
%if &outdir=%str() %then %do;
    %let outdir=&indir;%end; %else %do;
    %let filrf=testdir;
    %let rc=%sysfunc(filename(filrf,&outdir));
    %let did=%sysfunc(dopen(&filrf));
    %if &did=0 %then %do;
        %let outdir=&indir;%let wrongout=YES;%end;
    %else %do;%let rc=%sysfunc(dclose(&did));
        %let outdir1=%sysfunc(trim(&outdir));
        %let try=%substr(%sysfunc(reverse(&outdir1)),1,1);
        %if %index(&try,\) or %index(&try,/) %then %do;
%let outdir=%sysfunc(reverse(%substr(%sysfunc(reverse(&outdir1)),2)));
    %end;
%end;
%end;

```

4. RETRIEVE THE LST FILE NAMES IN THE GIVEN INPUT DIRECTORY OR FOLDER

In UNIX, UNIX command retrieves the lst file names. A SAS data set is generated and it has one column with all lst file names. It will be merged with the given order file to select the files to be concatenated.

```

%if &os=UNIX %then %do;
    X cd; X cd &indir; X ls *.lst > &outdir/_lstnam_.txt;

```

```

filename lstnm "&indir._lstnam_.txt";
data lstnames;
  infile "&outdir./_lstnam_.txt" dsd;
  input name:$char200.;
  name = trim(left(name));
  if name in("_toc_.lst","_alltbl_.lst","_fotnots_.lst") then
    delete;
run;
X cd; X cd &outdir;X rm _lstnam_.txt;
%end;
%else %if &os=WINDOW %then %do;
  data lstnames(keep=name);
  rc=filename('f_dir', "&indir");
  id_dir=dopen('f_dir');
  memcount=dnum(id_dir);
  do i=1 to memcount;
    name=dread(id_dir, i);
    if scan(name, 2)='lst' then output;
  end;
  rc=dclose(id_dir);
run;
%end;

```

5. DETERMINE THE ORDERING IN THE FINAL OUTPUTS

If an external text file is specified, it will be combined with all lst file names retrieved in section 4 to select the lst files for packing and to determine their ordering. The macro outputs any messages if the order file does not agree with what is found in the directory or folder. The files which were specified in the external order file were not found, or ones which were found were not specified in the external order file would be excluded from the final outputs. Their names would be output for user's reference. The macro outputs the ls file names and their ordering in the TOC and ones with empty tables. If no external order file is specified, the macro packs all files found in the directory or folder by the alphabetical order.

```

%if &orderfile^=%str() %then %do;
  %let try=%substr(%sysfunc(reverse(&orderfile)),1,4);
  %if %sysfunc(indexw(&try,txt.))=0 %then %do;
    %put NOTE: ERROR: The order file must be a text file with .txt as
      an extension. The SAS program stops here!;
  %goto bye;
%end;
%if &os=UNIX %then %do;
  %if %sysfunc(fileexist("&orderfile"))>0 %then %do;
    data orderfile;
      infile "&orderfile" dsd dlm='09'x;
      input lstname: $15.;
      if index(lstname, ".lst") then
        lstname=scan(lstname,1, '.');
      lstorder+1;
    run;%end;
  %else %do;
    %put "NOTE: The order file &orderfile is unable to find!";
    %goto wo_order;
  %end;

```

```

        %end;
    %end;
%else %if &os=WINDOW %then %do;
    %if %sysfunc(fileexist("&orderfile"))>0 %then %do;
        data orderfile;
            infile "&orderfile" dsd dlm='09'x;
            input lstname: $15.;
            if index(lstname, ".lst") then
                lstname=scan(lstname,1, '.');
            lstorder+1;
        run;%end;
    %else %do;
        %put "NOTE:The order file &orderfile is unable to find!";
        %goto wo_order;
    %end;
%end;
%let flag1 = 0 ;
%let flag2 = 0 ;
proc sort data=orderfile;by lstname;run;
proc sort data=lstnames(drop=lstorder);by lstname;run;
data notfound
    excluded
    lstnames;
merge lstnames(in=a) orderfile(in=b where=(lstname^=' '));
by lstname;
if a and not b then do;
    call symput('flag1','1');
    output notfound;end;
else if not a and b then do;
    call symput ('flag2','1');
    output excluded;end;
else if a and b then output lstnames;
run;
%if &flag1=1 %then %do;
proc sort data=notfound nodupkeys;by lstname;run;
proc print data=notfound;
title; title1;
title2 "The following files were found in the given directory but
        not specified in the order file";
title3 "%upcase(&orderfile).";
title4 "They will not be included in the outputs.";
var lstname;
run;
%end;
%if &flag2=1 %then %do;
proc sort data=excluded nodupkeys;by lstname;run;
proc print data=excluded;
title;title1;
title2 "The following files specified in the order file";
title3 "%upcase(&orderfile) were not found!";
title4 "They will not be included in the outputs.";
var lstname;
run;
%end;
%end;
%else %do;
%wo_order;;

```

```

proc sort data=lstnames;by lstname;run;
data lstnames;
    set lstnames;
    lstorder=_n_;
run;
%end;

```

6. PACK THE FILES AND RETRIEVE TITLES AND FOOTNOTES

First create a new directory or subfolder under the given output file directory or folder. Each file will be read in and redirected into the new temporary location.

When each file is redirected, a special character ‘□’ is added to the first space in the first line of each file, for it is a page separator used by SAS. It breaks pages between different files in the concatenated file. Each redirected file name is renamed by the specified ordering. So an operating system command can concatenate them in order. As reading each file, the macro also computes the page number for both TOC and addition of hyperlinks to the package. The macro retrieves the titles and footnotes from the first page of each file. The title section is determined by two macro parameters TITLSTRT and TITLEND specified by users. 1 is the default for TITLSTRT. The footnote section starts after a long line with at least 120 characters “_” at the bottom of each page. The first page data is reversely sorted and the lines before the long line with at least 120 characters “_” are kept for footnotes. The macro outputs the ls file names if the files are empty. After all files are redirected into the temporary location, Unix or DOS commands are issued using X statements within SAS to concatenate all lst files into a SAS lst file, named _alltbl_1.lst with the specified ordering.

```

X cd;
X cd &outdir;
X mkdir "&tempdir";
%do i=1 %to &numlst;
    data _emptyfl;if 0;run;
    data _temp;
    %if &os=UNIX %then %do;
        infile "%cmpres(&indir./&&lst&i).lst" dsd dlm='09'x;%end;
    %else %if &os=WINDOW %then %do;
        infile "%cmpres(&indir.\&&lst&i).lst" dsd dlm='09'x;%end;
        input lines:$char160.;
        maxlen=160;
        if _n_=1 then lines='_'||lines;
        if _n_=1 then pagetot_=1;
        else if index(trim(left(lines)),'_') then pagetot_+1;
    run;
    *** check if the file is empty;
    %let dsid=%sysfunc(open(_temp));
    %let cnt=%sysfunc(attrn(&dsid,nobs));
    %let rc=%sysfunc(close(&dsid));
    %if &cnt ne 0 %then %do;
        %let ilen=%sysfunc(length(%sysfunc(trim(&i))));
        %let try1=%sysfunc(substr(&tempvar,1,%eval(5-&ilen)));
        %let try2=%sysfunc(trim(&try1))%sysfunc(left(&i));
        data _pagetot(keep=pagetot_);
            set _temp end=eof;

```

```

        %if &os=UNIX %then %do;
            file "&tempdir./_temp_&try2..lst";%end;
        %else %if &os=WINDOW %then %do;
            file "&tempdir.\_temp_&try2..lst";%end;
            put @1 lines $varying. maxlen;
            if eof then output _pagetot;
        run;
        data _pagetot;
            length lstname $15.;
            set _pagetot;
            lstname="&&lst&i";
        run;
        data pagetots;set pagetots _pagetot;run;
    ***retrieving titles of each lst file by reading its first page;
    data _pageone(drop=maxlen);
        length lstname $15.;
        set _temp;
        if _n_=1 then lines=left(translate(lines,','),'_');
        lstname="%scan(%cmpres(&&lst&i),1, ".")";
        lstorder=&i;
        lineodr=_n;
        where pagetot_&=1;
    run;
    data _title(drop=pagetot_);
        length part $9.;
        set _pageone;
        part="titles";
        index=1;
        where lineodr>=&titlstrt and lineodr<=&titlend;
    run;
    data titltots;set titltots _title;run;
    *** retrieve the footnotes;
    proc sort data=_pageone;by lstname descending lineodr;run;
    data _ftnote(drop=pagetot_);
        length part $9.;
        set _pageone;
        by lstname descending lineodr;
        part="footnotes";
        index=2;
        if index(lines,repeat("_", 120))=0 then output;***120 can
                                                    be changed by users;
        else stop;
    run;
    data fotntots;set fotntots _ftnote;run;
%end; /** end of &cnt **/
%else %do;
    data _emptyfl;
        length lstname $15.;
        lstname="&&lst&i";
        cnt=%eval(&cnt);
        output;
    run;
%end;
    data emptyfls;set emptyfls _emptyfl;run;
%end; /** end of %do i=1 %to &numlst **/
%if &os=UNIX %then %do;
    X cd &tempdir;

```

```

        X "cat *.lst > &tempdir./_alltbl_1.lst";
        X cd ..;
%end;
%else %if &os=WINDOW %then %do;
        X cd;X cd &tempdir;
        X copy *.lst _alltbl_1.lst;
        X cd ..;X exit;
%end;

data titlfots;
    set titltots fotntots;
    linesize=length(trim(left(lines)));
    where length(trim(left(lines))>1;
run;

```

7. OUTPUT A SAS DATA SET WITH ALL TITLES AND FOOTNOTES

If requested by users, a SAS data set, named `_altitl_.sas7bdat`, which contains all titles and footnotes with full directory or folder pathname and file names, total page numbers of files, and the sizes of titles and footnotes, will be output to the given output directory or folder (`&outdir`).

```

%if %upcase(&alltittls)=YES %then %do;
    proc sort data=titlfots;by lstname index lineordr;run;
    libname _lib "&outdir";
    data _lib._altitl_;
        retain tabid2;
        length tabid2 $16;
        set titlfots;
        tabid2=trim(left(lstname))||".lst";
        %if &os=UNIX %then %do;
            in_file="&indir"||"/"||trim(left(tabid2));%end;
        %else %if &os=WINDOW %then %do;
            in_file="&indir"||"\ "||trim(left(tabid2));%end;
        label lstname="SAS lst file name"
            lines="titles/footnotes of each lst file"
            index="1=title, 2=footnote"
            lineordr="line no. of each title/footnote in each lst file"
            lstorder="order of lst file"
            pagetot="total page number of each lst file"
            part="title/footnote"
            linesize="number of char. of each title/footnote";
        where index=1;
    run;
%end;

```

8. OUTPUT THE TABLE OF CONTENTS (TOC)

REPORT procedure generates the table of contents (TOC), a SAS lst file, named as `_toc_.lst`. Its maximum number of line size is 156. The length of each title or footnote could be greater than 156, and adding the file names and dotted lines to each title or footnote could make their line sizes greater than 156. Wrapping the titles and footnotes

are necessary. The macro parameter TITLCUT determines the maximum of line size of TOC and 140 is default. If the length of title or footnote is greater than TITLCUT or 140, respectively, and the total length with its next one is less than 2* TITLCUT or 2*140, respectively, its “tail” will be wrapped into its next one; otherwise it will be split into two. Since appendix tables are landscape, splitting one long title or footnote into two is sufficient. If the table of all footnotes is requested, as a SAS lst file, named as _fotnots_.lst, will be generated.

9. APPEND THE CONCATENATED TABLES TO TOC

The Unix or DOS commands are issued using X statements within SAS to concatenate _toc_.lst and _alltbl_1.lst to generate a final review package, named _alltbl_.lst, and to clean up the temporary subdirectory or subfolder.

```
%if %upcase(&tottbls)=YES %then %do;
  %if &os=UNIX %then %do;
    X cd; X cd &outdir; X "cp _toc_.lst &tempdir";X cd;
    X cd &tempdir;
    X cat _toc_.lst _alltbl_1.lst > &outdir/_alltbl_.lst;
    X cd ..; X rm -Rf &tempdir;
  %end;
  %else %if &os=WINDOW %then %do;
    options noxwait noxsync;
    X cd; X cd &outdir;X
    copy _toc_.lst _temp_\_toc_.lst /y;X cd;

    X cd &tempdir; X copy _toc_.lst + _alltbl_1.lst ..\_alltbl_.lst;
    X cd ..;X rmdir _temp_ /s /q;X exit;
  %end;
%end;
```

10. CALL MICROSOFT VBA MACROS

In PC platform, the macro calls Microsoft VBA macros to convert SAS lst files into Microsoft word files, format them, and add hyper-links into the package.

```
%if &os=WINDOW %then %do;
  %_DocGfmt(inputloc=&outdir,_alltbl_=%str(_toc_.lst),
    outputloc=&outdir,ne,w_doc=_toc_);
  %x_myVB(visible=Y,minimize=N,notify=N,exitword=Y);
  %if %upcase(&tottbls)=YES %then %do;
    data _null_;rc=sleep(15);run;
    %_DocGfmt(inputloc=&outdir,_alltbl_=%str(_alltbl_.lst),
      outputloc=&outdir,new_doc=_alltbl_);
    %x_myVB(visible=Y,minimize=N,notify=N,exitword=Y);
    data _null_;rc=sleep(15);run;
    %_TOCparm(DesStart=13,inputloc=&outdir,
      _alltbl_=%str(_alltbl_),outputloc=&outdir);
    %x_myVB(visible=Y,minimize=N,notify=N,exitword=Y);
  %end;
  %if %upcase(&alftnote)=YES %then %do;
```

```

data _null_; rc=sleep(20);run;
%_DocGFmt(inputloc=&outdir,_alltbl_=%str(_fotnots_.lst),
          outputloc=&outdir,new_doc=%str(_fotnots_));
%x_myVB(visible=Y,minimize=N,notify=N,exitword=Y);
%end;
%end;

```

MICROSOFT VBA MACROS

There are three goals to be accomplished with VBA macro programs. First, we format the whole document with a specific publishing format from page set up to font. Second, VBA macros generate links between each entry of Table of Contents (TOC) with specific clinical reports within the document. The last part is to enable a report viewer to go back to where his selected is made in TOC from a report. This becomes convenient and user-friendly as several hundred pages reports and a few dozens of different reports stored in a single document.

11. Document Format

Here are the document format requirements;

- Font and size: Courier 8,
- Line size: 130 letters including a white space between words and symbols,
- Top and bottom margin set: 0.79" and 0.40",
- Left and right margin set: 0.30" equally,
- Page size: letter, or 8.5" x 11",
- Page orientation: landscape.

The following macro program accomplishes the above requirements.

```

Sub FormatThisDocinNDASubmissionFormat()
' Macro: FormatThisDocinNDASubmissionFormat
' Purpose: To select the whole document and format it
'          with NDA submission format.
'
Selection.HomeKey Unit:=wdStory
Selection.EndKey Unit:=wdStory, Extend:=wdExtend
With Selection.Font
    .Name = "Courier New"
    .Size = 8
End With
With ActiveDocument.PageSetup
    .LineNumbering.Active = False
    .Orientation = wdOrientLandscape
    .TopMargin = InchesToPoints(0.79)
    .BottomMargin = InchesToPoints(0.4)
    .LeftMargin = InchesToPoints(0.3)
    .RightMargin = InchesToPoints(0.3)
    .Gutter = InchesToPoints(0)
    .PageWidth = InchesToPoints(11)
    .PageHeight = InchesToPoints(8.5)

```

```
End With
End Sub
```

12. TOC Page Count

TOC is organized as that of a traditional text book. It contains three columns, chapter, a brief description of the chapter and its starting page number of the chapter. Similar to this publishing method, ours has a clinical report file name, a brief description of such a report and its starting page number.

The purpose to count page numbers of TOC is to know which page is the last one to search for a TOC entry. The TOC page count macro read the first line of the first TOC page as a key to recognize all TOC pages till the first line contents change in clinical reports. In the meanwhile, we also need a total page number count of such a Word document as a later reading stop.

Essentially we set the content of the first line of TOC as our search key and compare the value of the next page to the key in order to decide whether or not we have the last page of TOC.

```
Sub TOCPageCount(MyTOCPageCount, TotalPages)
' Macro: TOCTopLine
' Purpose: [1] read the 1st line of TOC,
'          [2] compare the result of Step 1 to that of the following
pages
'          until its difference so that we can find the last page
of TOC.
'          [3] Output of Sub TOCTopLine=PageCount
'
Dim TOCLine As String, MyRange As Range
Dim PageCount As Integer

Selection.GoTo what:=wdGoToBookmark, Name:="\StartOfDoc"
With Selection
    lineEnd = Selection.EndKey(Unit:=wdLine, Extend:=wdMove)
    Selection.HomeKey Unit:=wdLine, Extend:=wdExtend
    Set MyRange = Selection.Range
    TOCLine = Trim(MyRange)
End With
```

We assume TOC has less than 1000 pages. However, our page counting can stop long before hitting this cap. Due to the nature of leading white space ahead of each line in reading, we trim off those and compare values with the rest of non-blank letters and symbols. At the end of this macro, we also gained total page number of a clinical report as a WORD document.

```
For PageCount = 1 To 1000
    Selection.GoTo what:=wdGoToPage, which:=wdGoToNext, count:=1
    With Selection
        lineEnd = Selection.EndKey(Unit:=wdLine, Extend:=wdMove)
        Selection.HomeKey Unit:=wdLine, Extend:=wdMove
        Set MyRange = Selection.Range
```

```

        MyRange.SetRange Start:=MyRange.Start, End:=MyRange.End +
lineEnd
        MyRange.Select
        TOCLineNextPage = Trim(MyRange)
        If TOCLineNextPage <> TOCLine Then Exit For
    End With
    Next PageCount
    MyTOCPageCount = PageCount + 1
    TotalPages = Selection.Information(wdNumberOfPagesInDocument)
End Sub

```

13. Bookmark

The purpose of using bookmark in a Word document in our case is to mark on each of key elements associated with each other. Such a relation between a TOC entry and a specific clinical report is built with these bookmarks.

The bookmark macro parses each of three columns in TOC; filename, file description and page number. A bookmark is made with filename that appears in the first line of each clinical report. We use this filename value as a key to bookmark each clinical report correspondingly.

```

Sub BookmarkWithFilename()
'   Macro: BookmarkwithFilename
'   Purpose: BukMark macro generates a bookmark for each of filename
'           objects at its first page with the same name as the object
'
'   Call TOCTopLine() to get PageCount of TOC

```

We call in our TOCPageCount macro to tell us where the loops on page numbers should stop.

```

Call TOCPageCount(MyTOCPageCount, TotalPages)
'
'   Define Variables
Dim FirstWord As String, FileName(10000) As String
Dim PageNoChar(10000) As String, PageNumber As Integer
Dim MyBkMkTOC As String
'
'   Objective: Go to the first page of TOC.
'   Selection.GoTo what:=wdGoToPage, which:=wdGoToFirst
'
'   Objective: Scan TOC contents line by line.
'   i = 0
Do Until CurPageNumber = MyTOCPageCount

    Selection.HomeKey Unit:=wdLine, Extend:=wdMove
    lineEnd = Selection.EndKey(Unit:=wdLine, Extend:=wdMove)
    Selection.HomeKey Unit:=wdLine, Extend:=wdMove
    Set MyRange = Selection.Range
    MyRange.SetRange Start:=MyRange.Start, End:=MyRange.End +
        lineEnd
    MyRange.Select

```

One of a predefined layout of TOC is the file name column is set between the 1st and 12th columns, and description starting from 13th column. There is at least two white spaces between the description and page number in a TOC entry.

```

FirstWord = Mid(MyRange, 1, 12)
DotStr = InStr(13, MyRange, "...")
If Trim(FirstWord) <> "" And DotStr <= 0 Then
    i = i + 1
    FileName(i) = FirstWord
    MyBkMkTOC = Trim(FileName(i)) & "_TOC"
'
Create Bookmarks (TOC)
    Selection.HomeKey Unit:=wdLine, Extend:=wdMove
    lineEnd = Selection.EndKey(Unit:=wdLine, Extend:=wdExtend)
    Set MyTOCRange = Selection.Range
    MyTOCRange.SetRange Start:=MyTOCRange.Start,
    End:=MyTOCRange.End
    MyTOCRange.Select
    ActiveDocument.Bookmarks.Add Name:=MyBkMkTOC,
    Range:=MyTOCRange
ElseIf Trim(FirstWord) = "" And DotStr > 0 Then
    R1stChr = Right(RTrim(MyRange), 1)
    R1stPos = InStrRev(MyRange, R1stChr)
    R1stBlk = InStrRev(MyRange, " ", R1stPos)
    PageNoChar(i) = Mid(MyRange, R1stBlk, R1stPos)
ElseIf Trim(FirstWord) <> "" And DotStr > 0 Then
    R1stChr = Right(RTrim(MyRange), 1)
    R1stPos = InStrRev(MyRange, R1stChr)
    R1stBlk = InStrRev(MyRange, " ", R1stPos)
    PageNoChar(i) = Mid(MyRange, R1stBlk, R1stPos)
End If

Selection.MoveDown Unit:=wdLine, count:=1
Selection.HomeKey Unit:=wdLine, Extend:=wdMove

CurPageNumber =
    Selection.Information(wdActiveEndAdjustedPageNumber)
Loop

```

A predefined bookmark by VBA 'StartOfDoc' is applied to move our reading pointer to the very beginning of the document.

```

' Create Bookmarks
Selection.GoTo what:=wdGoToBookmark, Name:="\StartOfDoc"
For j = 1 To i
    BkMkPg = Trim(PageNoChar(j))
    MyBkMk = Trim(FileName(j))
    Selection.GoTo what:=wdGoToPage, which:=wdGoToAbsolute,
    Name:=BkMkPg
    ActiveDocument.Bookmarks.Add Name:=MyBkMk,
    Range:=Selection.Range
Next j
TotalFiles = j

' Create hyperlink from 1st title line of tables/listings back to TOC
'
Call Table2TOC(MyTOCPageCount, TotalPages, FileName, _

```

```

PageNoChar, TotalFiles)
End Sub

```

14. Build A Link from A TOC Entry to A Clinical Report

Having generated bookmarks, we will have them be linked to each other. At the end of this program, we should have a link from TOC entry to the starting page of its related clinical report within the document.

The macro, TOC2Table, links a TOC entry to its corresponding clinical report. The bookmark that has been generated in Step 13 is used as a key for the hyperlink between each report and TOC entry.

```

Sub TOC2Table(DesStart)
'
'   Macro: TOC2Table
'   Purpose: link TOC to the first page of Tables/Listing.
'   Style: Hyperlink to Bookmark
'
Call BookmarkWithFilename
Call TOCPageCount(MyTOCPageCount, TotalPages)
'
'   Define Variables
UpBound = TotalPages
Dim FirstWord As String, DesPhrase As String
Dim FileName(10000) As String, PageNoChar(10000) As String
Dim PageNumber As Integer
'
'   Objective: Go to the first page of TOC.
'   Selection.GoTo what:=wdGoToBookmark, Name:="\StartOfDoc"
'
'   Objective: Scan TOC contents line by line.
'   i = 0
Do Until CurPageNumber = MyTOCPageCount

    Selection.HomeKey Unit:=wdLine, Extend:=wdMove
    lineEnd = Selection.EndKey(Unit:=wdLine, Extend:=wdMove)
    Selection.HomeKey Unit:=wdLine, Extend:=wdMove
    Set MyRange = Selection.Range
    MyRange.SetRange Start:=MyRange.Start, End:=MyRange.End +
        lineEnd
    MyRange.Select

    MyRangeLen = Len(MyRange)
    FirstWord = Mid(MyRange, 1, DesStart - 1)
    DotStr = InStr(DesStart, MyRange, "..")

    If Trim(FirstWord) <> "" And DotStr <= 0 Then
        i = i + 1
        FileName(i) = FirstWord
        DesPhrase = Mid(MyRange, DesStart, MyRangeLen - DesStart +
            1)
        DesPhrase = Trim(DesPhrase)
    End If
End Do

```

```

Set MyHyperRange = Selection.Range
MyHyperRange.SetRange Start:=MyHyperRange.Start + DesStart-
1, End:=MyHyperRange.End
MyHyperRange.Select

```

Essentially, a hyperlink is created to associate bookmarks between a TOC entry and the clinical report. A 'screen tip' can be created as desired for a hint to the file name as a mouse point covering on the spot. Since our TOC description next to the file name is good enough to fulfill this purpose, a screen tip is skipped as a blank.

```

ActiveDocument.Hyperlinks.Add Anchor:=MyHyperRange,
Address:="", _
SubAddress:=Trim(FileName(i)), ScreenTip:="",
TextToDisplay:=DesPhrase
ElseIf Trim(FirstWord) = "" And DotStr > 0 Then
DesPhrase = Trim(MyRange)

Set MyHyperRange = Selection.Range
MyHyperRange.SetRange Start:=MyHyperRange.Start + DesStart -
1, End:=MyHyperRange.End
MyHyperRange.Select
ActiveDocument.Hyperlinks.Add Anchor:=MyHyperRange,
Address:="", _
SubAddress:=Trim(FileName(i)), ScreenTip:="",
TextToDisplay:=DesPhrase
ElseIf Trim(FirstWord) <> "" And DotStr > 0 Then
i = i + 1
FileName(i) = FirstWord
DesPhrase = Mid(MyRange, DesStart, Len(MyRange) - DesStart +
1)
DesPhrase = Trim(DesPhrase)

Set MyHyperRange = Selection.Range
MyHyperRange.SetRange Start:=MyHyperRange.Start + DesStart -
1, End:=MyHyperRange.End
MyHyperRange.Select

ActiveDocument.Hyperlinks.Add Anchor:=MyHyperRange,
Address:="", _
SubAddress:=Trim(FileName(i)), ScreenTip:="",
TextToDisplay:=DesPhrase
End If
'If i <= 3 Then
' MsgBox (DesPhrase)
'End If
Selection.MoveDown Unit:=wdLine, count:=1
Selection.HomeKey Unit:=wdLine, Extend:=wdMove
CurPageNumber = _
Selection.Information(wdActiveEndAdjustedPageNumber)
Loop
End Sub

```

15. A Recursive Link Back to TOC

The last VBA macro can point our cursor from a TOC entry to its associated clinical report as clicking on its description. As some of compiled clinical reports could well

contain thousand pages in a single Word document, a link from a specific clinical report back to the selection made in TOC is very helpful for a clinical reviewer.

The macro, Table2TOC, is to fulfill this user friendly attribute by linking a clinical report back to the TOC entry where a reviewer makes a selection. Once again, the bookmark is used as a key for the hyperlink between each report and TOC entry.

```
Sub Table2TOC(MyTOCPageCount, TotalPages, FileName, PageNoChar,
TotalFiles)
'
' Macro: TOC2Table
' Purpose: link TOC to the first page of Tables/Listing.
' Style: Hyperlink to Bookmark
' Selection.GoTo what:=wdGoToBookmark, Name:="\StartOfDoc"
' Selection.GoTo what:=wdGoToPage, count:=MyTOCPageCount - 1
'
' Objective: Loop within TotalPages
' For k = 1 To TotalFiles - 1
' Goto bookmark(k)
'   LocalBkMK = Trim(FileName(k))
'
'   Selection.GoTo what:=wdGoToBookmark, Name:=LocalBkMK
'
'   NextBkMkPage = Val(PageNoChar(k + 1)) ' if NextBkMkPage=0 then
STOP!
'   Do Until (CurPageNum = NextBkMkPage Or CurPageNum = TotalPages)
'
'   Read the first title line
'     Selection.HomeKey Unit:=wdLine, Extend:=wdMove
'     Selection.EndKey Unit:=wdLine, Extend:=wdExtend
'     Set MyTopRange = Selection.Range
'
'   Hyperlink the title line back to TOC
'     MyTOCTag = Trim(FileName(k)) & "_TOC"
'     ActiveDocument.Hyperlinks.Add Anchor:=Selection.Range,
Address:="", _
'     SubAddress:=MyTOCTag, ScreenTip:="",
TextToDisplay:=MyTopRange
'   Go To Relative Next Page
'     Selection.GoTo what:=wdGoToPage, which:=wdGoToNext,
count:=1
'   Test if current page number equals to PageNoChar(k+1)
'     CurPageNum =
Selection.Information(wdActiveEndAdjustedPageNumber)
'     Loop
'     CurPageNum =
Selection.Information(wdActiveEndAdjustedPageNumber)
'
'   If CurPageNum = TotalPages Then
'     Selection.HomeKey Unit:=wdLine, Extend:=wdMove
'     Selection.EndKey Unit:=wdLine, Extend:=wdExtend
'     Set MyTopRange = Selection.Range
'   Hyperlink the title line back to TOC
'     MyTOCTag = Trim(FileName(k)) & "_TOC"
```

```

        ActiveDocument.Hyperlinks.Add Anchor:=Selection.Range,
Address:="", _
        SubAddress:=MyTOCTag, ScreenTip="",
TextToDisplay:=MyTopRange
        Exit For
    End If
Next k
End Sub

```

16. A Customized Hyperlink Style

A conventional hyperlink style is to be customized since our font size is so small that an underline makes TOC entry hard to read for a reviewer. This VBA macro is to remove a predefined underline from hyperlink style. However, we maintain the conventional style in color selections for unread and read.

```

Sub HyperStyle()
'
' Change default Hyperlink style. This can be applied at the end of
Hyperlink creation
' [1] no underline
' [2] no font change at current request
'
    ActiveDocument.Styles("Hyperlink").BaseStyle = "Default Paragraph
Font"
    With ActiveDocument.Styles("Hyperlink").Font
        .Underline = wdUnderlineNone
        .Color = wdColorBlue
    End With
    Selection.Style = ActiveDocument.Styles("Hyperlink")

    ActiveDocument.Styles("FollowedHyperlink").BaseStyle = "Default
Paragraph Font"
    With ActiveDocument.Styles("FollowedHyperlink").Font
        .Underline = wdUnderlineNone
        .Color = wdColorViolet
    End With
    Selection.Style = ActiveDocument.Styles("FollowedHyperlink")
End Sub

Sub MainBuilding()
' Predefined parameters
' DesStart = 13
'
' Open targeted list file _alltable_.txt
' Documents.Open
FileName:="C:\SASEnv\TSD\STUDY_ISS\Doc\_alltbl_.doc", _
' ReadOnly:=True
'
' Build Hyperlink with Bookmarks and Filenames
'
    Call HyperStyle
    Call TOC2Table(DesStart)
    Selection.GoTo what:=wdGoToBookmark, Name:="\StartOfDoc"

' ActiveDocument.SaveAs
FileName:="C:\SASEnv\TSD\STUDY_ISS\Doc\_alltbl_lnkdoc.doc"

```

```
' Documents.Close SaveChanges:=wdSaveChanges
' Application.Quit SaveChanges:=wdPromptToSaveChanges
End Sub
```

CONCLUSION

In summary, the SAS macro “%mgettoc” with VBA macros can be used to automate the process of compiling the review package with hyper-linked TOC from SAS outputs, eliminating the traditional typing /copying the titles and manual assigning the page numbers and bookmarks. It also provides a table of all footnotes for easy reference and a SAS data sets with all titles and footnotes for users. The macro is easy to use and outputs a series of appropriate messages to help the user to track how the SAS lst files were handled.

REFERENCES

“Using SAS Software and Visual Basic for Applications to Automate Tasks in Microsoft Word: An Alternative to Dynamic Data Exchange”, Mark Stetz, Amgen Inc., SUGI 29.
“Fancy MS Word Reports Made Easy: Harnessing the Power of Dynamic Data Exchange – Against All ODS, Part II –”, William Viergever and Koen Vyverman, SUGI 28.

ACKNOWLEDGEMENTS

Appreciation goes to Cindy Song for her review and comments.

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 registered trademarks or trademarks of their respective companies.

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

Wei Zhang, M.S.

Independent Contractor
weizhang.nj.us@gmail.com

APPENDIX A: TABLE OF CONTENTS OUTPUT SAMPLE

STUDY123/ISS: Table of Contents

acA02a1t	Previous/ongoing diseases (all treatments) - safety population [US] controlled 12-week efficacy and safety studies in adults and adolescents	5
acA03r1t	Patients completion status (all treatments) - all randomized patients [US] controlled 12-week efficacy and safety studies in adults and adolescents	6
acA06i1t	Concomitant asthma medications received during double-blind treatment period (all treatments) - ITT population [US] controlled 12-week efficacy and safety studies in adults and adolescents	7
acA07s1t	Summary of treatment duration (double-blind) in days (all treatments) - safety population [US] controlled 12-week efficacy and safety studies in adults and adolescents	9
acA20s1t	Summary of treatment duration (double-blind) in days by age group (all treatments) - safety population [US] controlled 12-week efficacy and safety studies in adults and adolescents	10
acA21s1t	Summary of treatment duration (double-blind) in days by gender (all treatments) - safety population [US] controlled 12-week efficacy and safety studies in adults and adolescents	13
acA22s1t	Summary of treatment duration (double-blind) in days by race (all treatments) - safety population [US] controlled 12-week efficacy and safety studies in adults and adolescents	15
aeA27s1t	Overview of TEAEs - safety population [US] controlled 12-week efficacy and safety studies in adults and adolescents	18
aeA01s1t	All TEAEs by system organ class (all treatments) - safety population [US] controlled 12-week efficacy and safety studies in adults and adolescents	19
aeA01s2t	Possibly related TEAEs by system organ class (all treatments) - safety population [US] controlled 12-week efficacy and safety studies in adults and adolescents	35
aeA01s9t	Most common TEAEs (>= 3%) by system organ class (all treatments) - safety population [US] controlled 12-week efficacy and safety studies in adults and adolescents	38
aeA02s1t	All TEAEs by decreasing frequency (all treatments) - safety population [US] controlled 12-week efficacy and safety studies in adults and adolescents	39
aeA02s2t	Possibly related TEAEs by decreasing frequency (all treatments) - safety population [US] controlled 12-week efficacy and safety studies in adults and adolescents	53
aeA03s3t	All TEAEs by system organ class and intensity (all treatments) - safety population [US] controlled 12-week efficacy and safety studies in adults and adolescents	56