

My Word, it's a table of contents

John Woods, ICON Clinical Research, Dublin, Ireland
Esther Bayon, ICON Clinical Research, Dublin, Ireland

ABSTRACT

At the end of the day, the reason that statistical programmers program, is so that the output can be reviewed and analysed by others. As typical clinical trial projects can have large amounts of tables, figures and listings output, it is important that these are well presented and straightforward to navigate so that they can be accessed and cross examined easily. This paper describes an efficient way to present a project's statistical output as a single Word document by automatically appending all outputs into the document and creating a table of contents with bookmarked links to each individual output in the file.

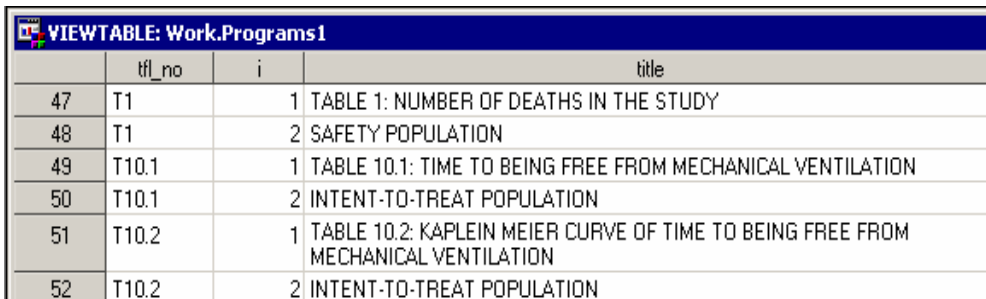
INTRODUCTION

The Word macro is the final part of the overall output processing of project Tables, Figures and Listings; and its main aim is to help automate a manual process by generating VBScript code based on the input parameters to it. For this paper, it is assumed that an individual file is created for each output. When all project program outputs have been created, there are two extra steps to complete in order to create this final document. The first step is to call the macro, developed using SAS®. The macro can be called from any SAS program (typically a batch program) after all outputs in the project have been created. The macro creates a .BAS (VBScript Source Code) file which is specific to the project, depending on input parameters passed to it. BAS programs are compatible with Microsoft's Visual Basic Scripting software, so the second step is to import this BAS file into the Visual Basic Editor in Word's macro facility, and here, it will be run to create the final document.

This paper will describe in detail the steps outlined above, that is, the macro definition and functionality, and using the BAS file to create the final document.

THE SAS MACRO OVERVIEW

This macro was developed in conjunction with the Titles and Footnotes macro, which was presented as a poster at the 2006 PhUSE conference. The basic idea of this macro is that all titles and footnotes are kept in Excel files, separate to the SAS TFL programs, with a spreadsheet listing each of the project outputs with a reference to all associated titles and footnotes. This makes it easy to maintain when many outputs use the same titles or footnotes in a study. This macro creates datasets in the working directory for reference during program execution. The Word macro calls the %setup() function from the titles and footnotes macro to create the titles dataset in the working directory. This displays the program name (tfl_no) and associated titles with sequence numbers (i) in order of appearance in the output, as in figure 1.



The screenshot shows a SAS dataset viewer window titled 'VIEWTABLE: Work.Programs1'. It displays a table with four columns: a row number, 'tfl_no', 'i', and 'title'. The data is as follows:

	tfl_no	i	title
47	T1	1	TABLE 1: NUMBER OF DEATHS IN THE STUDY
48	T1	2	SAFETY POPULATION
49	T10.1	1	TABLE 10.1: TIME TO BEING FREE FROM MECHANICAL VENTILATION
50	T10.1	2	INTENT-TO-TREAT POPULATION
51	T10.2	1	TABLE 10.2: KAPLEIN MEIER CURVE OF TIME TO BEING FREE FROM MECHANICAL VENTILATION
52	T10.2	2	INTENT-TO-TREAT POPULATION

Figure (1)

PhUSE 2007

The Word macro will use this dataset to read in the titles to be displayed in the final Word document's table of contents. It will also use these titles to match with the titles of the TFL output file to create the bookmarked link in the table of contents to the output's location in the Word document.

The input parameters of the macro describe where the project TFL outputs are stored (for automatic insertion), whether a table of contents is required in the document and typical Word document settings, such as page orientation, margins and font sizes. If these settings parameters are not given any values, then default values are generated by the program for the Word document page setup.

The macro generates the output as simply writing text strings (in the form of VBScript) to a dataset, and then writing this dataset to an output file with an extension of .BAS.

THE MACRO DEFINITION

This macro was developed with the following folder structure in place for all projects:



Figure (2)

The user must specify which output types (Efficacy / Listings / Safety) to include in the final output.

The definition of the macro is:

```
%macro process_output(outname=, type1=, prefix1=, suffix1=, toc1=, type2=, prefix2=,
suffix2=, toc2=, type3=, prefix3=, suffix3=, toc3=, type4=, prefix4=, suffix4=,
toc4=, suff=, ordtfl=, paper=, orient=, unit=, margint=, marginb=, marginl=,
marginr=, fontn=, fonts=);
```

From the definition, the only compulsory parameters are the *OUTNAME*, *TYPE1* and *SUFFIX1*.

The BAS file to be used in Word's macro functionality will be called 'Insert *OUTNAME*.BAS'.

TYPE1 is the name of the folder containing the files to be inserted into the Word document, and *SUFFIX1* is the extension of the files in this folder. There are four groups of variables; Type, Prefix, Suffix, TOC which allow up to four different locations of individual output files to be inserted into the final document. If the Type variable is not NULL then the Suffix variable must also be not NULL. It is assumed that all output files selected in each folder have the same prefix and extension.

SUFF is for the file extension of figures in any of the output files. If it is populated, it is compared to all output types and used as a flag to generate code for inserting images instead of text tables for all relevant outputs.

ORDTFL allows the user to order the outputs either by their type (1-4) or prefix (e.g. TFL).

The other parameters are associated with Page Setup in Word. Usually, these will be standard across a project, and so default values for each can be pre-defined in the macro. Typical values for these could be:

```
paper=A4, orient=L, unit=IN, margint=1.25, marginb=1, marginl=1.25, marginr=1.25,
fontn=Courier New, fonts=8
```

An example of a call to the macro:

```
%macro process_output(outname=Project1, type1=Efficacy, prefix1=F, suffix1=gif,
toc1=, type2= Safety, prefix2=T, suffix2=lst, toc2=(1 2), type3= Listings,
prefix3=L, suffix3=lst, toc3=(1 2), type4=, prefix4=, suffix4=, toc4=, suff=gif,
ordtfl=, paper=, orient=, unit=, margint=, marginb=, marginl=, marginr=, fontn=,
fonts=);
```

The result of this call to the macro will be an output file called 'Insert_Project1.BAS' which contains VBScript code to include all figures from the Efficacy folder (beginning with F and extension GIF), all outputs from the Safety folder (beginning with T and extension LST) and all outputs from the Listings folder (beginning with L and extension LST), with code to generate a table of contents showing the first two lines of the title for each output included. Code is also included for the table of contents to contain a bookmarked link to the associated output in the final Word document.

As none of the page set-up variables have been specified, default values will be used.

PhUSE 2007

THE MACRO FUNCTIONALITY

This is the basic process map of how the macro works:

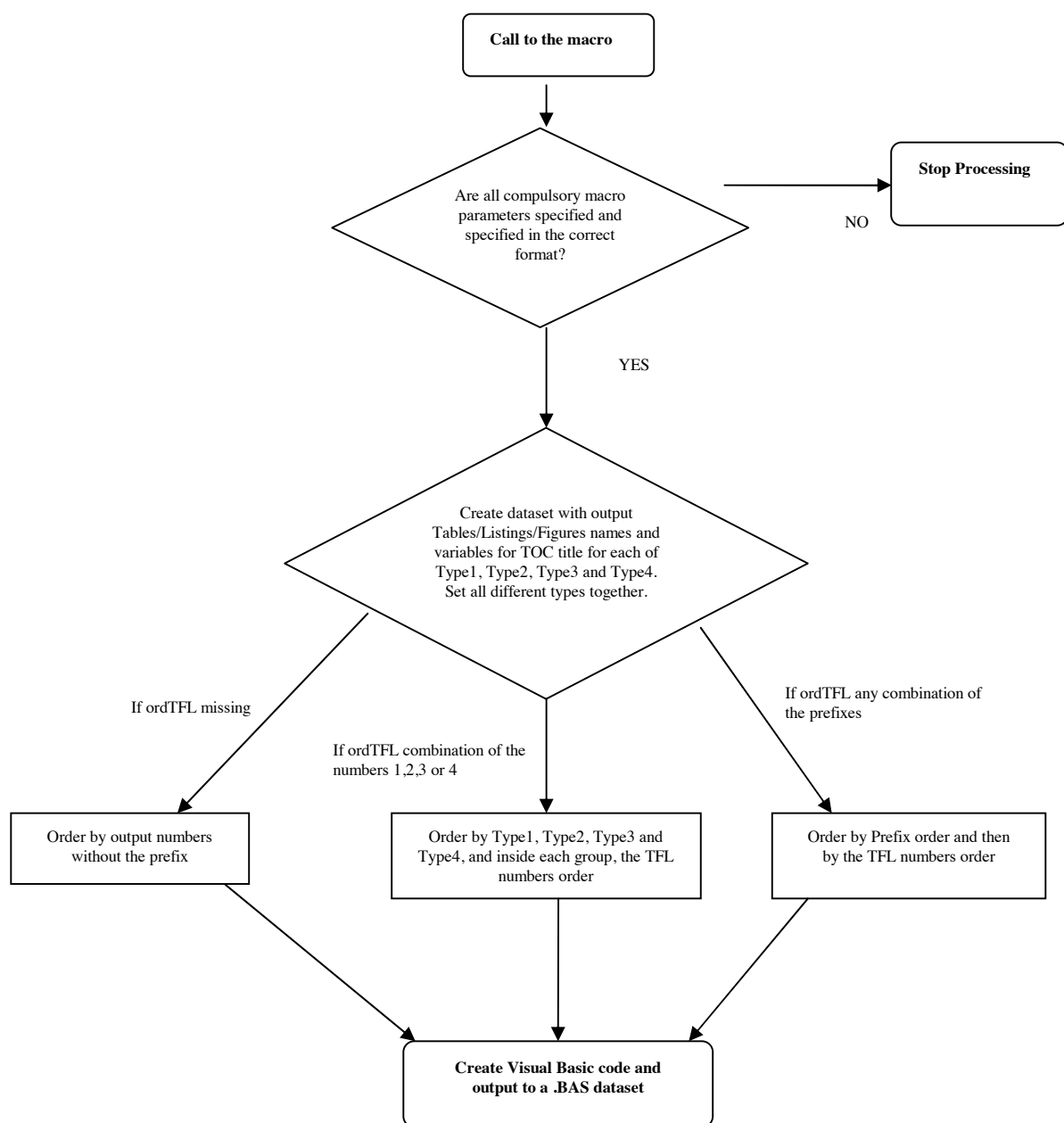


Figure (3)

When the macro is called, the program first checks each variable past in, and whether default values should be assigned. A typical check would be checking the data type of a variable, for example:

```
%if "&suffix1" ^= "" and %datatype(&suffix1) = NUMERIC %then %do;  
  %put ERROR: SUFFIX1 IS SPECIFIED AS NUMERIC, THIS SHOULD BE A VARIABLE NAME;  
  %goto exit;  
%end;
```

When all parameters have been checked and are correct, the program will build a dataset of all output file names and titles that are required, and will do any ordering specified in the macro call. The titles are taken from the titles dataset as set up by the titles and footnotes macro. These are transposed so that each observation in the new dataset represents one output and has all associated titles in COL1 COL2, etc columns, as many as specified in the TOC parameter passed into the macro.

PhUSE 2007

VIEWTABLE: Work.Dset					
	typesuf	name	tfl_no	COL1	COL2
1	Safety	T1.lst	T1	TABLE 1: NUMBER OF DEATHS IN THE STUDY	SAFETY POPULATION
2	Safety	T2.lst	T2	TABLE 2: SUMMARY STATISTICS OF ORGAN FAILURES BY TREATMENT	SAFETY POPULATION
3	Safety	T3.lst	T3	TABLE 3: ENROLMENT OF PATIENTS BY CENTRE AND MONTH	ALL ENROLLED PATIENTS

Figure (4)

If the table of contents is specified (using a flag, TOC) then the program will generate the following bookmarked links for each title at the file insertion point (to be used by the table of contents) by creating an array of titles and outputting script for each, as:

```
if toc = 1 then do;
  length numb $6.;
  array co {*} col: ;
  do j=1 to dim(co);
    if co(j) ^= '' then do;
      code = "Selection.Fields.Add Range:=Selection.Range, Type:=wdFieldEmpty,
        Text:= _";
      output;
      numb = " \1 " || trim(left(put(j,best.))) || " ";
      code = ""TC "" || quote(trim(left(co[j]))) || quote(numb) || ",
        PreserveFormatting:=False";
      output;
    end;
  end;
end;
```

It will then need to generate VBScript code for inserting each of the outputs. As the code for inserting images is a little different to text files, the program first checks if the type of output is a figure (from the *SUFF* input parameter above), otherwise continue with text file outputs, as:

```
/* For figures */
if fig = 1 then do;
  code = "Selection.Style = ActiveDocument.Styles("Plain Text")";
  output;
  code = "Selection.InlineShapes.AddPicture FileName:= _";
  output;
  code = ""&study.\&env.\Programs\" || left(trim(typesuf)) || "\Output\" ||
    left(trim(name)) || "" _";
  output;
  code = ", LinkToFile:=False, SaveWithDocument:=True";
  output;
end;
/* For tables and listings */
else do;
  code = "Selection.InsertFile FileName=""&study.\&env.\Programs\" ||
    left(trim(typesuf)) || "\Output\" || left(trim(name)) || "" _";
  output;
  code = ", Range:= "", ConfirmConversions:=False, Link:=False,
    Attachment:=False";
  output;
end;
if lastrec ne 1 then do;
  code = "Selection.InsertBreak Type:=wdPageBreak";
  output;
end;
```

This is the main SAS code for generating the VB script code for inserting each file, to be used in running the Word macro. Other than this, it is just a matter of outputting the code for page set-up, font type and font size, and for creating the actual table of contents using the links already embedded by the code above.

PhUSE 2007

INSERTING OUTPUT FILES AND GENERATING THE TABLE OF CONTENTS

This section details the VBScript code that is generated by the macro using the input parameters. The Process Logic of this macro has the following steps to write the code in the BAS file:

Step 1: To write the code in the BAS file to format the Word document. For example, the following code could be written by default if the setup parameters are not specified in the macro call:

```
With ActiveDocument.PageSetup
    .PaperSize = wdPaperA4
    .LineNumbering.Active = False
    .Orientation = wdOrientLandscape
    .TopMargin = CentimetersToPoints(2.5)
    .BottomMargin = CentimetersToPoints(2.5)
    .LeftMargin = CentimetersToPoints(2.5)
    .RightMargin = CentimetersToPoints(2.5)
    .Gutter = CentimetersToPoints(0)
    .HeaderDistance = CentimetersToPoints(1.25)
    .FooterDistance = CentimetersToPoints(1.25)
    .FirstPageTray = wdPrinterDefaultBin
    .OtherPagesTray = wdPrinterDefaultBin
    .SectionStart = wdSectionNewPage
    .OddAndEvenPagesHeaderFooter = False
    .DifferentFirstPageHeaderFooter = False
    .VerticalAlignment = wdAlignVerticalTop
    .SuppressEndnotes = False
    .MirrorMargins = False
    .TwoPagesOnOne = False
    .GutterPos = wdGutterPosLeft
End With
```

Step 2: Write the code in the BAS file to insert each output file with table of contents. For example, with title lines 1 and 2:

```
Selection.Fields.Add Range:=Selection.Range, Type:=wdFieldEmpty, Text:= _
"TC ""TABLE 1: NUMBER OF DEATHS IN THE STUDY"" \l 1 ", PreserveFormatting:=False
Selection.Fields.Add Range:=Selection.Range, Type:=wdFieldEmpty, Text:= _
"TC ""SAFETY POPULATION"" \l 2 ", PreserveFormatting:=False
Selection.InsertFile
FileName:="S:\SASData\Orion\234_010\Development\Programs\Safety\Output\T1.lst" _
, Range:="", ConfirmConversions:=False, Link:=False, Attachment:=False
Selection.InsertBreak Type:=wdPageBreak
```

Step 3: Write the code in the BAS file to select the font name and font size for the whole document (and have default set up as in step 1):

```
Selection.HomeKey Unit:=wdStory
Selection.WholeStory
With Selection.font
    .Name = "Courier New"
    .Size = 7
    .Bold = False
    .Italic = False
    .Underline = wdUnderlineNone
    .UnderlineColor = wdColorAutomatic
    .StrikeThrough = False
    .DoubleStrikeThrough = False
    .Outline = False
    .Emboss = False
    .Shadow = False
    .Hidden = False
    .SmallCaps = False
    .AllCaps = False
    .Color = wdColorAutomatic
    .Engrave = False
    .Superscript = False
    .Subscript = False
    .Spacing = 0
    .Scaling = 100
    .Position = 0
End With
```

PhUSE 2007

```
.Kerning = 0
.Animation = wdAnimationNone
End With
```

Step4: Write the code in the BAS file for the actual Table of Contents in the document. It links to the pre-defined bookmarks set-up at the time each file was inserted into the document.

```
Selection.MoveLeft Unit:=wdCharacter, Count:=1
Selection.TypeText Text:="TABLE OF CONTENTS"
Selection.TypeParagraph
With ActiveDocument
    .TablesOfContents.Add Range:=Selection.Range, RightAlignPageNumbers:= _
        True, UseHeadingStyles:=False, UseFields:=True, TableID:="C", _
        IncludePageNumbers:=True, AddedStyles:="", UseHyperlinks:=True, _
        HidePageNumbersInWeb:=True, UseOutlineLevels:=True
    .TablesOfContents(1).TabLeader = wdTabLeaderDots
    .TablesOfContents.Format = wdIndexIndent
End With
Selection.MoveUp Unit:=wdScreen, Count:=1, Extend:=wdExtend
Selection.font.Name = "Courier New"
Selection.font.Size = 7
```

USING THE .BAS FILE IN WORD

The output file from the SAS macro will be named "Insert *OUTNAME*.bas", where *OUTNAME* is the name passed to the macro. This output file should be imported into the Visual Basic Editor in a new Word document by selecting:

Tools -> Macro -> Visual Basic Editor, then File -> Import File.

To run, create a new blank document and select Tools -> Macro -> Macros and run 'Insert_*OUTNAME*'.

The new document should now contain everything as specified in the macro call.

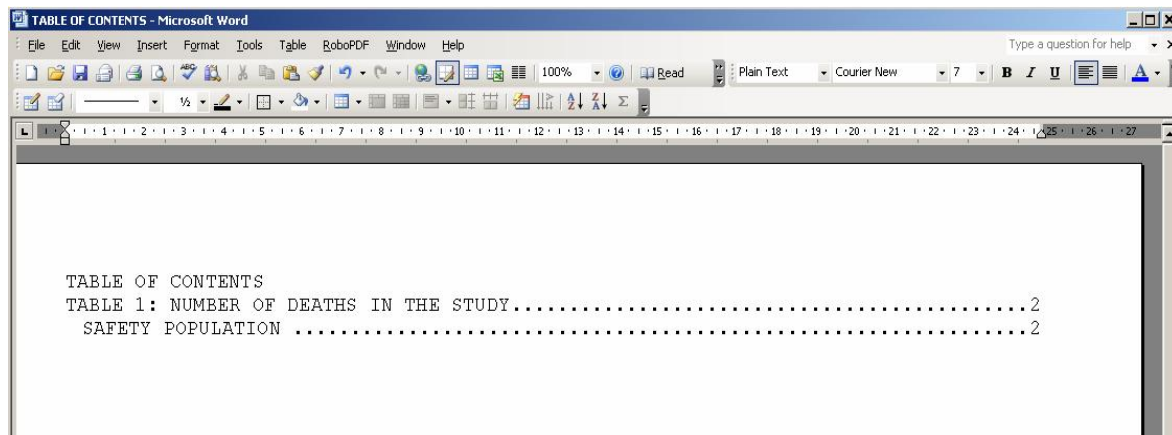


Figure (5)

CONCLUSION

In conclusion, the macro described in this paper automates the generation of a .BAS file to be run as a macro in a Word document. It presents all statistical output with a bookmarked link from a table of contents which makes it very easy for anyone reviewing to navigate between TFLs. For large projects especially, this saves a lot of time for both programmers and reviewers alike.

PhUSE 2007

REFERENCES

'Titles & Footnotes Macro': poster from PhUSE 2006 conference, by Jennie McGuirk (ICON Clinical Research).

ACKNOWLEDGMENTS

I would like to thank John Moone and Esther Bayon for their work on the macro development.

CONTACT INFORMATION

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

John Woods
ICON Clinical Research
South County Business Park
Leopardstown, Co. Dublin, Ireland
Work Phone: +353 (0) 1 291-2418
Fax: +353 (0) 1 291-2200
Email: woods@iconirl.com
Web: www.iconclinical.com

Esther Bayon
ICON Clinical Research
South County Business Park
Leopardstown, Co. Dublin, Ireland
Fax: +353 (0) 1 291-2200
Email: bayone@iconirl.com
Web: www.iconclinical.com

Brand and product names are trademarks of their respective companies.