

## Optimizing TLF Generation: Titles and Footnotes

### Applying a New Idea to a Basic Approach

Jake Adler, PROMETRIKA, LLC;  
Assir Abushouk, PROMETRIKA, LLC;

#### ABSTRACT

When receiving feedback from a sponsor on tables/listings one of the most frequent updates is title/footnote changes. These updates can be very time consuming, so what if there was a way to take the updated shells and automate that to the programs?

In this paper Excel Visual Basics for Applications (VBA) will be used to take the contents of titles/footnotes from the shells of a Word document and import that into SAS® so titles and footnotes can be read in automatically. To validate this process when the shells are updated Apache Subversion (SVN) will be used to compare changes in old shells versus new shells.

#### INTRODUCTION

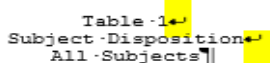
Title and footnote updates can take lots of time especially if there are many programs that need updating. Footnote updates can vary in their complexity; sometimes a simple change in capitalization is needed, while other times they need to be entirely rewritten. When writing Tables, Listings, and Figures (TLFs) we want them to be as robust as possible - if the data we read in changes, our programs will account for that. This is hard to do with titles and footnotes since these values are just text from a word document. Footnotes are hardcoded in the programs and we set them to a definite value so they don't change as data is updated like the rest of the program. Visual Basics for Applications is an event-driven programming language from Microsoft that is now primarily used with Microsoft office applications such as MS-Excel and MS-Word. This allows for macros to be created within Excel and in this case create a macro to read specific text in a word document and output into an excel sheet so we can read that into SAS®.

#### TABLE/LISTING/FIGURE SHELL FORMATTING

In order to read in the shells to excel a standard shell is needed. This includes appropriate paragraph markers as well as formatting. These paragraph markers can be visible by clicking the paragraph symbol (¶) in the home tab of Microsoft Word.

#### LINE BREAK

**Titles:** Every line in the title section must include a line break (SHIFT+ENTER) this allows the entire title to be one paragraph (Display 1).



```
Table 1  
Subject Disposition  
All Subjects
```

Display 1. Table title with the highlighted portions showing the line breaks

**Footnotes:** Within the body of the table in our shells, the last 2 rows must be footnote and PROGRAM:.... Those footer rows must also be included with the table contents not just by itself (Display 2). Line break is really useful for footnotes because the max number of footnotes that can be included in a proc report is 10 but with using line break there will only be one footnote.

```

Table 2*
Demographics and Baseline Characteristics
Safety Analysis Set

Age at Informed Consent (years)
  n
  Mean (SD)
  Median
  Min, Max
  < 55 years, n (%)
  < 65 years, n (%)
Sex, n (%)
  Male
Ethnicity, n (%)
  Hispanic or Latino
  Not Hispanic or Latino
Race, n (%)
  American Indian or Alaska Native
  Asian
  Black or African American
  Native Hawaiian or Other Pacific
  Islander
  White
  Other
  Not Reported
Country
  Canada
  United States
Footnote1
Footnote2
Footnote3
PROGRAM: filename.sas
Page Break
Section Break (Continuous)
DATE: DDMMYYYY.HH:MM

```

```

Table 2*
Demographics and Baseline Characteristics
Safety Analysis Set

Age at Informed Consent (years)
  n
  Mean (SD)
  Median
  Min, Max
  < 55 years, n (%)
  < 65 years, n (%)
Sex, n (%)
  Male
Ethnicity, n (%)
  Hispanic or Latino
  Not Hispanic or Latino
Race, n (%)
  American Indian or Alaska Native
  Asian
  Black or African American
  Native Hawaiian or Other Pacific
  Islander
  White
  Other
  Not Reported
Country
  Canada
  United States
Footnote1
Footnote2
Footnote3
PROGRAM: filename.sas
Page Break
Section Break (Continuous)
DATE: DDMMYYYY.HH:MM

```

Display 2. Image on the right is virtually the same as the left however the left image is the one needed as it includes footnote in same table as body text while right image separates it.

### SECTION BREAK

The shells must be broken up by a section page break instead of a regular page break. These section breaks will start right after the general programming notes and be included after each TLF in the shell (Display 3).

```

Table 2*
Demographics and Baseline Characteristics
Safety Analysis Set

Age at Informed Consent (years)
  n
  Mean (SD)
  Median
  Min, Max
  < 55 years, n (%)
  < 65 years, n (%)
Sex, n (%)
  Male
Ethnicity, n (%)
  Hispanic or Latino
  Not Hispanic or Latino
Race, n (%)
  American Indian or Alaska Native
  Asian
  Black or African American
  Native Hawaiian or Other Pacific
  Islander
  White
  Other
  Not Reported
Country
  Canada
  United States
Footnote1
Footnote2
Footnote3
PROGRAM: filename.sas
Page Break
Section Break (Continuous)
DATE: DDMMYYYY.HH:MM

```

Display 3. This shows a section break between table 2 and table 3.

## REPEATS

For repeat TLFs if the footnotes remain the same as previous TLF then a table with 2 rows under the new title will need to be added (Display 4).

```
Table 2.1
Demographics and Baseline Characteristics
Safety Analysis Set

Footnote1
Footnote2
Footnote3

PROGRAM: filename.sas DATE: DDMMYYYY HH:MM
```

**Display 4. Example of a repeat table with same footnotes. Note: table 2.1 is a repeat table of table 2 in Display 3 with a different title.**

## CREATING VBA CODE

In this section we will be describing the code that is used to read the TLF shell and output them into excel columns. Column 1 will display the title and column 2 will display the footnotes for each table in the shell. The Entire code is in Appendix 1.

```
Sub TLFnew()

Dim wd As New Word.Application
Dim doc As Word.Document
Dim Test As Word.Range
Dim sh As Worksheet
Dim r As Integer
Dim wordTbl As Table

wd.Visible = True
```

In the step below we will be entering the location of the shell along with the shell file name

```
shellloc = InputBox("Please enter the shell location:")
Set doc = wd.Documents.Open(shellloc)
```

Once the shells are open we can remove the first section since that will be the table of contents along with the programming notes which isn't necessary to read in. Following that we can count the total number of sections which will be the total count of all TLFs.

```
doc.Sections(1).Range.Delete
nsect = doc.Sections.Count
```

Now we can create a loop to go through each section and extract the contents needed.

```
For m = 1 To nsect
Set Test = doc.Sections(m).Range
```

When looking at a section we need to count the number tables within it. If there is any tables less than 2 rows that table will be deleted because it is not necessary. Every row from the first table will then be removed besides the last 2 rows. The first row of the last 2 rows that is kept will be assigned to column 2 which is the footnotes column.

```
tabcount0 = Test.Tables.Count

If tabcount0 > 0 Then
For w = tabcount0 To 1 Step -1
RowCount = Test.Tables(w).Rows.Count
If RowCount < 2 Then Test.Tables(w).Delete
Next w
End If

tabcount = Test.Tables.Count
If tabcount > 0 Then
For y = 1 To tabcount
colcount = Test.Tables(y).Rows.Count
For Z = colcount - 2 To 1 Step -1
Test.Tables(y).Rows(Z).Delete
colcount = Test.Tables(y).Rows.Count
Next Z
Next y
Col2 = Test.Tables(1).Rows(1).Range.Text
End If

If tabcount = 0 Then
Col2 = ""
End If
```

Here we will be getting column 1 values so we will be looking to see if the key word Table, Figure, or Listing will come up first within a section.

```
For c = r To lastsection
Set coll = doc.Paragraphs(c).Range
positionofSubstring = InStr(coll.Text, "Table ")
positionofSubstring2 = InStr(coll.Text, "Figure ")
positionofSubstring3 = InStr(coll.Text, "Listing ")
If positionofSubstring <> 0 And positionofRepeat = 0 Then
```

```

Tabnum = coll.Text
Exit For
End If
If positionofSubstring2 <> 0 And positionofRepeat = 0 Then
Tabnum = coll.Text
Exit For
End If
If positionofSubstring3 <> 0 And positionofRepeat = 0 Then
Tabnum = coll.Text
Exit For
End If

Next c

```

Finally, we assign the column 1 and column 2 values to the excel sheet and repeat the process for the rest of the sections (Display 5).

```

sh.Cells(j, 1).Value = Tabnum
sh.Cells(j, 2).Value = Col2

End Sub

```

	A	B
1	Table 1 \line Subject Disposition \line All Subjects	[a] Footnote1
2	Table 2 \line Demographics and Baseline Characteristics \line Safety Analysis Set	Footnote1 \line footnote2 \line footnote3
3	Table 3 \line Prior Cancer History at Initial Diagnosis \line Safety Analysis Set	Footnote4. \line footnote5
4	Listing 2 \line Serious Adverse Events	Footnote1
5	Table 2.1 \line Demographics and Baseline Characteristics \line Safety Analysis Set	Footnote1 \line footnote2 \line footnote3
6	Figure 17	Footnote 1

**Display 5. Here are the extracted shells with titles in left column and footnotes in right column.**

## IMPORT TO SAS AND PROC REPORT

Now that we have an excel sheet with all titles and footnotes from the shell we can export this into SAS® and create macro variables (&TITLE1., &FOOTNOTE1.) for the title and footnote depending on which TLF we are working on. The Entire code is in Appendix 2.

```
%macro TitleFoot(tlfnum=);

%global title1 footnote1;

proc import datafile="U:\TLFmacro\svn_excel.csv"
    out=shells_all
    dbms=csv
    replace;

    getnames=no;
run;

data Final(where=(VAR3=&tlfnum.) keep=VAR:);
    length VAR1 VAR2 $2000;
    set shells_all;
    format VAR1 VAR2;
    informat VAR1 VAR2;

    VAR1=compress(VAR1,,'kw');
    VAR2=compress(VAR2,,'kw');

    if index(VAR1,' \line ') gt 0 then
        tnum0=find(VAR1,' \line ');
    if tnum0 ne . then
        VAR3=strip(substr(VAR1,1,tnum0));

run;

proc sql noprint;
    select VAR1 into :TITLE1 - from Final;
    select VAR2 into :FOOTNOTE1 - from Final;
quit;

%mend;

%TitleFoot(tlfnum="Table 2");
```

This method is much easier to input titles and footnotes in the SAS® program and allows for less human error. Both methods below produce the same solution

```

title1 j=1 "&company" j=r "Page !{pageof}";
title2 j=1 "&protocol" ;
title3 "&TITLE1.";
footnote1 j=1 "&linetop &FOOTNOTE1.";
footnote3 j=1 "Program: &srcfile..sas" j=r "Run date: &actdate &acttime";
Vs.

```

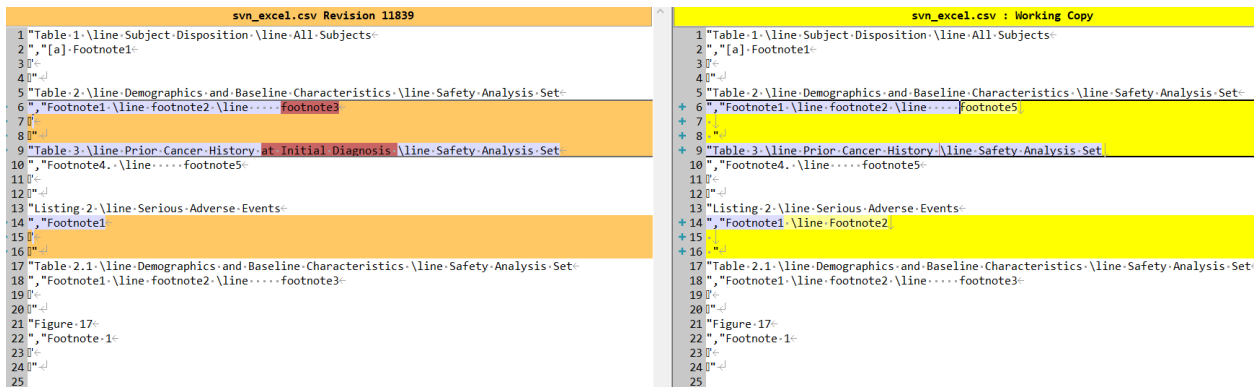
```

title1 j=1 "&company" j=r "Page !{pageof}";
title2 j=1 "&protocol" ;
title3 "Table 2";
title4 "Demographics and Baseline Characteristics";
title5 "Safety Analysis Set";
footnote1 j=1 "&linetop Footnote1";
footnote2 j=1 "Footnote2";
footnote3 j=1 " Footnote3";
footnote5 j=1 "Program: &srcfile..sas" j=r "Run date: &actdate &acttime";

```

## VALIDATION METHOD

Throughout the study footnotes will be updated and we need to rerun the VBA code to account for updated titles and footnotes. We want to make sure that what changed within the shell also changed in the spreadsheet that we read into SAS®. Our method to validate footnote updates was to use the compare feature in Apache Subversion (SVN). This feature produces a side-by-side comparison between the excel versions highlighting changes between the base and working copy. It also calculates for any additions or subtractions in TLF titles/footnotes and doesn't consider it a new TLF. SVN is not the only tool that can be used to validate footnote changes, other compare tools include but are not limited to, Beyond Compare, Kaleidoscope, ExamDiff, and even web based compare tools found by using a quick google search.



Display 6. Side by side display in SVN of old shell(left) versus new shell(right).

## CONCLUSION

Having the ability to programmatically get titles and footnotes will greatly decrease the time it takes to make TLF updates. It will also reduce in possible error of inputting manually. Through this paper we have realized how beneficial a tool like VBA can be and how it can be applied to many other tasks to help save time.

## REFERENCES

Thierry Jegoux, Maud Garnier. 2008. "Titles and footnotes management from the TLF shell to the final outputs" Proceedings of PhUSE 2008 Conference, Paper CS04.

Ajay Gupta. 2011. "Reading Title and Footnote from RTF Output into SAS™ utilizing Microsoft Excel" Proceedings of the PharmaSUG 2011 Conference, Paper CC11.

Dongsun Cao. 2012. "Automation of Comparing ODS RTF Outputs in Batch using VBA and SAS™." Proceedings of the PharmaSUG 2012 Conference, Paper CC08.

Tony Cardozo. 2017. "Automating Title and Footnote Extraction Using Visual Basic for Applications (VBA) and SAS™." Proceedings of the PharmaSUG 2017 Conference, Paper BB20.

Shuai Wu, Sean Yan. 2019. "Title and footnote automation, VBA helps you out." Proceedings of the PharmaSUG China 2019 Conference, Paper AD029.

Igor Goldfarb, Ella Zelichonok. 2021. "Macro to Compare Titles and Footnotes in Produced TLF and Corresponding Shells." Proceedings of the PharmaSUG 2021 Conference, Paper AD179.

Frolov, Alexander. "How to remove carriage returns (line breaks) from cells in Excel." Accessed March 14,2021. <https://www.ablebits.com/office-addins-blog/2013/12/03/remove-carriage-returns-excel/>

Gipson, Susan. "VBA in Excel: What is Visual Basic for Applications, How to Use." Accessed March 14,2021. <https://www.guru99.com/creating-your-first-visual-basic-for-applications-vba-in-excel.html>

## CONTACT INFORMATION

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

Name: Jake Adler  
E-mail: [jadler@prometrika.com](mailto:jadler@prometrika.com)

Name: Assir Abushouk  
E-mail: [aabushouk@prometrika.com](mailto:aabushouk@prometrika.com)

## APPENDICES

### Appendix 1: VBA Code for output Titles and Footnotes to Excel sheet

```
Sub TLFnew()  
  
Dim wd As New Word.Application  
Dim doc As Word.Document  
Dim Test As Word.Range  
Dim sh As Worksheet  
Dim r As Integer  
Dim wordTbl As Table
```



```

wd.Visible = True

shellloc = InputBox("Please enter the shell location:")
Set doc = wd.Documents.Open(shellloc)

doc.Sections(1).Range.Delete
nsect = doc.Sections.Count

r = 1
rr = 0
j = 1

For m = 1 To nsect
Set Test = doc.Sections(m).Range
tabcount0 = Test.Tables.Count

If tabcount0 > 0 Then
For w = tabcount0 To 1 Step -1
RowCount = Test.Tables(w).Rows.Count
If RowCount < 2 Then Test.Tables(w).Delete
Next w
End If
tabcount = Test.Tables.Count
If tabcount > 0 Then
For y = 1 To tabcount
colcount = Test.Tables(1).Rows.Count
For Z = colcount - 2 To 1 Step -1
Test.Tables(y).Rows(Z).Delete
colcount = Test.Tables(y).Rows.Count
Next Z
Next y
Col2 = Test.Tables(1).Rows(1).Range.Text
End If
If tabcount = 0 Then
Col2 = ""
End If

```

```

Set sh = ActiveSheet

nSections = Test.Paragraphs.Count
lastsection = nSections + rr

Tabnum = ""
repeatval = ""

For c = r To lastsection
Set coll = doc.Paragraphs(c).Range
positionofSubstring = InStr(coll.Text, "Table ")
positionofSubstring2 = InStr(coll.Text, "Figure ")
positionofSubstring3 = InStr(coll.Text, "Listing ")
If positionofSubstring <> 0 And positionofRepeat = 0 Then
Tabnum = coll.Text
Exit For
End If
If positionofSubstring2 <> 0 And positionofRepeat = 0 Then
Tabnum = coll.Text
Exit For
End If
If positionofSubstring3 <> 0 And positionofRepeat = 0 Then
Tabnum = coll.Text
Exit For
End If

Next c

sh.Cells(j, 1).Value = Tabnum
sh.Cells(j, 2).Value = Col2

j = j + 1

nnSections = lastsection
r = nnSections + 1
rr = nnSections

```

```
Next m
```

```
Dim MyRange As Range
```

```
Application.ScreenUpdating = False
```

```
Application.Calculation = xlCalculationManual
```

```
For Each MyRange In ActiveSheet.UsedRange
```

```
If 0 < InStr(MyRange, Chr(11)) Then
```

```
MyRange = Replace(MyRange, Chr(11), " \line ")
```

```
End If
```

```
Next
```

```
Application.ScreenUpdating = True
```

```
Application.Calculation = xlCalculationAutomatic
```

```
End Sub
```

## Appendix 2: SAS® Code to extract contents from excel sheet

```
%macro TitleFoot(tlfnm=);
```

```
%global title1 footnote1;
```

```
proc import datafile="U:\TLFmacro\svn_excel.csv"  
    out=shells_all  
    dbms=csv  
    replace;
```

```
    getnames=no;
```

```
run;
```

```
data Final(where=(VAR3=&tlfnm.) keep=VAR:);  
    length VAR1 VAR2 $2000;  
    set shells_all;  
    format VAR1 VAR2;  
    informat VAR1 VAR2;
```

```
    VAR1=compress(VAR1,,'kw');
```

```
    VAR2=compress(VAR2,,'kw');
```

```
    if index(VAR1,' \line ') gt 0 then
```

```
tnum0=find(VAR1,' \line ');
if tnum0 ne . then
VAR3=strip(substr(VAR1,1,tnum0));

run;

proc sql noprint;
    select VAR1 into :TITLE1 - from Final;
    select VAR2 into :FOOTNOTE1 - from Final;
quit;

%mend;

%TitleFoot(tlfnm="Table 2");
```