

Generating Microsoft Word Macros that Automate the Organization and Maintenance of SAS Tables, Listings and Figures

Liz Taylor, Endo Pharmaceuticals, Chadds Ford, PA

ABSTRACT

Generally, a great deal of time and effort is expended using SAS® to generate Tables, Listings and Figures for a Clinical Study Report. Timelines are usually based only upon the development process and little thought is given to the time and effort required to organize and maintain the Microsoft® Word Documents in which the Tables, Listings and Figures are stored. Depending upon the size of the study and the number of Tables, Listings and Figures, this can represent a significant amount of time. Microsoft Word Macros are a good tool to help organize and maintain these documents.

INTRODUCTION

Most Microsoft Word users are comfortable with performing such tasks as formatting a document (defining margins, orientation, font, etc), selecting a printer and printing all or a portion of a document. For a single document, this is relatively easy to accomplish using the toolbars and menus, but if multiple documents are involved then this manual process becomes very time consuming. This paper will answer the following questions:

What is a Word Macro?

How do I create a Word Macro?

- Recording a Macro
- Reviewing and Modifying a Macro

How do I run a Word Macro?

- Using the Menu
- Placing the Macro on the Toolbar

What are some useful Word Macros?

- Format a document that will adhere to FDA Guidelines
- Combine all Tables, Listings, and Figures into a single document
- Print selected pages

How can I run Word Macros from SAS?

What is a Word Macro?

A Word Macro is similar to a SAS Macro in that it stores together a group of instructions and allows the user to execute the Macro instead of having to execute each instruction separately. The main difference is that SAS Macros contain SAS Code and Word Macros contain Visual Basic Code.

How do I create a Word Macro?

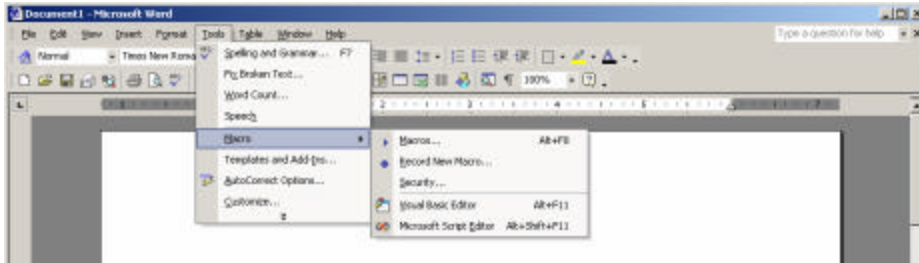
The easiest way to generate a Word Macro is to record the keystrokes that are necessary to perform the desired task and translate them into VBA statements that are stored in a Subroutine. Although this is an easy way to create a Macro, it requires some planning ahead of time to eliminate the possibility of adding unwanted keystrokes into the Macro. On the other hand, once a Macro is generated, the VBA editor can be used to edit or correct the VBA statements.

The Word Macros that are discussed here are usually generated by recording a Macro to provide the initial VBA statements which then can be modified by editing the VBA Code. Even if you are not a VBA programmer, it is possible to create useful Macros.

- **Recording a Macro**

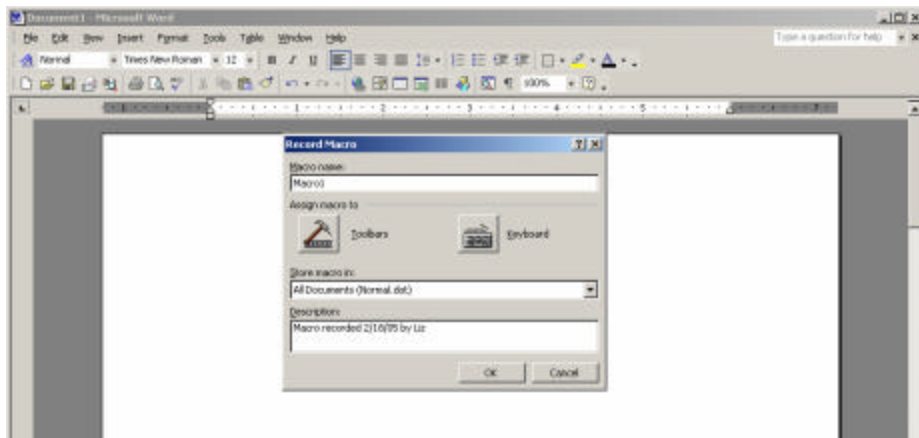
- From the Main Toolbar: select Tools, Macro and Record New Macro to bring up the Record Macro Screen. (See Figure 1)

Figure 1



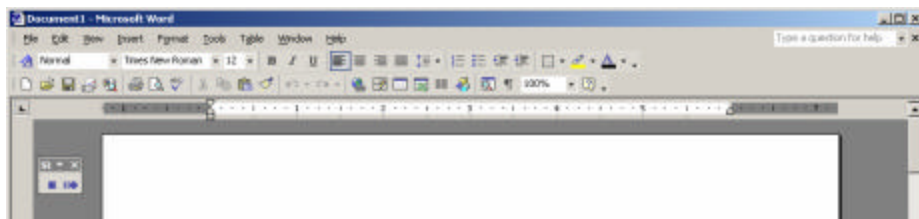
- In the Record Macro Screen: (See Figure 2)

Figure 2



- Define the Macro Name by using the Word generated Macro Name or entering a new Macro Name (It can be changed later).
 - If the name you enter is the same as an existing Word Macro, you will be prompted whether or not to replace the existing Macro.
 - You can assign the Macro to a Toolbar or Shortcut Keys or wait until after the Macro has been defined.
 - Define where the Macro is to be stored.
 - Storing the Macro in normal.dot makes it available to all documents generated using the default template.
 - Use the Word generated Description or enter a new Description (It can be changed later).
 - Click the OK Button to begin recording any subsequent keystrokes.
- A small window will appear during the recording session to indicate that keystrokes are being recorded. (See Figure 3)

Figure 3

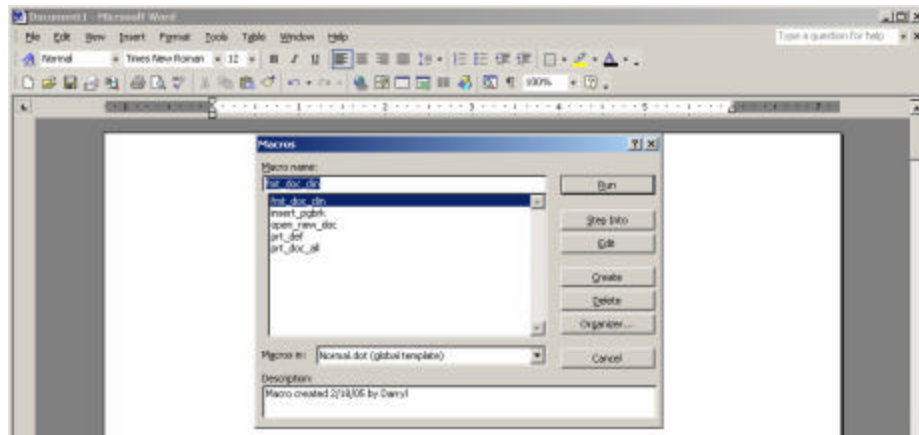


- Click on the leftmost icon to stop recording.
- Click on the right most icon to pause.

- **Reviewing and Modifying A Macro**

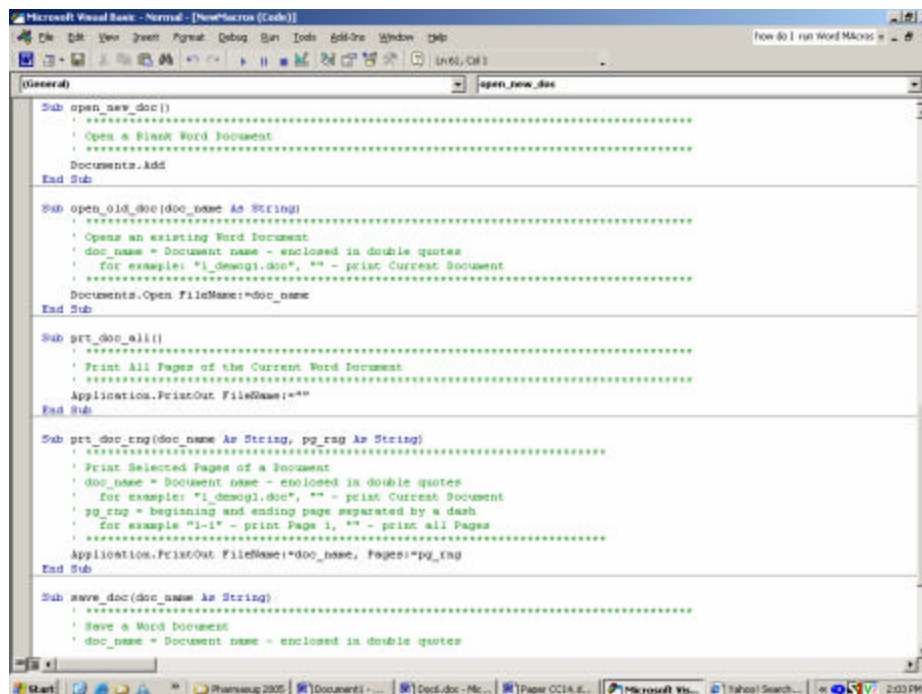
- From the Main Toolbar, select Tools, Macro and then Macros to bring up the Macros Screen. (See Figure 1)
- In the Macros Screen: (See Figure 4)

Figure 4



- Select a Macro Name from the list of Macros.
 - Click the Edit Button to open the VBA Editor.
- In the VBA Editor: (See Figure 5)

Figure 5

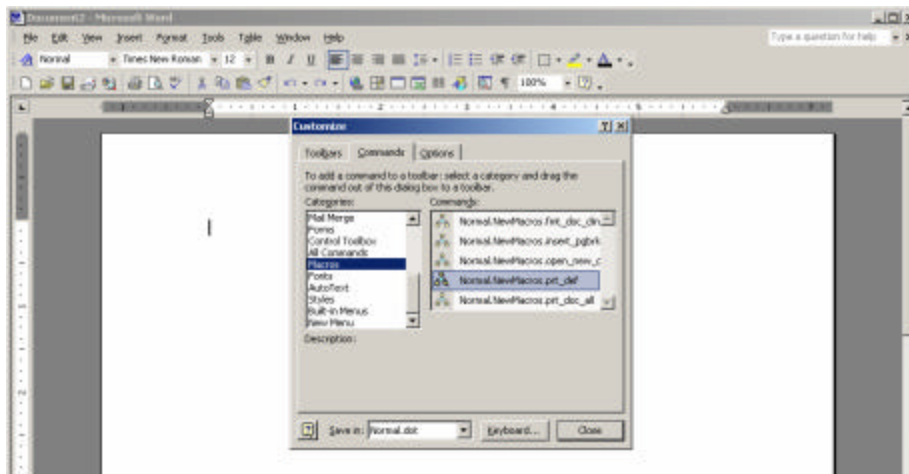


- Review VBA statements and modify as desired
 - Eliminate statements generated by unwanted keystrokes
 - Add Run-time parameters
- Add a new Macro
- Click on File/Close and Return to Microsoft Word to close the Editor and return to Word

How do I run a Word Macro?

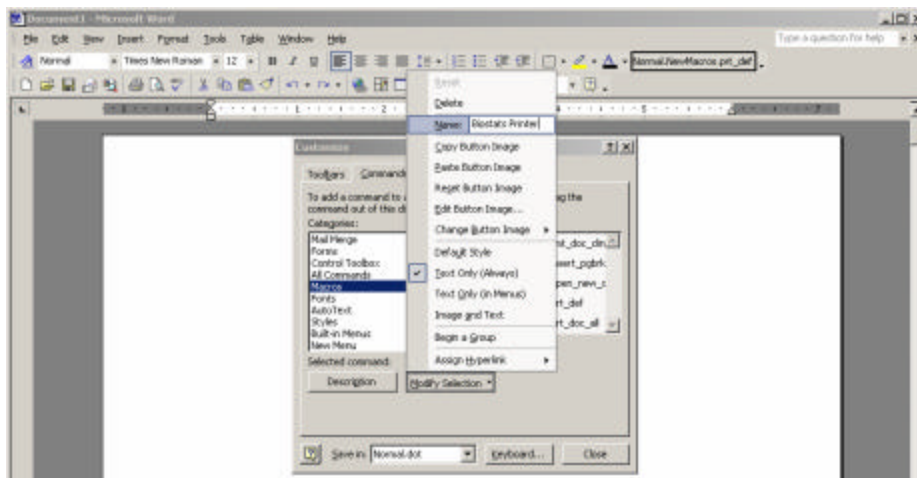
- **Using the Menu**
 - From the Main Toolbar, select Tools, Macro and then Macros to bring up the Macros Screen. (See Figure 1)
 - In the Macros Screen: (See Figure 4)
 - Select a Macro Name from the list of Macros.
 - Only stand-alone Macros that do not require run-time parameters will be displayed
 - Click the Run Button to execute the Macro.
- **Placing a Macro on the Toolbar**
 - Adding a Macro to the Toolbar makes the Macro more easily accessible
 - Macros can be grouped together into a Menu
 - From the Main Toolbar, select Tools and Customize to open the Customize Screen.
 - In the Customize Screen:
 - Click on the Commands Tab, Scroll through the Categories and Click on Macro to display all the Macros in the Commands column.
 - Select a Macro from the Commands Column and Drag and Drop it to the Toolbar. (See Figure 6)

Figure 6



- The Macro will be added to the Toolbar and identified by the Macro Name.
 - To change the identifier, click on the Modify Selection Button and modify the Name. (See Figure 7)

Figure 7



- Click the Close Button to return to Microsoft Word.
- On the Main Toolbar, the Macro will be displayed. (See Figure 8)

Figure 8



What are Some useful Word Macros

These Macros were generated by recording the Keystrokes and then editing the Macro to add Comments and define Run-time Parameters that will be passed to the Subroutine.

- Format a Document that will adhere to FDA Guidelines

```
Sub fmt_doc_clin()
' *****
' Format a Document in the Clinical Style
' (1" Margins, Landscape, Courier New 8)
' *****
With Selection.PageSetup
    .Orientation = wdOrientLandscape
    .TopMargin = InchesToPoints(1)
    .BottomMargin = InchesToPoints(1)
    .LeftMargin = InchesToPoints(1)
    .RightMargin = InchesToPoints(1)
End With
Selection.WholeStory
With Selection.Font
    .Name = "Courier New"
    .Size = 8
End With
End Sub
```

- Combine Listings into a single document

```
Sub insert_doc(bkmrk_name as String, doc_name As String)
' *****
' Insert a File into a Document, Bookmark it, & add a Page Break
' bkmrk_name = Bookmark identifier
'   for example: "L_1"
' doc_name = Document name
'   for example: "l_demogl.doc"
' Sample Call: insert_doc "L_1", "l_demogl.doc"
' *****
With ActiveDocument.Bookmarks
    .Add Range:=Selection.Range, Name:=bkmrk_name
Selection.InsertFile FileName:=doc_name
Selection.InsertBreak Type:=wdPageBreak
End Sub

Sub en3228_comb_lst()
' *****
' Combine Listings into a single Document
' *****
' Change the Default Directory
chg_dir "x:\biostats\en3228\output"

' Open a New Document
Documents.Add
```

```

' Insert Files in order by Appendix Number
insert_doc "L_1", "l_demogl.doc"
insert_doc "L_2", "l_displ.doc"
insert_doc "L_3", "l_incl1.doc"
insert_doc "L_4", "l_excl1.doc"
insert_doc "L_5", "l_medh1.doc"
insert_doc "L_6", "l_cmed1.doc"

' Save Document
ActiveDocument.SaveAs FileName:= "en3228_lst.doc"
End Sub

```

- **Print Selected Pages**

```

Sub prt_def()
' *****
' Define a Default Printer
' *****
ActivePrinter = "\\printers\Clinical BW"
End Sub

```

```

Sub prt_biostats()
' *****
' Define a Printer
' *****
ActivePrinter = "\\printers\Biostats"
End Sub

```

```

Sub prt_doc_rng(doc_name As String, pg_rng As String)
' *****
' Print Selected Pages of a Document
' doc_name = Document name - enclosed in double quotes
'   for example: "l_demogl.doc"; "" - print Current Document
' pg_rng = beginning and ending page separated by a dash
'   for example "1-1" - print Page 1; "1-9999" - print all Pages
' Sample Call: prt_doc_rng "l_demogl.doc", "1-1"
' *****
Application.PrintOut FileName:=doc_name, Pages:=pg_rng
End Sub

```

```

Sub en3228_prnt_lst()
' *****
' Print selected Pages
' *****
' Change the Default Directory
chg_dir "x:\biostats\en3228\output"

' Define the Printer
prt_biostats

prt_doc_rng "l_demogl.doc", "1-5"
prt_doc_rng "l_displ.doc", "1-5"
prt_doc_rng "l_incl1.doc", "1-5"
prt_doc_rng "l_excl1.doc", "1-5"
prt_doc_rng "l_medh1.doc", "1-5"
prt_doc_rng "l_cmed1.doc", "1-5"

' Re-Define the Default Printer
prt_def
End Sub

```

- Additional Macros

```

Sub save_doc(doc_name As String)
' *****
' Save a Word Document
' doc_name = Document name
'   for example: "l_demogl.doc"; "" - print Current Document
' *****
ActiveDocument.SaveAs FileName:=doc_name
End Sub

Sub open_new_doc()
' *****
' Open a Blank Document
' *****
Documents.Add
End Sub

Sub open_old_doc(doc_name As String)
' *****
' Opens an existing Document
' doc_name = Document name - enclosed in double quotes
'   for example: "l_demogl.doc"; "" - print Current Document
' *****
Documents.Open FileName:=doc_name
End Sub

Sub prt_doc_all()
' *****
' Print All Pages of the Current Document
' *****
Application.PrintOut FileName:=""
End Sub

Sub chg_dir(dir_name As String)
' *****
' Change default Directory
' dir_name = Directory Path
' *****
ChangeFileOpenDirectory dir_name
End Sub

Sub insert_pgbrk()
' *****
' Insert a Page Brk into a Document
' *****
Selection.InsertBreak Type:=wdPageBreak
End Sub

```

How can I run Word Macros from SAS?

If you don't use ODS to generate your Tables, Listings and Figures, then it may be necessary to convert your SAS Output into a Word Document. This usually involves jumping from SAS to Word, reading in the SAS Output, formatting it, printing it and saving it. Word Macros may make this process easier when working with multiple documents, but it is still time consuming.

Ideally, it would be nice to be able to execute these functions automatically and without having to leave SAS.

The following simple SAS Macro uses DDE to send Word Basic Commands to Word allowing the user to Open SAS Output and then format, print, and save it as a Word Document.

Note: Word must be open in order for the SAS Macro to execute.

```

%macro sas2word (dsn_sas=, dsn_wrd=, wrd_fmt=, wrd_prt);

  /* ***** */
  /* Parameters */
  /*   dsn_sas = fully qualified name of SAS List File (including extension */
  /*             and not enclosed in quotes) */
  /*   dsn_wrd = fully qualified name of Word File (including extension */
  /*             and not enclosed in quotes) */
  /*   wrd_fmt = name of a Word Macro that formats a Document */
  /*   wrd_prt = name of a Word Macro that Prints a Document */
  /* Sample Call */
  /*   %sas2word (dsn_sas=x:\biostats\en3228\output\l_demoq1.lst, */
  /*             dsn_sas=x:\biostats\en3228\output\l_demog1.doc, */
  /*             wrd_fmt=fmt_doc_clin, wrd_prt=prt_biostats); */
  /* ***** */

  filename wordcmd dde "winword|system";

  data _null ;
    file wordcmd;

    /* Open SAS Print File */
    put '[fileopen .name = "'@;
    put "&dsn_sas"@;
    put '"]';
    /* Run the Word Format Macro to perform the Print Setup */
    put '['@;
    put "&wrd_fmt"@;
    put ']';
    /* Run the Word Print Macro to Print */
    put '['@;
    put "&wrd_prt"@;
    put ']';
    /* Save SAS Print File as a Word Document */
    put '[filesaveas .name = "'@;
    put "&dsn_wrd"@;
    put '"]';
    /* Close File */
    put '[fileclose]';
  run;
%mend;

```

CONCLUSION

Word Macros that can format, save and print Tables, Listings and Figures can represent a significant time savings. Organizing the Final Documents into a single Document using the Word Macros makes the procedure easier especially if the procedure has to be repeated because of modifications.

CONTACT INFORMATION

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

Liz Taylor
 Endo Pharmaceuticals Inc.
 100 Endo Blvd.
 Chadds Ford, PA 19317

Phone: (610) 558-9800 x4268
 Email: taylor.liz@endo.com