

## Using Visual Basic for Application to Produce Table of Contents from SAS Output List Files

Zemin Zeng, Forest Research Institute, Inc., Jersey City, NJ  
Mei Li, ImClone Systems, Branchburg, NJ  
Meng Pan, ImClone Systems, Branchburg, NJ

### ABSTRACT

The paper is a continuation and extension of our earlier work on applying Visual Basic for Application (VBA)® procedures in the Microsoft Word® to manipulate SAS® output Table/Listing (TL) files [Zeng & Li, SAS Global Forum 2010]. In this paper, we shall explore further to show how to assemble SAS list files into a single file by the order of their table/listing numbers, and produce Table of Contents (TOC) from SAS output Table/Listing (TL) files. All these manipulation tasks on SAS list files shall be accomplished by clicking buttons to execute VBA macros. The testing of the VBA macros was verified in a Windows Operating System PC with Microsoft Office 2003 or 2007.

### INTRODUCTION

In Pharmaceutical industry, it is common for SAS programmers to create customized Rich Text Format (RTF) or Word tables from the SAS output Table/List (TL) files. The SAS output TL files are space-delimited and not tab-delimited, necessary post-process of the output files is needed to create publication quality tables in RTF or Word format for clinical study reports. For the convenience of quality review and cross checking all the RTF or Word tables and lists, it is highly demanded to assemble the individual RTF or Word files into a single file by the order of table/listing numbers and add a Table of Contents (TOC) to the fully assembled file. In this paper, we shall show how we could accomplish these tasks by applying Visual Basic for Application (VBA) techniques.

Visual Basic for Application (VBA) is a very powerful object-oriented programming language that can be used to add to the functionality of Microsoft Office applications, including Microsoft Word. Microsoft Visual Basic allows people to define their own VBA projects to programmatically execute tasks in Microsoft Word, all manual operations in Microsoft Word essentially can be programmatically achieved via VBA. In the paper (Zeng & Li, SAS Global Forum, 2010), authors demonstrated dynamical interaction between VBA and SAS programming and developed a SAS macro to create VBA procedures for the below applications:

1. Convert SAS list files in a given folder to RTF/Word file;
2. Assemble all SAS list files (by file names) to a single Word file;
3. Extract the one pages of each SAS list file;
4. Assemble all first pages into a single Word file.

In this paper, we shall explore further to add three additional applications:

5. Assemble list files to a single file by table numbers;
6. Assemble all first pages to a single file by table numbers;
7. Add Table of Contents to a Word file.

Among the above three applications, adding Table of Contents (TOC) to a Word table/listing file is the most challenging one. The process of automating Microsoft Word from Visual Basic for Application for producing TOC based on SAS output TL files shall be introduced by the following main steps in the coming section:

1. Obtain all SAS output Table/Listing (TL) file names in a given folder and order them by TL numbers;
2. Insert all TLs by TL numbers into a single file and convert it to a Word file with appropriate page setup;
3. Set the TLs title style as Headings and call VBA TableofContents Object to add TOC to the Word file.

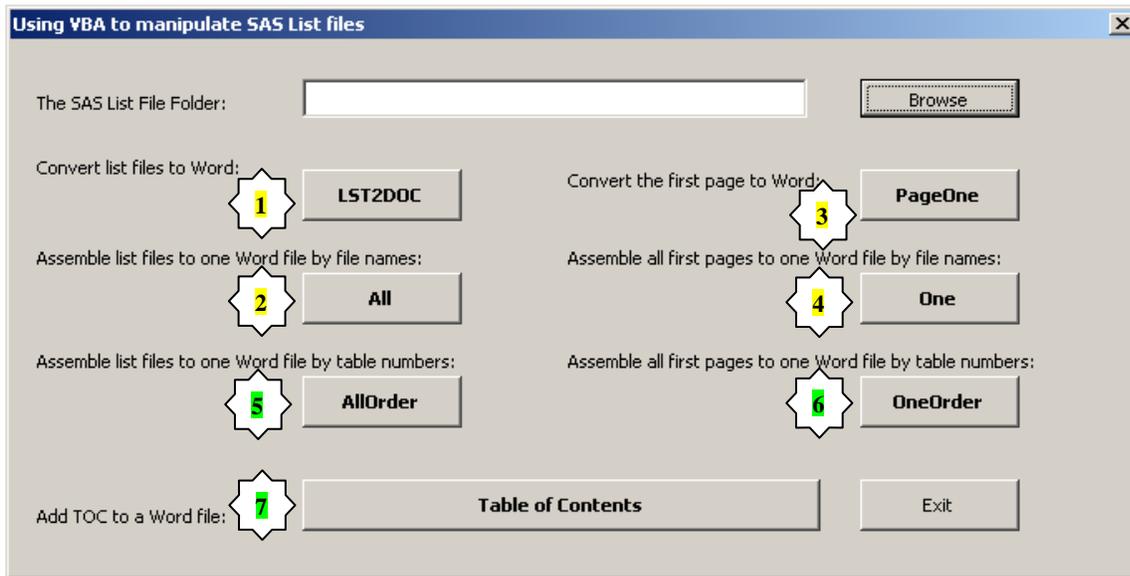
Also we use Visual Basic for Application (VBA) techniques to make VBA buttons and make these seven applications easy and convenient to use. Figure 1 is the VBA utility tool we developed based on our company-customized SAS output TL files at work, the users may simply click the buttons to execute VBA macros to manipulate SAS output Table/Listing (TL) files. We shall attach our VBA utility tool along with the paper for demonstration purpose, and hope it will inspire colleagues in Pharmaceutical industry to develop similar tools for their companies based on their company-customized SAS output TL files by following the ideas and detailed procedures described in the paper.

## USING VBA TO PRODUCE TOC BASED ON SAS OUTPUT TL FILES

A table of contents can be helpful, especially in a long Clinical Study Report. To get an idea about making TOC in Word, we may briefly recall what we usually manually do as below:

1. In a Word document, select the heading or sub heading of a section of the document;
2. In the Styles and Formatting style-selection box, click the down-arrow to open the styles menu and select a heading style, say Heading 1;
3. Repeat for each section heading;
4. If you have sub sections, repeat but select a higher level Heading 2 from the styles menu;
5. Click the document where you want to insert the table of contents;
6. On the Insert menu, point to Reference, and click Index and Tables then click the Table of Contents tab (set show levels to 2)

**Figure 1: The VBA tool to manipulate SAS list files**



The Goal of this section is to show our process of producing TOC based on SAS output Table/Listing files in a given folder by using VBA macros (See Figure 2 for a sample of our SAS output table and Figure 3 for a sample of the targeted TOC). In our setting, the section is the table/listing number of each SAS output TL and the subsection is the titles of the SAS output TLs. The TOC we would like to add at the beginning of an assembled Table/Listing Word file which consists of many tables and listings is as in Figure 3. For each table and listing we shall present the table/listing number, its titles and page number, and create a link in the TOC to its first page.

**Figure 2: Identify Table numbers and titles**

Forest Research Institute  
Protocol XXX-MD-XXX

XXXXXXXXXX  
Page 1 of 2

Table 14.2.1A  
Demographics and Baseline Characteristics  
Double-Blind Safety Population

Parameter (Unit)	Placebo (N=xxx)	XXXXXXXXXX (N=xxx)	Total (N=xxx)	P-value
Age (years)				
Mean	52.9	52.6	52.8	0.7827
SD	10.2	11.2	10.6	
Median	52.0	52.0	52.0	
Min, Max	23, 74	26, 77	23, 77	
n	xxx	xxx	xxx	

### Figure 3: Sample TOC

#### TABLE OF CONTENTS:

Table 14.1.1.....	5
Patient Disposition by Study Center	
Screened Population	
Table 14.1.3A.....	6
Premature Discontinuations	
Double-Blind Randomized Withdrawal Phase	
Randomized Population	
Table 14.1.3B.....	7
Premature Discontinuations	
Double-Blind Down-Titration Phase	
Randomized Population	
Table 14.1.4A.....	8
List of Patients Who Prematurely Discontinued	
Double-Blind Randomized Withdrawal Phase	
Randomized Population	
Table 14.1.4B.....	9
List of Patients Who Prematurely Discontinued	
Double-Blind Down-Titration Phase	
Randomized Population	

Since our starting point is from individual SAS output tables/listings, we need to first sort the table/listing numbers and then assemble them before we follow the above steps to add TOC to a Word document. Since SAS output TL files are space-delimited without any format applied, the most challenging parts to make TOC from SAS output TL files are to identify the table/listing numbers and titles of the SAS tables/listings and apply heading styles to them programmatically through VBA macros. Below are the steps of our process of creating TOC from SAS output TL files:

#### **STEP 1. OBTAIN ALL SAS OUTPUT TABLE/LISTING (TL) FILE NAMES IN A GIVEN FOLDER AND ORDER THEM BY TL NUMBERS**

Assuming all our SAS output TL files reside in a folder, we do the following to complete this step:

- Browse to the folder
- Get the full path of the folder
- Use the DIR function and wildcard \*.lst to get the file names of all SAS list files in the folder
- Open each SAS table/listing file in Word, based on our company-customized SAS table/listing output patterns. Search each SAS output table/listing for table numbers and table titles (See **Figure 2** for a sample of our SAS output table/listing)

Note: One may use the next line to 'Page 1 of ' to identify the table number line based on the company-customized output pattern. Then identify the next rows till the blank line as the table titles.

- Use arrays to store the file names and associated table/listing numbers
- Modify the Table/Listing numbers by pre-filling '0's if for the numbers between the periods are in between 0 and 9; for example, modify Table 14.5.4.2A to Table 14.05.04.02A. Then sort the table numbers by alpha-beta order. This modification ensures a right order of tables/listings (for example, the modification of Table 14.5.4.2 is Table 14.05.04.02 and is before the Table 14.05.04.10 in alpha-beta order)

Note: If one section has more than 99 tables/listings, one may need to pre-fill two zeros to a single digit numbers and one zero to a two digits numbers to ensure expected alpha-beta order.

- Use array to store the sorting order of the modified TL numbers, so now for each table/listing, we have arrays to store the information of file names, table/listing numbers and their sorting order.

#### **STEP 2. INSERT ALL TLS BY TL NUMBERS INTO A SINGLE FILE AND CONVERT IT TO A WORD FILE**

We use the VBA InsertFile to assemble all the Tls to a single file under loop through each SAS list file. If we use DIR to initiate the loop, then our assembled file is ordered by file names of the individual SAS list files. For inserting all Tls by TL numbers, we use sorting order stored in the array obtained in Step 1 to loop through all SAS list files.

To keep the same layout as the SAS list files, one needs to appropriately setup the pages in Word based on the company conventions on SAS output files, including page Orientation, top, left, right and bottom margins, page width and height, and font name and size. Then save it as a Word file and complete Step 2.

#### **STEP 3. SET THE Tls TITLE STYLE AS HEADING AND CALL VBA TABLEOFCONTENTS OBJECT TO ADD TOC TO THE WORD FILE**

In this step, we shall loop through all the lines of a Word file, add Heading styles to the table numbers and titles, then call VBA TableofContents Object to add the TOC to the Word file. This step is the most challenge part and we shall list all the sub-steps below and all the problems we faced, and show the ways we work around.

- Get the total number of lines of the assembled Word file and start loop by lines

Notes: For a Word file with huge number of lines, one need to declare the variable for the total number of lines of the Word file as Long or Double type in VBA. For Integer variables, they are stored as 16-bit (2-byte) numbers ranging in value from -32,768 to 32,767, which is only good for the maximum page number of 618 (=32767/53=total numbers of lines/page size). For Long (long integer) variables, they are stored as signed 32-bit (4-byte) numbers and will be good for a file with less than 40 million pages. If more than 40 million pages, we can refer to Double type which defines a much wider range of numbers. If the total number of the lines is out of the range of a particular variable type in the declaration in VBA, we shall get a *Run-time error '6': Overflow* message and VBA program stops running.

- For each line, check if it contains the table number or table titles. If it contains the table number and the table number appears the first time, define the style of the line as Heading 1; If it contains the table titles and appears the first time, define the style of the lines as Heading 2

Notes: Once the table number/title lines are selected, we may define the Heading style (say Heading 1 style) as below:

```
Selection.Style = ActiveDocument.Styles("Heading 1")
  With ActiveDocument.Styles("Heading 1").Font
    .Name = "Courier New"
    .Size = 7
    .Bold = False
  End With
  With ActiveDocument.Styles("Heading 1").ParagraphFormat
    .SpaceBefore = 0
    .SpaceAfter = 0
  End With
  With Selection.Font
    .Bold = False
  End With
ActiveDocument.Styles("Heading 1").NoSpaceBetweenParagraphsOfSameStyle =False
```

The font and its size is based on the company-customized SAS output setting, users may need to change accordingly per their company convention on SAS output tables/listings. To avoid blank lines before or after the selected line, we specify SpaceBefore=0 and SpaceAfter=0 in the Heading style definition. Users may define the Bold font in the TOC if they specify Bold=true in the Heading style definition.

- Move the cursor to the beginning of the Word file, call VBA TableofContents to add TOC to the Word file  
Note: Alternatively, one may directly add the TOC field code "TOC lo 1-2 ln 2-2 lf 1h lz" to the beginning of the Word file. Such TOC field code could be found in the Table of Content part in a Word file with TOC added by pressing ALT + F9.

- Insert Word page break between the TOC and the first table/listing of the file
- Move the cursor to the beginning of the Word file and add the paragraph Table of Contents
- Call VBA TableofContents to update the page numbers

Note: Due the addition of Table of Contents paragraph to the Word file, the actual page numbers in the TOC might be not correctly presented, we need to call VBA TableofContents object to update the page numbers.

- Add a right-aligned page number to the primary footer in the Word file

Note: The code below does the job; one may change the alignment of the page numbers in the footer by specify wdAlignPageNumberLeft for left and wdAlignPageNumberCenter for center.

```
With ActiveDocument.Sections(1).Footers(wdHeaderFooterPrimary)
  .PageNumbers.Add PageNumberAlignment:=wdAlignPageNumberRight
End With
```

By following the above detailed steps and sub steps, we complete our process of producing TOC based on SAS output Table/Listing files in a given folder by using VBA macros. For the interested readers, we are welcome to request more information on our VBA utility tool we developed based on our company-customized SAS output tables/listings along with this paper. The VBA utility tool could be easily adapted to fit the readers need by modifying the font and its font size, page margins and orientation based on their company SAS output settings. The assembled Word file with TOC added is really helpful for reviewing the SAS output tables/listings in clinic study reports to Clinic, Biostatistics and Statistical Programming groups.

## **CONCLUSION**

The process of automating Microsoft Word from Visual Basic for Application for producing TOC based on SAS output TL files could be easily adapted. The VBA utility tool is very convenient to use and the combined Word file of SAS output tables and listings with TOC added is very helpful for reviewing the clinical study results.

## **REFERENCES**

Zeng, Zemin and Li, Mei (2010). *Using SAS® Software and Visual Basic for Application to Dynamically Manipulate SAS List Files*, Coders' Corner Section, Proceedings of the SAS Global Forum 2010.

## **ACKNOWLEDGMENTS**

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.

## **CONTACT INFORMATION**

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

Zemin Zeng, Forest Research Institute, Inc., Jersey City, NJ  
Email: [zengzemin@gmail.com](mailto:zengzemin@gmail.com) / [zemin.zeng@frx.com](mailto:zemin.zeng@frx.com)

Mei Li, ImClone Systems, 33 ImClone Drive, Branchburg, New Jersey  
Email: [Jennifer.li@imclone.com](mailto:Jennifer.li@imclone.com)

Meng Pan, ImClone Systems, 33 ImClone Drive, Branchburg, New Jersey  
Email: [meng.pan@imclone.com](mailto:meng.pan@imclone.com)