

## **Using SAS® to Control the Post Processing of Microsoft® Documents**

**Nat Wooding, J. Sargeant Reynolds Community College, Richmond, VA**

Chen, SUGI 31, showed how to use SAS and VBA to automate the post processing of MS Word ® documents. This paper uses different SAS coding to control a modified version of Chen's code to convert RTF files into Word documents and discusses other post processing activities that might be useful. The target audience is MS Windows® users.

### **Introduction**

Recently, a client needed a number of SAS tables which I produced and then used ODS to create Rich Text Format (RTF) output. However, when she forwarded these tables to a colleague, she found that the colleague was using a Macintosh® computer that for some reason could not read RTF files. While I could, of course, have opened each of these files in MS Word and then saved it as a Word document, I knew that I would be producing dozens more RTF files like these and I did not want to go through this manual process each time that I produced a set. Plus, coding a solution is a lot more fun than doing multiple manual file opens and saves. Ling Y. Chen, SUGI 31, offered a possible solution which I was able to build upon.

### **The Overview of the Process**

The solution to this problem consists of several steps:

- 1) Create a set of documents in RTF format and store these in a folder.
- 2) In a Word session, record a VBA macro that will perform a "Save As" operation saving the RTF as a Word document in the Word format that you desire. This involves all of the steps needed to make changes to the document. In my case, it was simply to do a file "Save As" operation and then close the file. Since the recorded macro shows the name of the file that was open during the macro recording process, the macro must be edited so that there are additional lines that store the name of whichever file is currently open and the name of the file that is being saved uses the stored information.
- 3) Read the names of the RTF files stored in the folder using SAS. SAS then uses CALL EXECUTE statements to generate the necessary commands that will open each file, one after the other, and tell Word to use the VBA macro to save each RTF as a Word document. The VBA macro will close the file.

### **Recording the Macro**

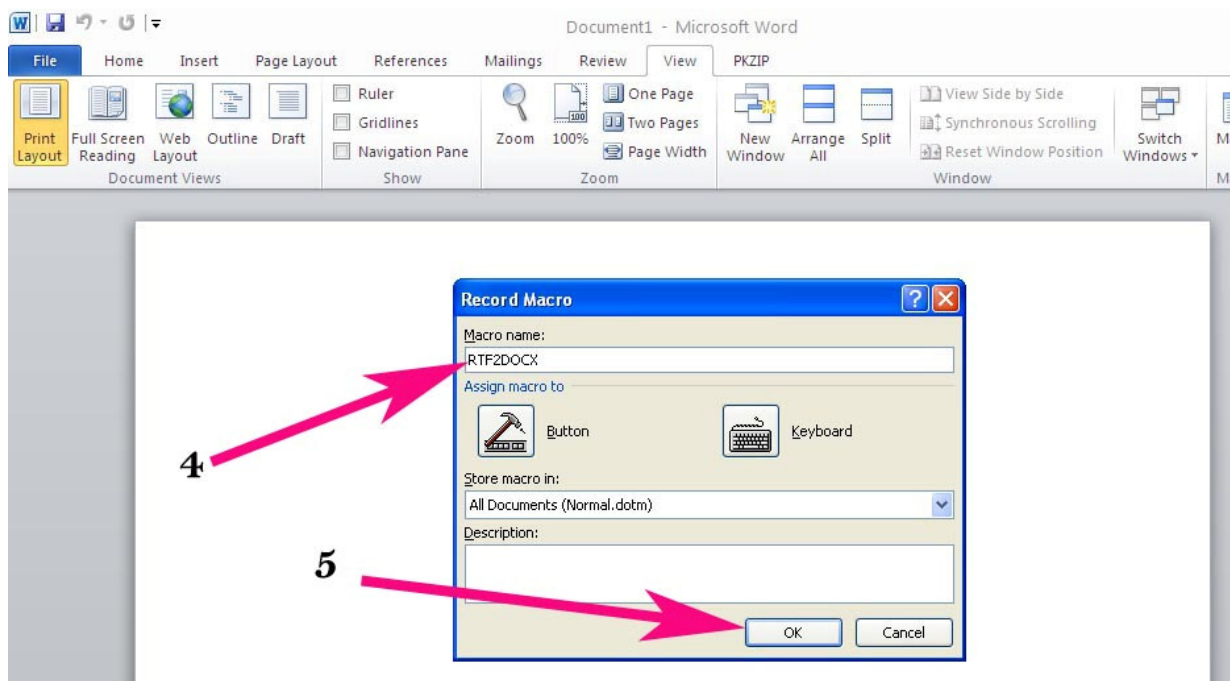
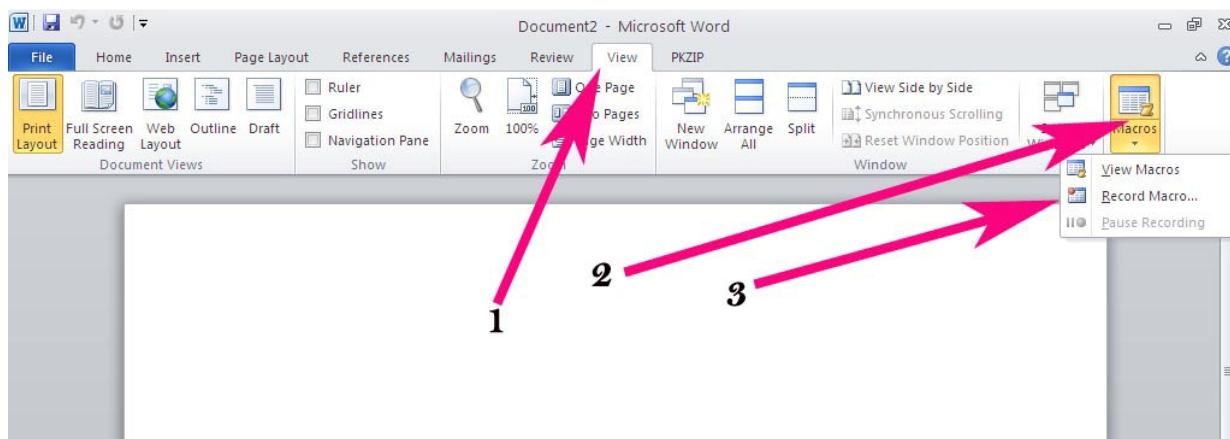
The procedure shown here is based on Microsoft Office 2010 ®. Other versions of Office may appear slightly differently in terms of where the Word tools such as recording macros are located and what some commands are called.

As shown in Figure 1, the first steps involved in recording the macro are:

- 1) Click the View tab
- 2) Click the down arrow below the Macros tab on the top right of the screen
- 3) Click Record Macro
- 4) Name this macro RTF2WORD or whatever else you please
- 5) Click the O.K. button at the bottom of the window

At this point, Word will record our actions so we go through the next steps.

**Fig 1.**  
**The First Steps of Recording a VBA Macro**



- 6) Go through the steps "File", "Save As", Browse to find the proper folder, Select the Word format that is appropriate for your system, Click "Save", Close the document but DO NOT CLOSE WORD.
- 7) Stop the macro recording session by clicking on the Stop Recording command in the Macro recording window.
- 8) Manually edit the macro as described below and shown in Figure 2.

When you click on the Macros down arrow and then click View Macros, you should see your macro listed. Select "Edit" and your macro should have at the top Sub RTF2WORD() and you should see the steps that you recorded.

At this point, you need to edit your macro by adding the following lines at the bottom of the macro

#### **Dim Str As String**

**Str = ActiveDocument.Path & "\" & Left(ActiveDocument.Name, (Len(ActiveDocument.Name) - 3))**

These are needed to capture the name of the file that we have open and place it in a string which will be used by the VBA macro to identify the file that is being saved.

Also note the field that reads something like

**ActiveDocument.SaveAs FileName:= "MYFILE.DOCX",**

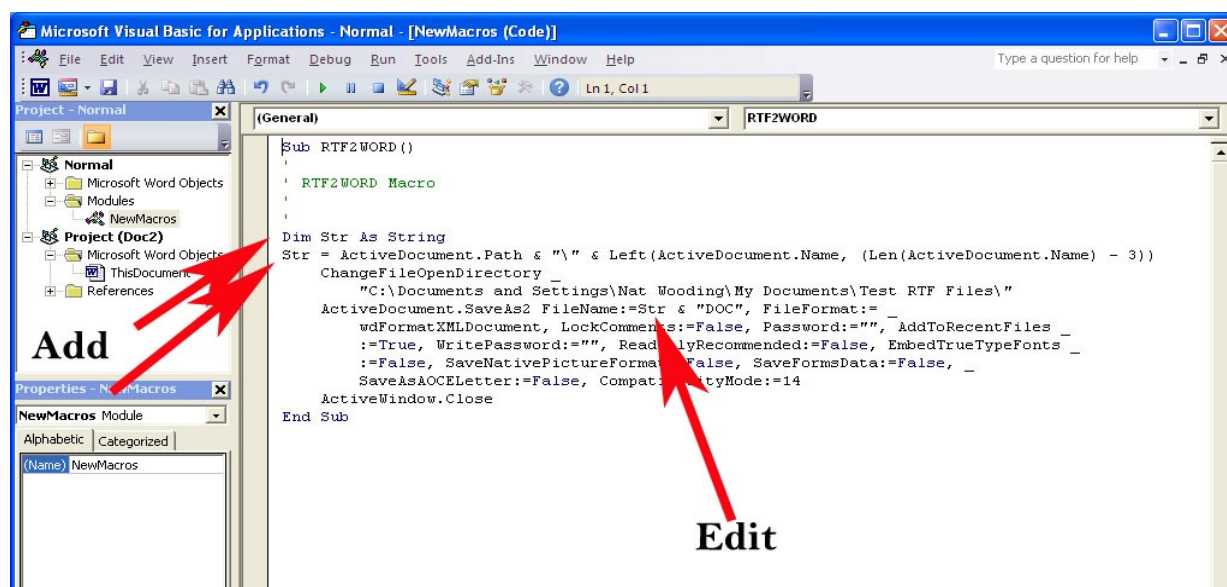
You need to edit this to read

**ActiveDocument.SaveAs FileName:=Str & ".DOCX"**

so that it will use the name of our open file instead of the name of the file that we saved when we recorded the macro.

Now, save this macro and close the Macro window.

**Fig 2.**  
**The Edited VBA Macro**



## **Now SAS Gets Involved**

The full code is shown in Appendix 1.

We will use SAS to gather a list of the files that we want to convert and then build a set of commands that will open each file in sequence and have Word execute our macro.

First, SAS issues a command to open a copy of Word.

To gather the list of RTF documents in our directory, we will use a pipe to create a list that we can read with a data step.

First, create a macro variable that will identify the directory name and the MS DOS® commands to list the names of the files in the directory.

```
%LET RTFDIR =dir /s/b C:\RTFS\ ;
*                ^^^^^^^^^^^^^^* This is the directory to be searched;
```

**Filename Myfiles Pipe "&rtfdir" Lrecl=32767;**

Now, read the pipe and extract the name of the path.

```
Data Listnames ;
  Infile myfiles Truncover;
  Input Line $char2000. ;
  Type = Scan( line , -1 , '.' ) ;    * Extract the file extension;
  If Ucase( Type ) = 'RTF'; * Exclude any other files that might have crept into the folder;
  Outname = Scan( Line , -1 , '\' ) ; * This is the name of a file;
  Line = left( Reverse( Line ) ) ;
  Pathname = ( Reverse( Substr( Line , Length( Outname ) + 2 ) ) || '\' || Outname ) ; ;
  Keep Pathname;
Run;
```

Now, we must use our list of file names to create a set of SAS commands that will open each RTF successively and then save it using the macro that we recorded. The code uses Direct Data Exchange (DDE) to all SAS to communicate with our Word session. DDE is a Microsoft feature that allows one program to communicate with another program that is also running. Also, we use the SAS Call Execute function to generate a data step with a series of SAS statements that will issue the commands to Word to cause each RTF to be opened, saved, and closed in sequence. The data step generated by Call Execute will be automatically run at the end of our \_NULL\_ data step.

```
Filename Cmnds DDE 'Winword|System';
Data _Null_ ;
Set Listnames End = Eof ;
```

```
Length Str1 - Str2 $200.;
If _n_ = 1 Then
Call Execute( "Data _null_ ; File Cmnds; " );* start a data step with a file statement that will communicate
with a Word session using DDE;
```

```
FirstPart = " Put '[FileOpen.Name=" ; * Create a VBA statement that will open a file;
Pathname = CATS ( "" , Pathname , "" ] ";);
Str1 = ( FirstPart || Pathname );
```

```
Call Execute( str1); *We have built the command to open an RTF doc;
```

```
S1 = "put '[ToolsMacro .Name = " ;
S2= "RTF2Wrd" ;
S3 = ",.Run]";" ;
Str2 = CATS( s1 , "" , s2 , "" , s3);
Call Execute( str2 ); * Now invoke the macro to save it in the format used by our copy of Word;
```

```
If Eof Then Call Execute( 'Run;'); * End this data step;
```

**Run;**

Note that there will be a FileOpen and Word macro invocation for each RTF stored in Listnames set. The process here is that when the SAS statements generated by the Call Execute functions run, they generate a set of Put statements that are written to the copy of Word that SAS opened. These tell Word to open the particular file and then invoke our VBA macro which will save the file as a Word document and then close the file. We will see each file open and close.

**GOING BEYOND THE CURRENT PROBLEM**

Chen needed to save a Word file to RTF format and he also needed to make changes to the document's layout and formatting. In my case, I simply needed to change the format from RTF to MS Word and I wanted to do this to all of the RTF files in a folder and I did not want to manually create the statements needed to convert the files. Going beyond my simple need, the possibilities of having SAS make use of the features of Microsoft products are limited only to the processing options offered by the particular product that you wish to use. Remember, the goal is to be able to post process multiple files with the least work possible. Here are some possible uses:

1. Change the font in a Word document. This might be useful if you have recipients who do not have some particular font that others users demand.
2. Make global changes to groups of documents. For example, you could have a set of documents that have references to a particular person or department and the names could change over time. A global find/replace would be useful here.
3. Saving a set of Microsoft Excel® worksheets as CSVs.
4. Use a macro to update Microsoft Access ® tables.

**CONCLUSIONS**

Some of the products in the Microsoft Office suite have built in functions that SAS can take advantage of to perform useful tasks. If the execution of these tasks can be stored in a VBA macro or even old Excel macro commands, then it is possible (at least in the cases shown here), to make use of them from a SAS session. This can allow a SAS user to perform repetitive applications on Microsoft files with minimal manual work on his or her part. Furthermore, in the case that I presented, I was able to use a Microsoft product that I already had and I did not have to purchase third-party file conversion software.

**REFERENCES**

Chen, Ling Y., 2006 Converting Multiple SAS Output Files to Rich Text Format Automatically without Using ODS, SUGI31, pp. 131-133

SAS 9.2 Language Reference: Dictionary, Fourth Edition, SAS Institute, Cary, N.C. 2011

**ACKNOWLEDGEMENTS**

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.

Other brand and product names are registered trademarks or trademarks of their respective companies.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please feel free to contact the author at:

Nat Wooding

Office of Institutional Effectiveness

J. Sargeant Reynolds Community College

1651 East Parham Road

Richmond, VA 23228

Email: [Nwooding@reynolds.edu](mailto:Nwooding@reynolds.edu)

or [Nathani@verizon.net](mailto:Nathani@verizon.net)

## APPENDIX 1

\* RTF2Word;

\* Nat Wooding

\* Office of Institutional Effectiveness

\* J. Sargeant Reynolds Community College

\* 804-523-5884

\* [nwooding@reynolds.edu](mailto:nwooding@reynolds.edu)

\* January, 2012.

\* Purpose: Convert all of the RTF files in a folder into .doc or .docx files

\* Tools: SAS, Word Macro, Visual Basic (which is what Word uses) ;

\*

\* Getting started:

1) Create a folder where you will store your RTF files. The code below uses C:\RTFS. If you wish to use a different name, you will need to change the following SAS code and you will need to use that folder name when you record the Word macro in step 2.

2) Record a MS Word macro using the name RTF2WRD. To do this, in a Word session, do the following

a) Click Tools

b) Select Macro

c) Record New Macro

d) Name this macro RTF2WRD

e) Go through the steps "File", "Save As", Browse to find the proper folder, Select the Word format that is appropriate for your system, Click "Save", Close the document but

DO NOT CLOSE WORD.

f) Stop the macro recording session by clicking on the stop icon in the Macro recording window.

3) Your macro should resemble the following one:

```
Sub RTF2WRD()
```

```
,
```

```
' RTF2WRD Macro
```

```
' Macro recorded 2/11/2012 by Nat Wooding
```

```
,
```

```
' The first two lines were entered manually
```

```
,
```

```
Dim Str As String
```

```

Str = ActiveDocument.Path & "\" & Left(ActiveDocument.Name, (Len(ActiveDocument.Name) - 3))
ChangeFileOpenDirectory "C:\RTFS\"
ActiveDocument.SaveAs FileName:=Str & "DOC", FileFormat:=wdFormatDocument _
, LockComments:=False, Password:="", AddToRecentFiles:=True, _
WritePassword:="", ReadOnlyRecommended:=False, EmbedTrueTypeFonts:=False, _
SaveNativePictureFormat:=False, SaveFormsData:=False, SaveAsAOCELetter:= _
False
ActiveWindow.Close
End Sub

```

You now need to edit the macro (Tool, Macro, Macros, highlight RTF2WRD and click Edit)  
 Add the lines that start "DIM" and "STR". The line that starts "ActiveDocument" will read  
 ActiveDocument.SaveAs FileName:="Doc1.doc", FileFormat:=wdFormatDocument, \_

Change FileName:="Doc1.doc" to read FileName:=Str & "DOC", [or DOCX]  
 Doing this will cause the new file to have the same name as the old one but with a DOC or DOCX extension.

4) Save the changes.

\* REF: Converting Multiple SAS? Output Files to Rich Text Format Automatically without Using ODS  
 Ling Y. Chen, Rho, Inc., Newton, MA, SUGI31, 131-33

```

;
options symbolgen NOXWAIT NOXSYNC;

```

\* Start a Word session. If there is an open Word session, do not run the next statement;

```

X "C:\Program Files\Microsoft Office\Office11\WinWord.EXE";

```

\* Scan the directory and capture the names of all of the RTF files in it;  
 \* First, define a macro variable, RTFDIR, and in it place the name of the folder where the RTF files are stored. The path with this folder MUST match the folder that you used when you recorded the Word macro. The first part of the RTFDIR are DOS commands which will list the contents of the folder;

```

%LET RTFDIR =dir /s/b C:\RTFS\ ;
*          ^^^^^^^^^^^^^^ This is the directory to be searched;

```

\* Use a pipe to make the names of the files in the folder available to your session. Note that any files that do not end in .RTF are ignored in the data step.;

```

filename myfiles pipe "&rtfdir" lrecl=32767;
data Listnames ;
infile myfiles trunccover;
input line $char2000. ;

```

```

Type = scan( line ,-1, '.');
If UPCASE( Type) ='RTF';
Outname = scan( line ,-1, '\');
Line = left( Reverse(line)) ;

```

```

Pathname = (Reverse( Substr( line , Length(Outname) + 2 )) || '\' || outname ) ; ;
keep pathname;
run;

```

\* The following Data Step writes commands to your Word session telling it to open each file in  
 \* turn and then invokes the Word Macro which causes it to be saved as a DOC file or whatever  
 \* is appropriate to your version of Word.;

```
filename cmnds dde 'winword|system';
```

```
Data _Null_ ;
```

```
set listnames end = eof ;
```

```
length str1-str2 $200.;
```

```
if _n_ = 1 then
```

```
Call Execute( "Data _null_ ; file cmnds; " );* start a data step;
```

```
FirstPart = " Put '[FileOpen.Name=" ;
```

```
Pathname = CATS ( "" , Pathname , "" ] ";);
```

```
Str1 = ( FirstPart || Pathname );
```

```
Call Execute( str1); *We have built the command to open an RTF doc;
```

```
s1 = "put '[ToolsMacro .Name = " ;
```

```
s2= "RTF2Word" ;
```

```
s3 = ",.Run]";
```

```
str2 = CATS( s1 , "" , s2 , "" , s3);
```

```
Call Execute( str2 ); * Now invoke the macro to save it in the format used by our copy of Word;
```

```
if eof then Call Execute( 'run;'); * End this data step;
```

```
run;
```