

Combining RTFs While Retaining Each Original Document's Page Numbering

Jim Worley, PPD, Inc., Morrisville, NC

ABSTRACT

In clinical trials, there is an increasing need for SAS programmers to combine multiple RTF files into a single document. The main obstacle programmers face when performing this function is the maintenance of the page numbering for each individual document within the combined document. With current SAS limitations, there is no easy method to provide this functionality. This paper addresses this limitation through the use of a SAS program and a Word macro. With some minor modifications to the RTF code, programmers can quickly combine multiple files while retaining their distinctive information and page numbering within a larger document.

INTRODUCTION

Due to the confusing nature of the wording to describe this paper's objectives, here is a quick example to think about as you read through the paper. There exist two RTF documents, Paper A which contains 3 pages and Paper B which contains 4 pages. The goal is to combine Paper A and B, which is a simple task in itself. However, the problem that results is that the page numbering would be "Page x of 7" rather than the preferred "Page x of 3" and "Page x of 4", respectively. Having the final document show "of 7" is unacceptable and fixing this glitch requires an interesting approach.

Before coming up with a way to combine multiple RTFs, first it is necessary to understand the way SAS creates them. When someone creates a new page in an RTF, there are two methods to choose from: a page break or a section break. A simple page break is what most people would commonly think of, but SAS takes a different approach. When SAS creates a new page in an RTF file, it will always use a section break.

A "section" in an RTF document can be thought of as a document within a document. Each section can have unique headers and footers, as well as individual page numbering. Unfortunately, because each page that SAS outputs is its own section, some tinkering is needed to use this section page numbering to combine documents. By replacing the SAS section breaks with page breaks, we can then combine documents, separated by section breaks, and use section page numbering to retain each individual document's total number of pages.

The one other thing that needs to be considered is the way page numbers are inserted into the original RTF documents. This paper assumes that the following common page numbering code is used:

```
{Page}\-{\field{\*\fldinst {PAGE }}\-of}\-{\field{\*\fldinst {NUMPAGES }}\-}
```

The main thing to notice out of that long string of RTF commands is the keyword NUMPAGES. This will return the total number of pages in the entire document, not something we want if the plan is to have each document retain the original total number of pages after being combined. Luckily, there also exists the keyword SECTIONPAGES which we can substitute in to do what we need.

PREPARING THE RTFS

The first step in this process of combining documents is to set up the RTF files. This requires reading in a folder of RTF files, replacing certain strings of RTF code, and then outputting the documents to a new folder as to not replace the originals.

Here is the beginning of the SAS program. This code will look in a target directory and pick out all the RTF files contained within.

```
%let indir = C:\in_files;

filename filelist pipe "dir ""&indir."" /b" ;

data filelist ;
    length rtfname $256 ;
    infile filelist length=size ;
    input rtfname $varying256. size ;
run ;
```

The output of this code is a dataset called FILELIST which contains a list of all the RTFs in a column called RTFNAME. This dataset can then be looped through to access all the RTF files to be combined.

The code below assumes that each RTF file was placed into a macro variable &RTF1 - &RTF*. This could also be done in a data step loop. What happens is that SAS will open each RTF document, look for and replace two strings, and then resave the modified RTF document into a new folder.

```
%let outdir = C:\out_files;

%do i = 1 %to &rtfcnt;
data _null_;
  infile "&indir.\&&rtf&i" lrecl=32767 length = lg;
  file "&outdir.\&&rtf&i" lrecl=32767;
  input line $varying32767. lg;
  line = tranwrd(line, '{Numpages}', '{SECTIONPAGES}');
  line = tranwrd(line, '\sect\sectd\ ', '\page\par\ ');
  new_lg = length(line);
  put line $varying32767. new_lg;
run;
%end;
```

In this code the infile line specifies which RTF file is being read in, and the file line specifies where the file will be saved to. As each line is read in, tranwrd is used to replace certain strings to insert SECTIONPAGES and page breaks into the document. After this is done, the modified line is then written to the new file. This is a fast process that can go through many large documents in little time.

Now would be a good time to check the modified RTFs. If everything worked correctly there should be no apparent difference between the modified versions and the originals. Headers, footers, page numbering, and spacing should all be the exact same.

COMBINING THE RTFS

The second thing to do now is combine all of the RTF documents. To do this, we will need to copy the first RTF into a new word document, create a section break, copy in the next RTF document, set the page numbers to begin at one for that section, and then repeat for each RTF needed. Since this would take quite some time, a simple Word macro can be used to automate the process. This document assumes that the reader possesses the basic knowledge of creating a Word macro. If not, there are many resources on the web which can help.

The beginning of the Word macro will ask the user where the data resides, and then look in the directory to locate all the RTF documents.

```
Sub Combine_files()

filepath = InputBox("Name of directory with rtf files, with no ending slash! (e.g.
c:\define\nda)")

Documents(1).Close SaveChanges:=wdDoNotSaveChanges
Documents.Add DocumentType:=wdNewBlankDocument

Application.ScreenUpdating = False

Set fs = Application.FileSearch
With fs
  .LookIn = filepath
  .FileName = "*.RTF"
  If .Execute = 0 Then
    MsgBox "There were no files found."
  End If
End With
```

The first line uses an InputBox to ask the user for the directory of RTFs and then saves that information to the variable filepath. It then goes on to close the current documents and open a new blank document (which is always needed, even if Word was just opened and is sitting at a blank document). Next, the Word code turns off ScreenUpdating. This is used because it will speed up the process of opening large files tremendously. Lastly, it searches through the directory and finds all the RTF documents. These are then saved into an object variable named FS.

The rest of the code is devoted to combining all the RTFs together:

```
For i = 1 To fs.FoundFiles.Count

  Documents.Open FileName:=fs.FoundFiles(i), Visible:=True
```

```

Selection.HomeKey Unit:=wdStory
Selection.EndKey Unit:=wdStory, Extend:=wdExtend
Selection.Copy

Documents(fs.FoundFiles(i)).Close SaveChanges:=wdDoNotSaveChanges
Selection.Paste

If i < fs.FoundFiles.Count Then
    Selection.InsertBreak Type:=wdSectionBreakNextPage
End If

Next i

For Each s In ActiveDocument.Sections
    With s.Headers(wdHeaderFooterPrimary).PageNumbers
        .RestartNumberingAtSection = True
        .StartingNumber = 1
    End With
Next s

Application.ScreenUpdating = True

ActiveDocument.SaveAs FileName:=filepath & "\all.rtf", fileformat:=wdFormatRTF

MsgBox "I'm Done!"

End Sub

```

There are two loops here. The first loop using the variable 'i' will open each RTF in the target directory, copy the entire document, close that document without saving any changes, and then paste the document into the new file. The code then checks to see if there are more RTF files, if so it will create a section break before copying in the next file.

The next loop, controlled by the variable 's', starts after all the RTFs have been combined into a single document. It goes to the beginning of each section of the new document and sets the starting page number to one. Lastly ScreenUpdating is turned back on and the file is saved.

CONCLUSION

Although the idea seems simplistic in nature, maintaining original page numbering while combining RTFs can be a challenging process. This paper should be used as a starting point in understanding how SAS controls pagination. Only through this understanding can programmers make the necessary changes in order to achieve their goal.

REFERENCES

SAS OnlineDoc® documentation for SAS version 8.2.

Microsoft Developers Network, <http://www.msdn.com>.

CONTACT INFORMATION

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

Jim Worley
 PPD, Inc.
 3900N Paramount Parkway
 Morrisville, NC 27560
 Fax: (919) 654-5753
 Email: James.Worley@rtp.ppd.com

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 trademarks of their respective companies.