PharmaSUG China 2019 - Paper AD-029

Title and footnote automation, VBA helps you out

Shuai Wu, PRA Health Sciences;
Sean Yan, PRA Health Sciences

ABSTRACT

During our daily work of creating tables, listings for clinical trial data, we insert titles and footnotes of the output using the Proc Report of SAS™. The titles and footnotes are copied then pasted from Statistical Analysis Plan (SAP) shells and this is quite time-consuming with some potential manual mistakes. (And the process could be painful when the SAP gets updated rapidly.) There're several papers discussed about how to extract the title and footnote from the SAP. However, this paper will give an alternate option and introduce what the Visual Basic for Applications (VBA) in Microsoft Word is and how to extract titles and footnotes from a word document utilizing VBA and SAS™. This will make the whole titles/footnotes process from shells to TFLs dynamic and automated and provide the study with higher quality and efficiency.

INTRODUCTION

Generally speaking, most interns will receive a task which is to copy titles/footnotes of every output in the shell into a excel file and complaint will be raised up. The process of extracting titles/footnotes is absolutely a time-consuming "copy-paste" work. So this paper will try to use the power of VBA in Word to extract titles/footnotes from shell into excel file and then read this excel file into SAS™ as a dataset. Finally, titles/footnotes variables in this dataset will be invoked when generating corresponding output. Meanwhile during the process of extraction, the sub/super scripts and special character will be transferred into ODS equivalent values , too.

SHELL ASSUMPTIONS AND INTRODUCTION TO VBA IN WORD

In different pharmaceutical or CRO companies, shell will be in different styles. So It's necessary to set some assumptions so that we can use the power of programming to extract the titles and footnotes.

GENERAL ASSUMPTIONS

Different from previous paper has discussed, we will have less assumptions like below:

1. The type of shell must be word file and can be edited by VBA.
2. Each new output is in a new page and titles/footnotes are not in the footer section and header section.
3. For repeat outputs, they should be presented like the first output and can't be simply left out.
4. Each output can have no more than 10 titles and footnotes just like figure below.

Table 1.1
Disposition of Subjects Screened
Analysis Set: Enrolled Set

Region	Site/ (Number of Sites)	Principal Investigator/ (Number of Principal Investigators)	First Subject In (YYYY-MM-DD)	Last Subject Out (YYYY-MM-DD)	Number of Subjects Screened	SS	FAS
Overall	(XX)	(XX)	YYYY-MM-DD	YYYY-MM-DD	XXX	XX	XX
Region	XX	Dr Invest1	YYYY-MM-DD	YYYY-MM-DD	XXX	XX	XX
Region	XX	Dr Invest2	YYYY-MM-DD	YYYY-MM-DD	XXX	XX	XX
Region	(XX)	Dr Invest3	YYYY-MM-DD	YYYY-MM-DD	XXX	XX	XX
Region	(XX)	Dr Invest4	YYYY-MM-DD	YYYY-MM-DD	XXX	XX	XX

FAS=Full Analysis Set, SS=Safety Set.
Note: First Subject In=Date of first Screening visit. Last Subject Out=Date of last visit (in clinic) for last subject
Note: (XX) denotes a count rather than a specific site or investigator identification number.

Reference: Listings 1.2, 1.5
Program: PROGRAMNAME.SAS, YYYY-MM-DD at HH:MM Page x of y

Figure 1. shell template

Regarding to the superscripts, subscripts and special character in the titles/footnotes, we also take them into consideration to make them can be displayed in the RTF output.

VISUAL BASIC FOR APPLICATIONS (VBA) IN WORD

Many people are very familiar to the VBA in Excel and can use it masterly but they may not know VBA in Word very clearly. But actually, VBA in Word is very similar to the VBA in Excel because VBA is an application which can be used in many applications of Microsoft Corporation. We also can invoke Word application in VBA in Excel.

1. First, we need to invoke the VBA editor by “Alt” + “F11”.
2. Then, we ought to click “Tools” button and click “Preference” button.
3. Last, we need to tick “Microsoft Office 16.0 Object Library” like figure below.

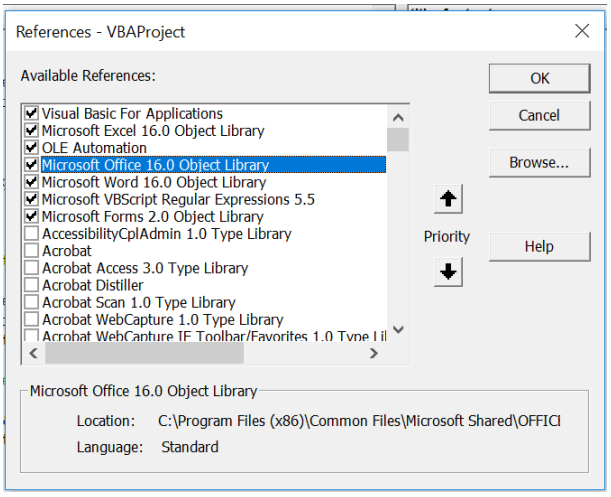


Figure 2. VBA references library

In the VBA of Word, the basic algorithm of VBA in Word is similar to it in Excel. But, every component of Word file can be an object and can be programmed and customized like paragraphs object and sentences object. Meanwhile, a page can be separated as Header section, Body section and Footer section. As previous paper discussed, they succeed to extract the titles/footnotes located in the Header section and Footer section. But in this paper, we will discuss how to extract titles/footnotes in the Body section.

EXTRACTION PROCESS WITH VBA

CREATE SETUP SHEET IN EXCEL

Before we start to extract the titles/footnotes, we need to do some preparations.

- Firstly, we need to create a new excel file and name first sheet as “setup page”, and we need user to add the path of shell meanwhile add the start page number of first output in shell so that we can make a loop from this page. The “Setup Page” is like figure below.

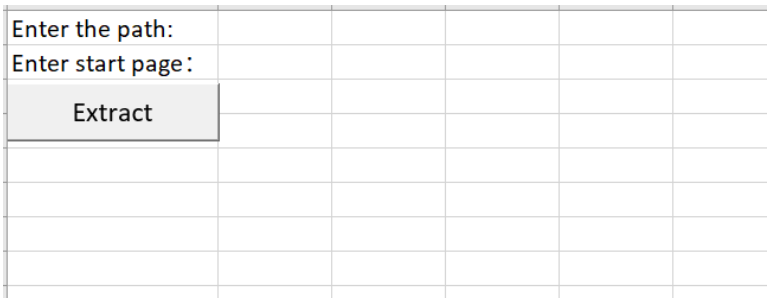


Figure 3. "Setup Page" sheet of excel file

- We need to create a new sheet named "title_footnote" using VBA code below to let the titles/footnotes in. Besides, fill the first row as "Title 1" to "Title 10" and "Footnote 1" to "Footnote 10" automatically.

```
Sub TITLE_FOOTNOTES()
    Dim XDOC As Word.Document
    Dim STARTPAGE As Word.Range, ENDPAGE As Word.Range
    Dim WDAPP As Word.Application
    Dim STRINPUT As String
    Dim XCEL As Workbook
    Dim WS As Worksheet
    Dim TOCNUM As Integer, PAGENUM As Integer, STARTNUM As Integer,
    ENDNUM As Integer
    Dim FULLNAME As String

    'Activate the word applcation
    Set WDAPP = CreateObject("Word.Application")
    FULLNAME = XCEL.Sheets("Setup_Page").Range("B1").Value
    Set XDOC = WDAPP.Documents.Open(Fullname)
    XDOC.Activate
    WDAPP.Visible = False

    'Activate the excel application and assign the titles/footnotes
    sheets
    Set XCEL = ActiveWorkbook
    XCEL.Sheets.Add After:=Sheets("setup page")
    ActiveSheet.Name = "title footnote"
    Set WS = XCEL.Worksheets("title footnote")

    'Add the title for this sheets
    For y = 1 To 10
    WS.Cells(1, y) = "title " & CStr(y)
    WS.Cells(1, y + 10) = "footnote " & CStr(y)
    Next
```

EXTRACT THE TITLES AND FOOTNOTES IN A LOOP

We will then do a loop from start page to the end page of output to extract the titles/footnotes of each page.

From the assumption, we can know titles are from the first paragraph of this page so it will be easily extracted meanwhile we can find each title are followed by a line break which can be presented as Chr(12). So we can use this delimiter to split each title and put them into the excel sheet. Meanwhile we will redefine a function to find and replace the subscripts and superscripts.

```
Sub FINDREP_SCRIPT(RANGE As Word.range, SPSCPT As Boolean, SBSCPT As
Boolean, FTXT As String, RTXT As String)
```

```
        RANGE.Find.ClearFormatting
        With RANGE.Find.Font
        .Strikethrough = False
        .DoubleStrikeThrough = False
        .Hidden = False
        .Smallcaps = False
        .Allcaps = False
        .Superscript = SPSCPT
        .Subscript = SBSCPT
        End With

        RANGE.Find.Replacement.ClearFormatting
        With RANGE.Find
        .Text = FTXT
        .Replacement.Text = RTXT
        .Forward = True
        .Wrap = wdFindContinue
        .Format = True
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
        End With
        RANGE.Find.Execute Replace:=wdReplaceAll
    End Sub
```

This new Sub procedure can be used to find and replace the subscripts and superscripts. For first variable “RANGE” it is where we want to search the superscripts and subscripts. The second and third variable “SPSCPT”, “SBSCPT” are Boolean variables can be defined to find/replace the subscripts or superscripts. For example, setting SPSCPT as True and SBSCPT as False means we want to find/replace the superscripts.

After the superscripts/subscripts is located, we need to define what needs to be replaced. For example, in this paper, if we find a superscript in title is “th”, we want to replace it as “~~{super th}” then we can use replace function to replace “~~” as “^” so that we can present the superscript/subscript correct in the rtf output. Code below is following part of sub routine TITLE_FOOTNOTE.

```
        'For loop to extract the title and footnotes
        STARTPAGE = XCEL.Sheets("Setup Page").Range("B2").Value
        ENDPAGE = XCEL.Sheets("Setup Page").Range("B2").Value

        For i = STARTNUM To ENDNUM
        Set STARTPAGE = XDOC.GoTo(What:=wdGoToPage, Which:=wdGoToAbsolute,
        Count:=i)

       'Replace the superscripts
        Call FINDREP_SCRIPT(STARTPAGE, True, False, "", "~~(super ^&)")
       'Replace the subscripts
        Call FINDREP_SCRIPT(STARTPAGE, False, True, "", "~~(sub ^&)")

        STARTPAGE.SetRange Start:=STARTPAGE.Paragraphs(1).Range.Start,
        End:=STARTPAGE.Paragraphs(1).Range.End
        STRINPUT = STARTPAGE.Text

       'Chr(12) means page break, Chr(11) means line break
        STRINPUT = Replace(Replace(STARTPAGE.Text, Chr(12), " "), "~~", "^")
       'Separate the title into title1 to title10
```

```
Dim TITLEARR() As String
TITLEARR = Split(STRINPUT, Chr(11))

For x = 0 To UBound(TITLEARR)
WS.Cells(i - 8, x + 1).Value = TITLEARR(x)
Next
```

After extracting the titles, we need to extract the footnotes and the logic is also very simple.

- Firstly, we need to define a text which can be searched which it will be “Source: ” sometimes after titles and then we can expand the area before this text.
- Then, we can do a loop to extract each paragraph before this text since each paragraph is the footnote what we want.
- Finally, we can just close the shell without saving it.

```
    'Extract the footnote
    Set ENDPAGE = XDOC.GoTo(What:=wdGoToPage, Which:=wdGoToAbsolute,
    Count:=i)
    'Replace the superscripts
    Call FINDREP_SCRIPT(ENDPAGE, True, False, "", "~~(super ^&)")
    'Replace the subscripts
    Call FINDREP_SCRIPT(ENDPAGE, False, True, "", "~~(sub ^&)")

    FOOTNUM = 1

    'To find the a word so that we can expand the area for footnotes
    Do
    With ENDPAGE.Find
    'The word we are looking for then expand to the previous paragraphs
    .Txt = "Reference: "
    .Execute
    If .Found Then ENDPAGE.Expand unit:=wdParagraph
    Set ARANGE = ENDPAGE.Previous(unit:=wdParagraph, Count:=Footnum)
    FOOTNOTE = ARANGE.Text
    WS.Cells(i - 8, Footnum + 9).Value =
    Replace(Replace(FOOTNOTE, Chr(7), ""), "~~", "^")
    FOOTNUM = FOOTNUM + 1
    End If
    End With
    Loop Until FOOTNUM > 10
    Next

    'Close the word without saving to release the memory
    XDOC.Close SaveChanges:=False
    Set XDOC = Nothing
    Set WDAPP = Nothing

End Sub
```

POST PROCESSING WITH SAS™

After extracting the titles/footnotes into excel file, we can import this file into the SAS as SAS dataset. Meanwhile we can invoke a macro to call out the titles/footnotes directly when we generate the output. The code is as below.

First step is we import the excel file into SAS and use COMPRESS function to delete the some annoying characters from titles and footnotes.

```
*Import the excel file;
options validvarname=upcase;

proc import datafile="&rootpath\&filename.xlsm" replace dbms=excel
      out=titfoot;
    getnames=yes;
    sheet='title footnote';
quit;

*delete the special character in titles;
data titfoot1;
    length OUTPUT_NAME $200;
    set titfoot;

    *According to the ASCII coding system, byte(0)-byte(32) and
      byte(127) are non-printable characters;
    array TITLE {*} TITLE_1-TITLE_10;
    array FOOTER {*} FOOTNOTE_1-FOOTNOTE_10;
    array NONPRINT {*} TEMP1-TEMP34 (0:32 127);
    do i=1 to dim(TITLE);
        do j=1 to dim(NONPRINT);
         if TITLE(i)^='' then
         TITLE(i)=strip(tranwrd(TITLE(i),byte(NONPRINT(j)),''));
         if FOOTER(i)^='' then
         FOOTER(i)=strip(tranwrd(FOOTER(i),byte(NONPRINT(j)),''));
        end;
    end;

    OUTPUT_NAME=title_1;

    *drop the temporary variable;
    drop TEMP: i j;
run;
```

After that, what we need to do is to reverse the footnote since we extract them from the bottom to top, which means if an output has totally 5 footnote, we extract N0.5 footnote as No.1 footnote into the excel file. So we need to reverse them to be in correct position.

```
*To reverse the footnote;
data titfoot2;
    set titfoot1;
    *get the number of non-missing footnote;
    MISSNUM=cmiss(of FOOTNOTE_1-FOOTNOTE_10);
    TOTALNUM=10-MISSNUM;
    *to reverse the footnotes for each number of non-missing footnote;
    %macro RENAME_FOOTER(NUM=);
    %if &NUM=1 %then %do;
        if TOTALNUM=&NUM then do;
            FOOTER_1=FOOTNOTE_1;
        end;
    %end;
    %else %do;
        if TOTALNUM=&num then do;
            array _&NUM._FOOTNOTE {&NUM} FOOTNOTE_1-FOOTNOTE_&NUM.;
            array _&NUM._FOOTER {&NUM} FOOTER_&NUM-footer_1;
            do i=1 to dim(_&NUM._FOOTNOTE);
                _&NUM._FOOTER(i)=_&NUM._FOOTNOTE(i);
            end;
```

```
        end;
    %end;
    %mend;
    %RENAME_FOOTER(NUM=1);
    %RENAME_FOOTER(NUM=2);
    %RENAME_FOOTER(NUM=3);
    %RENAME_footer(NUM=4);
    %RENAME_FOOTER(NUM=5);
    %RENAME_FOOTER(NUM=6);
    %RENAME_FOOTER(NUM=7);
    %RENAME_FOOTER(NUM=8);
    %RENAME_FOOTER(NUM=9);
    %RENAME_FOOTER(NUM=10);

    drop FOOTNOTE: i;

run;
```

Finally, we can invoke macro defined below to generate the titles/footnotes according to which output we want to generate.

```
options mprint mlogic symbolgen noquotelenmax;
%macro INVOKE_TITF(OUTNAME=);
data output_titfoot;
    set titfoot2;
    if OUTPUT_NAME='&OUTNAME';
run;

*count the valid number of titles and footnotes;
proc sql noprint;
    select count(name) into :TIT_NUM from dictionary.columns
    where upcase(libname)="WORK" and upcase(memname)="OUTPUT_TITFOOT"
      and  index(name,'TITLE') and length>3
    group by name;

    select count(name) into :foot_num from dictionary.columns
    where upcase(libname)="WORK" and upcase(memname)="OUTPUT_TITFOOT"
      and index(name,'FOOTER') and length>3
    group by name;
quit;

*Using titles/footnotes function to call out the titles/footnotes ;
%if &TIT_NUM ne 0 %then %do;
    %do i=1 %to &TIT_NUM;
        data _null_;
            set output_titfoot;
            call symput('temp',TITLE_&i);
        run;
        title%eval(&i) j=c font="Arial" h=9 pt  "&temp";
    %end;
%end;

%if &FOOT_NUM ne 0 %then %do;
    %do i=1 %to &FOOT_NUM;
        data _null_;
            set output_titfoot;
                call symput('temp',FOOTER_&i);
        run;
```

```
            footnote%eval(&i) j=l font="Arial" h=9 pt "&temp";
        %end;
    %end;

    %mend;
```

CONCLUSION

Different from previous paper discussed, we extract the titles/footnotes from shell in a different style so that we implement different logic to extract the titles/footnotes including the way of invoking the titles/footnotes in the report process without shell actively open before extraction. This paper has presented the process and advantage of SAS-VBA handshake in extracting the titles/footnotes from the shell. The overall benefits are it's easy operation and convenient update which can eliminate the time of extraction and give people an alternative way to get rid of the clumsy work of "copy-paste". Besides, this process can also help people to make updates quickly when statistician make changes of titles/footnotes of shell so that the whole process of updating the TFLs will be of obvious efficiency.

REFERENCES

Tony Cardozo. 2017. "Automating Title and Footnote Extraction Using Visual Basic for Applications (VBA) and SAS™." Proceedings of the PharmaSUG 2017 Conference.

CONTACT INFORMATION

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

Name: Shuai Wu
Enterprise: PRA Health Sciences
Address: 51F Raffles City, No. 268 Xizang Middle Road, Huangpu District • Shanghai
E-mail: wushuai@prahs.com

Name: Sean Yan
Enterprise: PRA Health Sciences
Address: 51F Raffles City, No. 268 Xizang Middle Road, Huangpu District • Shanghai
E-mail: yansean@prahs.com

Any brand and product names are trademarks of their respective companies.