

Visual Basic for Applications to Produce Titles and Footers from Shell Paragraphs

Fan Cui, Pharmaron (Chengdu) Clinical Services Co., Ltd.

ABSTRACT

This article introduces a way for using Word Visual Basics for Applications (Word VBA) to extract the titles and footnotes of tables, listings, and figures (TLFs) from paragraph properties within the shell. The approach is to loop over each paragraph, report its text, and then verify if it is a title or a footnote depending on whether the text matches the title's regular expression. The next step is to utilize SAS to deal with these titles and footnotes in order to give metadata for later applications of automatically adding titles and footers for each TLF. Concurrently, the tool will also produce a tracker file.

When compared to the manual procedure of producing TLF titles and footnotes, which involves copying those from the shell one by one for hundreds of TLFs, programmatic processing has significantly reduced time and increased efficiency.

INTRODUCTION

A new drug needs to pass clinical trials and provide meaningful data which should be analyzed seriously according to the Statistical Analysis Plan (SAP) before it can reach the market. Via TLFs defined in a shell, another crucial component of SAP, the analysis results visually demonstrate the drug's safety and effectiveness.

Preparing titles and footnotes for TLFs with the shell can be challenging given the large number of TLFs that require these. Thus Goldfarb^[1] developed a VBA macro mSHELLvsTLF to obtain the title and footnotes of the shell document, which should be prepared in a specific way (left of Figure 1.). However, our shell (right of Figure 1.) does not match this criteria since titles and footnotes are usual text, as illustrated in Figure 1. Adler^[2] created a tool to get titles and footnotes from the TLF shell through VBA in which titles and footnotes are usual text. Unfortunately, line breaks (SHIFT+ENTER) are required for every line in the title section and TLFs are broken down by section page breaks instead of regular page breaks. Compared to our work with accessible VBA, Zhang^[3] obtained titles and footnotes using Python, and their principle was similar to ours, requiring a clean shell. As previously stated, our VBA macro does not have strict control over shell formatting, and no further software is required to get titles and footnotes.

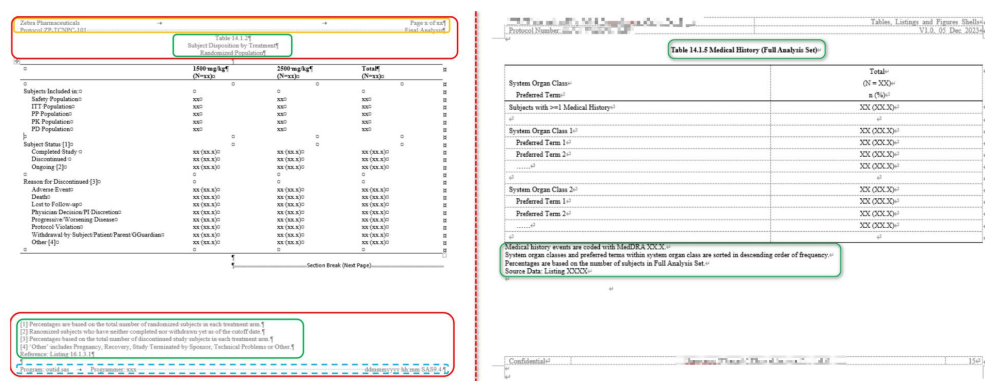


Figure 1. The contrast between Goldfarb's shell requirements (left) and our shell (right)

In brief, we developed a Word VBA macro that loops every paragraph to extract the title and footnote text and then automatically outputs to Excel metadata files in which titles and footnotes are added to each TLF. Besides output to Excel metadata files by SAS, in this step, we also perform a TLFs check to confirm if a TLF is ignored or the title format is appropriate (there should be a blank space between the title id and title name in TLFs) as well as producing a tracker file. The all steps are shown in Figure 2.

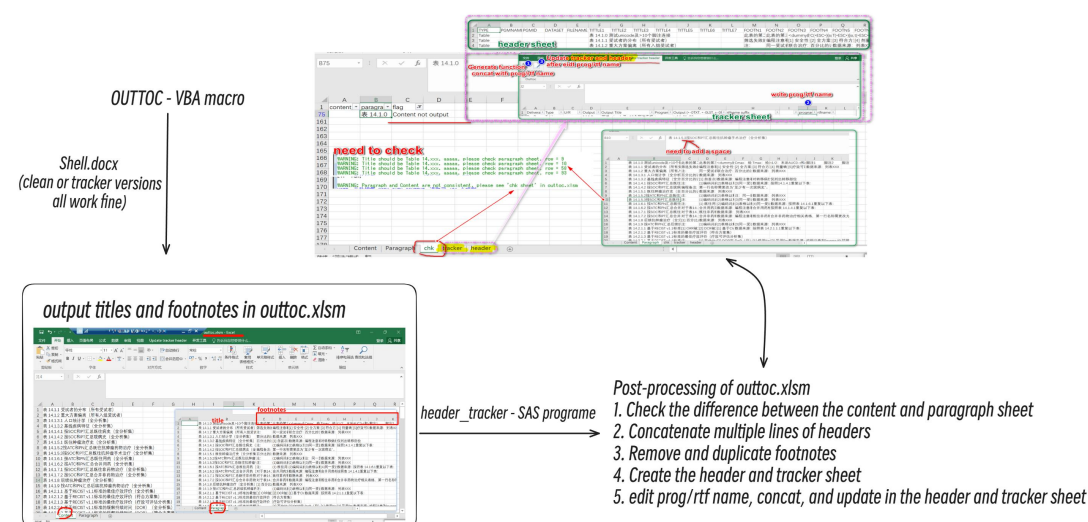


Figure 2. Steps of this tool.

MATERIAL AND METHOD

FORMAT SPECIFICATION FOR SHELL

- (1) It is ideal but not necessary that all the footnotes for TLFs are written rather than presented as “xx footnotes will be updated as xxxxxxxx” or “Update as Table 14.xx.xx.xxx”.
- (2) No texts should be automatically numbered.

WORD VBA – GET HEADER AND FOOTER TEXT FROM SHELL

The whole code is detailed in Appendix A at the end of the article. First, we use the OpenFileDialog to choose a shell file, accepting modifications and removing comments if it is the tracked version file.

```

'Open word file window
Set myDialog = Application.FileDialog(msoFileDialogFilePicker)
With myDialog
    .Filters.Clear
    .Filters.Add "Word File", "*.doc;*.docx", 1
    .AllowMultiSelect = False
    .InitialFileName = ThisDocument.Path & ""
    If .Show = -1 Then
        docpath = .SelectedItems(1)
    End If
Set oDoc = Documents.Open(docpath)
Else
    MsgBox "Open Shell File。 "
Exit Sub
End If
End With
Word.Application.DisplayAlerts = False
Word.Application.ScreenUpdating = False

' Delete comments and accept all revisions
oDoc.AcceptAllRevisions
oDoc.TrackRevisions = False
If oDoc.Comments.Count > 0 Then oDoc.DeleteAllComments

```

Second, we extract text by iterating over paragraphs, its text will be lost if it occurs after a section break and a page break on the same line. So in lieu of a section or page break, we'll use a paragraph break. In addition, while iterating over paragraphs, "xx" table texts may be extracted too, which is undesirable. Hence, we will remove tables as needed.

```

' Loop for each section
Dim orng As Range
Set orng = oDoc.Content
With orng.Find
    .ClearFormatting
    .MatchWildcards = False
    'Replace page break to paragraph break
    .Text = "^m"
    .Execute ReplaceWith:="^p", Replace:=wdReplaceAll
    'Replace section break to paragraph break
    .Text = "^b"
    .Execute ReplaceWith:="^p", Replace:=wdReplaceAll
End With

```

```

' Delete tables
Dim tbl As Table
For Each tbl In oDoc.Tables
    tbl.Delete
Next tbl

```

Then we will update the table of contents (TOC) for comparison of titles generated from followed looping paragraphs and delete the TOC later.

```

' Update TOC
With ActiveDocument
    If .TablesOfContents.Count >= 1 Then .TablesOfContents(1).Delete
    Selection.GoTo What:=wdGoToParagraph, Which:=wdGoToPrevious, Count:=1, Name:=""
    .TablesOfContents.Add Range:=Selection.Range, RightAlignPageNumbers:=_
        True, UseHeadingStyles:=True, UpperHeadingLevel:=1, _
        LowerHeadingLevel:=3, IncludePageNumbers:=False, AddedStyles:="", _
        UseHyperlinks:=True, HidePageNumbersInWeb:=True, UseOutlineLevels:=_
        True
    .TablesOfContents(1).TabLeader = wdTabLeaderDots
    .TablesOfContents.Format = wdIndexIndent
End With

'Output TOC content to column A
contr = 1
Set tocrange = oDoc.TablesOfContents(1).Range
For Each contpg In tocrange.Paragraphs
    contstr = contpg.Range.Text
    ctsheet.Range("A" & contr) = (Trim(Replace(Replace(Replace(Replace(Replace(contstr,
Chr(10), vbNullString), Chr(13), vbNullString), Chr(12), vbNullString), Chr(11), vbNullString), Chr(9),
vbNullString))))
    contr = contr + 1
Next contpg

' Delete TOC contents
oDoc.TablesOfContents(1).Delete

```

Finally, we loop over each paragraph, extracting paragraph text by paragraph.Range.Text property. However, despite the fact that the line break utilized by biostatisticians as a paragraph break is visually identical, it is not classified as establishing a new paragraph in this attribute. As a result, we substitute line breaks with paragraph breaks and generate paragraph texts that may be recognized as titles or footnotes if they match the title's regular expression. As a consequence, we have a spreadsheet with rows for one TLF, column B for the title, and others for footnotes.

```
'Replace line break to paragraph break
```

```
With oDoc.Range.Find
```

```
.ClearFormatting
```

```
.MatchWildcards = False
```

```
.Text = "^I"
```

```
.Replacement.Text = "p"
```

```
.Execute Replace:=wdReplaceAll
```

```
End With
```

```
Dim regEx As Object
```

```
Set regEx = CreateObject("VBScript.RegExp")
```

```
regEx.Pattern = "^(( 表 ([\s])?14\.)|( 表格 ([\s])?14\.)|( 列表 ([\s])?16\.)|( 图 ([\s])?14\.)|( 图表 ([\s])?14\.)|(Table 14\.)|(Listing 16\.)|(Figure 14\.))(.*)(([\^: :])$)"
```

```
'Loop for title and footer
```

```
i = 0
```

```
clrow = 1
```

```
For Each pg In oDoc.Paragraphs
```

```
pgtext = Trim(Replace(Replace(Replace(Replace(Replace(pg.Range.Text, Chr(10), vbNullString), Chr(13), vbNullString), Chr(12), vbNullString), Chr(11), vbNullString), Chr(9), vbNullString))
```

```
' Excute regular expression
```

```
match = regEx.Test(pgtext)
```

```
If match And pgtext <> vbNullString Then
```

```
i = i + 1
```

```
col = 2
```

```
xlsheet.Cells(i, col).Value = pgtext
```

```
Elseif Not (match) And pgtext <> vbNullString Then
```

```
col = col + 1
```

```
xlsheet.Cells(i, col).Value = pgtext
```

```
End If
```

```
pgtext = ""
```

```
Next
```

Now we have a spreadsheet of all titles and footnotes texts. Some footnotes, such as "programming note: xxxxx", should not be generated as well as texts need to be encoded. Taking those into account, we use SAS to handle titles and footnotes: 1) specifying lines of TLFs titles through SAS macro variables titlen, ftitlen, and ltitlen or writing in column A of paragraph sheet 2) Unicode implementation through SAS macro variables unicode_from and unicode_to 3) deleting footnotes (SAS macro variable dele_txt for deletion of footnotes that meet the regular rules; macro variable prg_note for deletion of programming notes and footnotes after it) 4) repeating footnotes through SAS macro variable repeat_txt 5) generating tracker file and header metadata. 6) Confirming titles are consistent between content output and paragraph looping. The code is displayed in Appendix B.

DISCUSSION

If TLF titles are not regular, the title regex should be modified. The regex matches title texts, whereas other texts belong to footnotes. The Word VBA macro may extract all text from the shell, and the procedure is more efficient if all footnotes are included. In some scenes, the SAS program will produce superfluous duplications for some TLFs without footnotes which the preceding footnote is displayed as “repeat from the table as 14.xxx”. Additionally, deleting

footnotes can be well implemented in the SAS program, including deleting programming notes and subsequent text, deleting a single line of text such as "<Labs parameter: xx>".

REFERENCES

- [1] Goldfarb, I., and Zelichonok, E, (2021), Macro to Compare Titles and Footnotes in Produced TLF and Corresponding Shells, Proceedings of the PharmaSUG 2021, Paper AD-179.
- [2] Adler, J., and Abushouk, A, (2022), Optimizing TLF Generation: Titles and Footnotes Applying a New Idea to a Basic Approach, Proceedings of the PharmaSUG 2022, Paper AD-145.
- [3] Zhang, W., (2022), Extracting Titles and Footnotes from TLF SHELL with PYTHON, Proceedings of the PharmaSUG 2022, Paper AD-115

ACKNOWLEDGMENTS

I am grateful to my line manager Shuangquan, manager Xuhua, and other colleagues for their assistance and helpful recommendations.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please feel free to contact me at:

Author Name: Fan Cui

Company: Pharmaron (Chengdu) Clinical Services Co., Ltd

Email: fan.cui@pharmaron.com

APPENDIX

APPENDIX A – WORD VBA SCRIPT TO GENERATE SHELL'S TITLES AND FOOTNOTES TEXT

```
Sub GetHeadFooter()  
    Dim pg As Paragraph, contpg As Paragraph  
    Dim r As Range, tocrange As Range  
    Dim i As Integer, contr As Integer  
    Dim col As Integer  
    Dim filepath, docpath, constr, pgtext, pfilepath As String  
    Dim xlApp As Object  
    Dim xlbook, xlbook2 As Object  
    Dim xlsheet, ctsheet, xlsheet2 As Object  
    Dim myDialog As FileDialog  
  
    'Open word file window  
    Set myDialog = Application.FileDialog(msoFileDialogFilePicker)  
    With myDialog  
        .Filters.Clear  
        .Filters.Add "Word File", "*.doc;*.docx", 1  
        .AllowMultiSelect = False  
        .InitialFileName = ThisDocument.Path & "\"  
        If .Show = -1 Then  
            docpath = .SelectedItems(1)  
  
            Set oDoc = Documents.Open(docpath)  
        Else  
            MsgBox "Open Shell File。 "  
            Exit Sub  
        End If  
    End With  
  
    Word.Application.DisplayAlerts = False  
    Word.Application.ScreenUpdating = False  
  
    ' Delete comments and accept all revisions  
    oDoc.AcceptAllRevisions  
    oDoc.TrackRevisions = False  
    If oDoc.Comments.Count > 0 Then oDoc.DeleteAllComments  
  
    ' Loop for each section  
    Dim orng As Range  
    Set orng = oDoc.Content
```

```

With orng.Find
    .ClearFormatting
    .MatchWildcards = False
    ' Replace page break to paragraph break
    .Text = "^m"
    .Execute ReplaceWith:="^p", Replace:=wdReplaceAll

    ' Replace section break to paragraph break
    .Text = "^b"
    .Execute ReplaceWith:="^p", Replace:=wdReplaceAll

End With

' Delete tables
Dim tbl As Table

For Each tbl In oDoc.Tables
    tbl.Delete
Next tbl

' Update TOC
With ActiveDocument
    If .TablesOfContents.Count >= 1 Then .TablesOfContents(1).Delete

    Selection.GoTo What:=wdGoToParagraph, Which:=wdGoToPrevious, Count:=1, Name:=""

    .TablesOfContents.Add Range:=Selection.Range, RightAlignPageNumbers:= _
        True, UseHeadingStyles:=True, UpperHeadingLevel:=1, _
        LowerHeadingLevel:=3, IncludePageNumbers:=False, AddedStyles:= "", _
        UseHyperlinks:=True, HidePageNumbersInWeb:=True, UseOutlineLevels:= _
        True

    .TablesOfContents(1).TabLeader = wdTabLeaderDots
    .TablesOfContents.Format = wdIndexIndent
End With

' -----
' -----Get TOC Content-----
' -----

filepath = oDoc.Path & "\" & "outtoc.xlsm"

Set xlApp = CreateObject("Excel.Application")
Set xlbook = xlApp.Workbooks.Open(filepath)
Set ws = xlbook.Sheets.Add(After:=xlbook.Sheets(xlbook.Sheets.Count))

```

```

'Delete other sheets
Do While xlbook.Sheets.Count > 1
    xlApp.DisplayAlerts = False
    xlbook.Sheets(1).Delete
Loop

Set ctsheet = xlbook.Sheets(1)
ctsheet.Select
ctsheet.Range("A:Z").ClearContents

contr = 1

'Output TOC content to column A

Set tocrange = oDoc.TablesOfContents(1).Range

For Each contpg In tocrange.Paragraphs
    contstr = contpg.Range.Text
    ctsheet.Range("A" & contr) = (Trim(Replace(Replace(Replace(Replace(Replace(contstr,
Chr(10), vbNullString), Chr(13), vbNullString), Chr(12), vbNullString), Chr(11), vbNullString), Chr(9),
vbNullString))))
    contr = contr + 1
Next contpg

' Delete TOC contents
oDoc.TablesOfContents(1).Delete

' -----
' -----Get Header & footer-----
' -----

Set ws = xlbook.Sheets.Add(After:=xlbook.Sheets(xlbook.Sheets.Count))
'Set regular expression for use
Set xlsheet = xlbook.Sheets(2)
xlsheet.Select
Dim regEx As Object
Set regEx = CreateObject("VBScript.RegExp")
regEx.Pattern = "^(( 表 ([\s])?14\.)( 表格 ([\s])?14\.)( 列表 ([\s])?16\.)( 图 ([\s])?14\.)( 图表 ([\s])?14\.)(Table 14\.)(Listing 16\.)(Figure 14\.))(.*)(([^: :])$)"

For Each pg In oDoc.Paragraphs
    pgtext = pg.Range.Text

```

```

' Execute regular expression
Dim match As Boolean
match = regEx.Test(Trim(Replace(Replace(Replace(Replace(Replace(pgtext, Chr(10),
vbNullString), Chr(13), vbNullString), Chr(12), vbNullString), Chr(11), vbNullString), Chr(9),
vbNullString))))

```

```

' Delete undesired paragraph
If Not match Then pg.Range.Delete
If match Then Exit For
Next

```

```

' Clear style and texts of column A and B
xlsheet.Range("A:AZ").ClearContents
xlsheet.Range("A1").Select

```

```

' Replace line break to paragraph break
With oDoc.Range.Find
    .ClearFormatting
    .MatchWildcards = False
    .Text = "^l"
    .Replacement.Text = "^p"
    .Execute Replace:=wdReplaceAll
End With

```

```

'List font style
Dim sty As Style
Dim style_normal As String
style_normal = ""
For Each sty In ActiveDocument.Styles
    If sty.NameLocal = "正文" Or sty.NameLocal = "Normal" Then
        style_normal = sty.NameLocal
    Exit For
End If
Next sty

```

```

' Output all paragraph in a file when loop

```

```

pfilepath = oDoc.Path & "\" & "_paragraph.xlsx"
If Dir(pfilepath) = "" Then
    Set xlApp = CreateObject("Excel.Application")
    Set xlbook2 = xlApp.Workbooks.Add
    xlbook2.SaveAs pfilepath

```

```

xlbook2.Close
Set xlbook2 = Nothing
Set xlApp = Nothing
End If

```

```

Set xlApp = CreateObject("Excel.Application")
Set xlbook2 = xlApp.Workbooks.Open(pfilepath)
Set xlsheet2 = xlbook2.Worksheets(1)
xlsheet2.Range("A:B").Font.Color = Black
xlsheet2.Range("A:B").ClearContents
xlsheet2.Range("A1").Select

```

```

xlbook2.Worksheets(1).Cells(1, 1).Value = "Text"
Set rng = xlbook2.Worksheets(1).Range("A:A")
rng.AutoFilter

```

```

'Loop for title and footer

```

```

i = 0
clrow = 1

```

```

For Each pg In oDoc.Paragraphs

```

```

    pgtext = Trim(Replace(Replace(Replace(Replace(Replace(pg.Range.Text, Chr(10),
vbNullString), Chr(13), vbNullString), Chr(12), vbNullString), Chr(11), vbNullString), Chr(9), vbNullString))
    pg.Range.Style = oDoc.Styles(style_normal)

```

```

' Excute regular expression

```

```

match = regEx.Test(pgtext)

```

```

If match And pgtext <> vbNullString Then

```

```

    i = i + 1

```

```

    col = 2

```

```

    xlsheet.Cells(i, col).Value = pgtext

```

```

Elseif Not (match) And pgtext <> vbNullString Then

```

```

    col = col + 1

```

```

    xlsheet.Cells(i, col).Value = pgtext

```

```

End If

```

```

If pgtext <> vbNullString Then

```

```

    clrow = clrow + 1

```

```

    xlbook2.Worksheets(1).Cells(clrow, 1).Value = pgtext

```

End If

pgtext = ""

Next

xlbook.Sheets(1).Name = "Content"

xlbook.Sheets(2).Name = "Paragraph"

xlbook.Save

xlbook.Close

xlbook2.Save

xlbook2.Close

Set xlbook = Nothing

Set xlbook2 = Nothing

Set xlApp = Nothing

oDoc.Close SaveChanges:=wdDoNotSaveChanges

MsgBox "Output to outtoc.xlsm"

End Sub

APPENDIX B – SAS CODE TO HABLE FOOTNOTES AND TITLES

```
* -----
* *****Step 01 set up parameters*****
* -----;
***** parameters are requied and can't be null *****;
/*for title*/
%let ttiten=1; /*lines of table title*/
%let ftiten=1; /*lines of figure title*/
%let ltiten=1; /*lines of listing title*/

/*for unicode*/
%let unicode_from=%nrstr(
%|
&|
t1/2|
Cmax|
Tmax|
AUC0-t|
AUC0-inf|
λz|
```

```
Vz||||
89Zr||||
Zr89||||
kg/m2||||
kg/cm2||||
[Height (cm)/100]2);
```

```
%let unicode_to=%str(
(*ESC*){unicode 0025}||||
(*ESC*){unicode 0026}||||
t(*ESC*){sub 1/2}||||
C(*ESC*){sub max}||||
T(*ESC*){sub max}||||
AUC(*ESC*){sub 0-t}||||
AUC(*ESC*){sub 0-inf}||||
(*ESC*){unicode 03BB}(*ESC*){sub z}||||
V(*ESC*){sub z}||||
(*ESC*){super 89}Zr||||
(*ESC*){super 89}Zr||||
kg/m(*ESC*){super 2}||||
kg/cm(*ESC*){super 2}||||
[Height (cm)/100](*ESC*){super 2});
```

```
***** parameters are requied and can be null, write in regular expression format if assigned*****,
/*delete footnotes that match regular expression*/
/*e.g: %let dele_txt=%str((^(<)(.*)>)$)|重复以下|REPEAT THE FOLLOWING|REPEAT FOR|重复第二阶
段|重复下图); */
%let dele_txt=%str(^(<)(.*)>)$|注: );
```

```
/*delete programming note and others after it*/
/*e.g: %let prg_note=%str(编程备注|编程注意|programming note); */
%let prg_note=%str(编程);
```

```
/*for repeat note*/
/*e.g: %let repeat_txt=%str(重复以下|重复下图);*/
%let repeat_txt=%str(重复以下表);
```

```
* -----
* *****Step 02 start programming*****
* -----;
```

```
*****Automatically finding the study folders*****;
```

```
%macro rootpath;
%global _folder _rootpath ;
```

```

%local _fpath;

%let _fpath=%sysfunc(getoption(sysin));

%if %length(&_fpath)=0 %then %do;
    %let _fpath=%sysget(SAS_EXECFILEPATH);
%end;

data _null_;
    _folder=substr("&_fpath.",1,find("&_fpath.", "\",-200)-1);
    _rootpath=cats(substr(_folder,1,find(_folder,"csr","i")), "sr");
    call symputx('_folder',_folder);
    call symputx('_rootpath',_rootpath);
run;
%mend rootpath;

%rootpath;

proc datasets lib=work mt=data kill nolist nowarn;
quit;

proc format;
    value $typee
        "t"="Table"
        "l"="Listing"
        "f"="Figure";
run;

*****import excel*****;
proc import datafile="&_folder.\outtoc.xlsx" out=content dbms=excel replace;
    getnames=n;
    sheet="Content";
run;

proc import datafile="&_folder.\outtoc.xlsx" out=paragraph dbms=excel replace;
    getnames=n;
    sheet="Paragraph";
run;

*****content and paragraph compare*****;
data content ;
    length f1 $2000 titleid $200;
    set content(where=(f1 ne "")) ;
    format _all_;

```

```

informat _all_;
f1=tranwrd(tranwrd(tranwrd(f1,"表14.","表 14."),"列表16.","列表 16."),"图14.","图 14.");
f1=tranwrd(tranwrd(tranwrd(f1,"表格14.","表格 14."),"列表16.","列表 16."),"图表14.","图表
14.");
if prxmatch("/^(表[格]*|列表|图[表]*|Table|Listing|Figure)( )(1[46]\.)([^\d.]+)([a-zA-Z\d\.]*) ( )/",f1)
then do;
titleid=cats(substr(compbl(f1),1,find(compbl(f1)," ",find(compbl(f1),"")+1)) );
end;

if missing(titleid) then titleid=f1;

run;

data paragraph;
retain sum 0;
set paragraph end=lst;
f1fl=input(cats(f1),??best.)>;
sum=sum+f1fl;
if lst then do;
call symputx('nlinewt',sum);
end;
drop lst f1fl sum;

run;

%put &nlinewt.;

%macro parout;
data paragraph;
format _all_;
informat _all_;
length a $8 b $1000 titleid $200;
%if &nlinewt.>0 %then %do;
set paragraph(where=(f2 ne ""));
b=tranwrd(tranwrd(tranwrd(f2,"表14.","表 14."),"列表16.","列表 16."),"图14.","图 14.");
b=tranwrd(tranwrd(b,"表格14.","表格 14."),"图表14.","图表 14.");
a=cats(f1);
drop f1 f2;
%end;

%else %do;
set paragraph(where=(f1 ne ""));
b=tranwrd(tranwrd(tranwrd(f1,"表14.","表 14."),"列表16.","列表 16."),"图14.","图 14.");
b=tranwrd(tranwrd(b,"表格14.","表格 14."),"图表14.","图表 14.");
a="";

```

```

drop f1 ;
%end;

if prxmatch("/^(表[格]*|列表|图[表]*|Table|Listing|Figure)( )(1[46]\.)([d\.]+)([a-zA-Z\d\.]*) ( )/",b)
then do;
    titleid=cats(substr(compbl(b),1,find(compbl(b)," ",find(compbl(b)," ")+1)) );
end;
if missing(titleid) then titleid=b;

run;
%mend;
%parout

proc sql;
create table chek_content as
select a.titleid as content_titleid,b.titleid as paragraph_titleid
from content as a full join paragraph as b on a.titleid=b.titleid;
quit;

data chek_content;
set chek_content end=fn;
length flag $200;
retain lst;
if cmiss(content_titleid,paragraph_titleid)=0 then flag="No difference";
if cmiss(content_titleid,paragraph_titleid) ne 0 then do;
    if missing(paragraph_titleid) then flag="Paragraph not output";
    if missing(content_titleid) then flag="Content not output";
    lst+1;
end;

if fn and lst>. then do;
    put "WARNING: Paragraph and Content are not consistent, please see 'chk sheet' in
outtoc.xlsm";
end;

if fn then do;
drop lst;
call execute('proc export data=chek_content outfile="&_folder.\outtoc.xlsm" dbms=excel
replace;

sheet="chk";
run;');

end;
run;

```

```

*****output tracker and header*****;
data paragraph;
    set paragraph;
    length titlen $8;
    format _all_;
    informat _all_;
    titlen=cats(a);
    if prxmatch("/Table|表/",(kscan(titleid,1," "))) then _tlf="t";
    if prxmatch("/Listing|列表/",(kscan(titleid,1," "))) then _tlf="l";
    if prxmatch("/Figure|图/",(kscan(titleid,1," "))) then _tlf="f";
    drop a;
run;

proc contents data=paragraph noprint out=__prop;
run;
proc sort;
    by varnum;
    where name not in ("titleid","titlen","_tlf");
run;
proc sql noprint;
    select count(name) into :varn trimmed
    from __prop;
quit;

data _null_;
    length rename $1000 name1 $200;
    retain rename;
    set __prop end=lst;
    n=_n_;
    name1="col"||cats(n);
    rename=catx(" ",rename,cats(name,"=",name1));
    if lst then do;
        call symputx("rename",rename);
    end;
run;
%put _user_;
data _toc;
    set paragraph(drop=titleid);
    if missing(input(cats(titlen),??best.)) then titlen=symget(cats(_tlf,'titlen'));
    obs=_n_;
    rename &rename.;
run;

option mprint;

```

```

%macro multitle;
proc sql noprint;
    select distinct titlen into :titlens separated by " "
    from _toc;
quit;

%do ttn=1 %to %sysfunc(countw(&titlens));
%let titlen=%scan(&titlens.,&ttn.,%str( ));
data toc;
    length tt ff1-ff%sysevalf(&varn.-&titlen.) $1000;
    set _toc;
    where titlen="&titlen.";
    tt=catx("(*ESC*)n",of col1-col&titlen.);
    array ff ff1-ff%sysevalf(&varn.-&titlen.) ;
    array cls col%sysevalf(&titlen.+1)-col&varn.;
    do i=1 to dim(cls);
        ff[i]=cls[i];
        if "&repeat_txt." ne "" then do;
            if prxmatch("/&repeat_txt./i",upcase(ff[i])) and not prxmatch("/1\d\./",ff[i]) then repeat="Y";
        end;
        else repeat="";
    end;
run;

data toc2;
    set toc;
    array oraw[*] ff1-ff%sysevalf(&varn.-&titlen.);

    do _i=1 to dim(oraw);
        if "&dele_txt." ne "" then do;
            if prxmatch("/&dele_txt./i",strip(oraw[_i])) then oraw[_i]="";
        end;

        if "&repeat_txt." ne "" then do;
            if prxmatch("/&repeat_txt./i",strip(oraw[_i])) then oraw[_i]="";
        end;

        if oraw[_i] in (",") then oraw[_i]="";

        if "&prg_note." ne "" then do;
            if prxmatch("/&prg_note./i",oraw[_i]) then flag=_i;
            if _i>=flag. then oraw[_i]="";
        end;
    end;
end;

```

```
run;
```

```
data toc2_unicode;
```

```
length _toc $32767 unicode_from unicode_to $1000;
```

```
set toc2;
```

```
_toc=catx("||||",of ff1-ff%sysevalf(&varn.-&titlen.));
```

```
unicode_from="&unicode_from.";
```

```
unicode_to="&unicode_to.";
```

```
do _uni=1 to countw(unicode_from,"||||");
```

```
_toc=tranwrd(_toc,cats(kscan(unicode_from,_uni,"||||"),cats(kscan(unicode_to,_uni,"||||"))));
```

```
tt=tranwrd(tt,cats(kscan(unicode_from,_uni,"||||"),cats(kscan(unicode_to,_uni,"||||"))));
```

```
end;
```

```
drop _uni unicode_from unicode_to;
```

```
run;
```

```
data tocfinal_&titlen.;
```

```
retain __toc;
```

```
length tittle1-tittle7 _footnote1-_footnote9 __footnote1-__footnote9 FOOTN1-FOOTN9 $2000
```

```
__toc $32767 _tittle1 filename $2000 num1-num10 $200;
```

```
set toc2_unicode;
```

```
format _all_;
```

```
informat _all_;
```

```
if not missing(_toc) then __toc="";
```

```
if repeat="Y" then __toc=_toc;
```

```
array _ft _footnote1-_footnote9;
```

```
array __ft __footnote1-__footnote9;
```

```
array ft FOOTN1-FOOTN9;
```

```
do _ii=1 to 9;
```

```
if "&repeat_txt." ne "" and _tif="t" then do; /*add other condition for repeat*/
```

```
/* for repeat footnote*/
```

```
_ft[_ii]=kscan(_toc,_ii,"||||");
```

```
__ft[_ii]=kscan(__toc,_ii,"||||");
```

```
if _ii=9 then do;
```

```
if kscan(_toc,10,"||||") ne "" then
```

```
_footnote9=tranwrd(substr(_toc,find(_toc,kscan(_toc,9,"||||")),length(_toc)),"||||",("(*ESC*)n");
```

```
if kscan(__toc,10,"||||") ne "" then
```

```
__footnote9=tranwrd(substr(__toc,find(__toc,kscan(__toc,9,"||||")),length(__toc)),"||||",("(*ESC*)n");
```

```

end;

ft[_ii]=ifc(cmiss(of ff1-ff%sysevalf(&varn.-&titten.))
ne %sysevalf(&varn.-&titten.),_ft[_ii],__ft[_ii]);    /*repeat: equal to original value if have any footer,
otherwise repeat previous footer*/
end;

else do;
ft[_ii]=kscan(_toc,_ii,"||||");
if _ii=9 and kscan(_toc,10,"||||") ne "" then
FOOTN9=tranwrd(substr(_toc,find(_toc,kscan(_toc,9,"||||")),length(_toc)),"||||",("ESC*n"));
end;
end;
tittle1=tt;
call missing(of tittle2-tittle7);

*tittle: Table 14.xx xxxx*;
if prxmatch("/^(表[格]*列表|图[表]*|Table|Listing|Figure)( )(1[46]\.)([d\.])(.*)/" ,tittle1) then do;
_tittle1=kscan(compbl(cats(tranwrd(tittle1,("ESC*n", " "))),2, " "));
end;

array num[*] $200 num1-num10;
do _iii=1 to countw(_tittle1, ".");
num[_iii]=put(input(kscan(_tittle1,_iii, "."), ??best.),z2.);
if num[_iii] in(" .", "") and not missing(kscan(_tittle1,_iii, ".")) then
num[_iii]=kscan(_tittle1,_iii, ".");
end;

filename=catx("-",_tlf,num1,num2,num3,num4,num5,num6,num7,num8,num9,num10);

drop ff: _footnote1-_footnote9 __;;
run;
%end;
%mend;
%multitle;

data tocfinal;
set tocfinal_;;
if not prxmatch("/^(表[格]*列表|图
[表]*|Table|Listing|Figure)( )(1[46]\.)([d\.])([a-zA-Z\d\.]*)/" ,tranwrd(tittle1,("ESC*n", " "))
then put "WARNING: Title should be Table 14.xxx. aaaaa, please check paragraph sheet, row
= " obs;

```

```

run;

proc sort data=tocfinal out=_title_nodup dupout=_title_dup nodupkey;
    by tt;
run;

data _null_;
    set _title_dup end=end;
    if _n_>=1 then put "WARNING: Title duplicate, title = " tt;
run;

/*output tracker*/

data tracker;
    retain a tlf type ur _title1 rftt1 pgmname rtf filename;
    length a tlf type _tif $200 rftt1 filename _rtfname $2000 _title1 $2000
            num1-num10 $200;
    set tocfinal( keep=_title1 _title1 obs);
    if _title1 ne "";

    if prxmatch("/表|table/",lowercase(kscan(_title1,1," "))) then _tif="t";
    if prxmatch("/列表|listing/",lowercase(kscan(_title1,1," "))) then _tif="l";
    if prxmatch("/图|figure/",lowercase(kscan(_title1,1," "))) then _tif="f";

    array num[*] $200 num1-num10;
    do _iii=1 to countw(_title1,".");
        num[_iii]=put(input(kscan(_title1,_iii,"."),??best.),z2.);
        if num[_iii] in(" ","") and not missing(kscan(_title1,_iii,".")) then num[_iii]=kscan(_title1,_iii,".");
    end;

    filename=catx("-",_tif,num1,num2,num3,num4,num5,num6,num7,num8,num9,num10);

    _rtfname=compbl(tranwrd(_title1,"(*ESC*)n"," "));

    rftt1=strip(substr(_rtfname,find(_rtfname," ",find(_rtfname,"")+1)));
    a="Dry Run";

    tlf type=cats(put(_tif,$typee.),"s");
    call missing(ur,pgmname,rtf);
    keep a filename tlf type _title1 rftt1 obs ur pgmname rtf ;
    label a="Deliverable" tlf type="Type" ur="U/R" _title1="Output ID" rftt1="Output Title"
    rtf="Output (*-0TXT, *-0LST, *-0RTF) Datasets" pgmname="Program Name" filename="rtfname suffix";
run;

```

```

proc sort data=tracker ;by obs;run;

/*output Header footnote*/
data header;
    retain TYPE PGMNAME PGMID DATASET FILENAME TITTLE1-TITTLE7 FOOTN1-FOOTN9;
    length type pgmname dataset _tlf $200;
    set tocfinal;

    type=put(_tlf,$typee.);
    call missing(pgmname, pgmid, dataset,_prefix);
    pgmname="";
    filename="";
    keep type pgmname pgmid dataset filename TITTLE1-TITTLE7 FOOTN1-FOOTN9 obs;
run;
proc sort data=header;by obs;run;

/*output xlsx*/
proc export data=tracker(drop=obs) outfile="&_folder.\outtoc.xlsx" dbms=excel replace label;
    sheet="tracker";
run;

proc export data=header(drop=obs) outfile="&_folder.\outtoc.xlsx" dbms=excel replace;
    sheet="header";
run;

```

APPENDIX C – SIMPLE EXCEL VBA SCRIPT TO UPDATE RTFNAME AND PROGNAME IN TRACKER

```

Sub sas2header_tracker()
    Dim ottsheet As Object, folderpath As String

    folderpath = ActiveWorkbook.Path

    '-----
    '----- for tracker-----
    '-----

    Set ottsheet = Sheets("tracker")
    ottsheet.Select

    Dim lastRow As Long
    lastRow = ottsheet.Cells(ottsheet.Rows.Count, "D").End(-4162).row

    ottsheet.Range("J1") = "progname"
    ottsheet.Range("K1") = "rtfname"

```

'Modify columns style

ottsheet.Range("J:J,K:K").Interior.Color = vbYellow

ottsheet.Range("E1,G1,H1").EntireColumn.ColumnWidth = 24.33

' Clear previous content

ottsheet.Range("F2:F" & lastRow).ClearContents

ottsheet.Range("G2:G" & lastRow).ClearContents

' Create formula

```
ottsheet.Range(ottsheet.Cells(2, 6), otsheet.Cells(lastRow, 6)).Formula =  
"=CONCATENATE(LOWER(LEFT(B2,1)), "-" & J2, ".sas")"  
ottsheet.Range(ottsheet.Cells(2, 7), otsheet.Cells(lastRow, 7)).Formula = "=CONCATENATE(H2,  
"-" & K2)"
```

' Add Filter

ottsheet.Range("A1").EntireRow.AutoFilter

MsgBox "Complete"

End Sub

Sub Update_header_tracker()

Dim otsheet, otsheet As Object, folderpath As String

folderpath = ActiveWorkbook.Path

```
'-----  
'-----'Coy tracker to header-----  
'-----
```

Set otsheet = Sheets("tracker")

Set otsheet = Sheets("header")

Dim lastRow As Long

lastRow = otsheet.Cells(otsheet.Rows.Count, "D").End(-4162).row

ottsheet.Range("AA2:AA" & (lastRow)).Value = otsheet.Range("F2:F" & (lastRow)).Value
ottsheet.Range("AB2:AB" & (lastRow)).Value = otsheet.Range("G2:G" & (lastRow)).Value

ottsheet.Range("F2:F" & (lastRow)).Value = otsheet.Range("AA2:AA" & (lastRow)).Value
ottsheet.Range("G2:G" & (lastRow)).Value = otsheet.Range("AB2:AB" & (lastRow)).Value

otsheet.Range("B2:B" & (lastRow)).Value = otsheet.Range("F2:F" & (lastRow)).Value
otsheet.Range("E2:E" & (lastRow)).Value = otsheet.Range("G2:G" & (lastRow)).Value

'Unhide columns and delete extra columns

ottsheet.Range("H:AB").EntireColumn.Delete

```
MsgBox "Complete"  
End Sub
```