

No More Manual PDF Bookmarks! An Automated Approach to Converting RTF files to a Consolidated PDF with Bookmarks

Tyler Plevney, Emanate Biostats, Inc.

ABSTRACT

Preparing output documents for data review meetings and presentations can be a tedious process. What if I told you, it does not have to be? Instead of opening and closing multiple rich text format (RTF) file or Excel files with various sorts and filters to review, sometimes hundreds of outputs, why not consolidate them in an easy-to-review PDF document with all outputs easily separated with linked bookmarks? Doing this manually by converting RTF output files to PDF can take a long time. Add on the time to type out the bookmarks for each individual output and it does not seem worth it. In this paper I will present an automated way to convert an entire folder of RTF outputs to PDF while simultaneously changing the filenames to the bookmark text once consolidated into a combined PDF file. This is all done via a SAS® program containing some VBScript code for file moving and converting in conjunction with an Excel spreadsheet containing the RTF output filenames and the title of the outputs.

INTRODUCTION

If you have attended a data monitoring committee (DMC) presentation or similar data review meeting, you probably notice a lot of wasted time swapping between different RTF files of table, figure, and listing outputs trying to make sense of the data. As the number of open Word documents increases, so can the level of confusion and disorganization; especially if those RTF files are not named with reviewer-friendly file names. If separate RTF files are being used for the review process and the files have not been named in a convention that allows the reviewer to understand the content of the output (many times these are just labelled as the output number, and not a meaningful title), reviewers will be spending minutes, if not hours, clicking open each RTF file trying to search for the output they want to review. Excel versions of the data output files are a good alternative to review the data due to their ability to sort and filter, but even that option has its limits in the reviewing process. Viewing figures, for example, is a much better experience through other means, and summary table reviews in Excel can become problematic due to the difficulties in perceiving subsections of summaries in a grided, cellular format. Arguably the best way to view the entirety of a study's outputs is through a single PDF document. This allows the ability to navigate through the bookmarks from output to output. It makes the viewing of specific data outputs much more elegant in a formal meeting than pulling up several individual Word or Excel documents with various sorts and filters. In this paper, you will learn the process of converting an entire folder of RTF outputs into a single PDF file with each output having its own linked bookmark.

There are a few requirements for this process. The first requirement includes an Excel spreadsheet containing the following information in separate columns: the numbering of the output (often controlled by clinical study report (CSR) standards, but not limited in any way), the titles of the outputs (often available via a table of contents for outputs produced alongside the statistical analysis plan [SAP]), and the filenames of the RTF output files. The second requirement is a VBScript program developed and executed within the final SAS® program and is only viable on Windows. The third requirement is a PDF editing software which allows for consolidation of multiple PDF files while retaining file names as bookmarks. Adobe Acrobat Pro is used for the purposes of this paper.

PREPARING THE PREREQUISITES

The first step is preparing the file structure which will include a parent folder with two child folders.

- For this paper, the parent folder will be named ConvertRTFtoPDF and will be represented by the macro variable xPath in subsequent SAS® code.
- A child folder for your RTF files (referred to as "RTF" for the purposes of this paper). Place the RTF files you wish to convert to PDF inside this folder.

- Create a folder for your PDF files (referred to as “PDF” for the purposes of this paper)
- Create an Excel (.xlsx) document containing the filenames of the RTF outputs, the output numbers according to CSR standards, and the name of the bookmark you wish to apply to the output in the consolidated PDF.

Name	Date modified	Type	Size
PDF	3/23/2023 11:19 AM	File folder	
RTF	3/23/2023 3:03 PM	File folder	
ConversionChart.xlsx	3/23/2023 11:17 AM	Microsoft Excel W...	21 KB
rfttopdf.sas	3/23/2023 3:16 PM	SAS System Progr...	4 KB

Figure 1: Recommended prerequisite folder structure

CREATING THE VBSCRIPT

Via a SAS® program, a VBScript will be created to allow for swift file conversion of the RTF files housed in the RTFs folder to PDF files in the PDFs folder. We will need to start by initializing our parent folder path (C:\ConvertRTFtoPDF in this case, but is represented as a macro variable in the final code set) and the VBScript program location:

```
%let xpath = C:\ConvertRTFtoPDF;
%let vbscript=&xpath.\rtftopdf.vbs;
```

The `dlcreatedir` option can be utilized to create the PDF child folder if it does not already exist. For the purposes of this paper example, we have already created a PDF child folder to reference. This folder should be empty before running the full script code. We will also need to create a libname for easy file referencing for the PDF folder within the VBScript program code and create a filename reference for the VBScript itself:

```
options dlcreatedir;
libname pdf "&xpath.\PDF";
filename vbscript "&vbscript";
```

Now we can begin creating the VBScript program through a SAS® `put` statement:

```

data _null_;
  file vbscript;
  put
    / "bRecursive = False"
    / "' Determine script location for VBScript"
    / "Dim oFSO : Set oFSO = CreateObject("""Scripting.FileSystemObject"")"
    / "Dim sScriptDir : sScriptDir =
      oFSO.GetParentFolderName(WScript.ScriptFullName) "
    / "sFolder = sScriptDir & ""\RTF\\""
    / "Set oRTF = CreateObject("""Scripting.FileSystemObject"")"
    / "Set oWord = CreateObject("""Word.Application"")"
    / "oWord.Visible = False"
    / "bRecursive = False"
    / "Set oFolder = oRTF.GetFolder(sFolder) "
    / "ConvertFolder(oFolder) "
    / "oWord.Quit"
    / "Sub ConvertFolder(oFldr) "
    / "  For Each oFile In oFldr.Files"
    / "    If LCase(oRTF.GetExtensionName(oFile.Name)) = ""rtf"" Then"
    / "      Set oDoc = oWord.Documents.Open(oFile.path) "
    / "      Str = replace(left(oFile,len(oFile)-4),""\RTF\",""\PDF\\"" )
    / "      oWord.ActiveDocument.SaveAs Str & ""pdf"", 17"
    / "      oDoc.Close"
    / "    End If"
    / "Next"
    / "Wscript.Echo ""RTFs have been converted to PDFs!"""
    / "If bRecursive Then"
    / "  For Each oSubfolder In oFldr.Subfolders"
    / "    ConvertFolder oSubfolder"
    / "  Next"
    / "End If"
    / "End Sub"
  ;
run;
filename vbscript;

```

Our VBScript program called rftopdf.vbs should now be present in our parent folder location and executable. Our next step is to execute our script to convert the RTF files in the RTF child folder to PDF files in the PDF child folder. We can do this via the %sysexec SAS® macro function on our newly created VBScript:

```
%sysexec "&path.\rftopdf.vbs";
```

Our PDF child folder should now contain PDF versions of all the RTF outputs in the RTF child folder.

Name	Date modified	Type	Size
ahf1.1-alt 11FEB2022.rtf	3/23/2023 11:19 AM	Rich Text Format	122 KB
f14.4.1.1-bcva 11FEB2022.rtf	3/23/2023 11:19 AM	Rich Text Format	680 KB
l14.3.2.1-death 11FEB2022.rtf	3/23/2023 11:19 AM	Rich Text Format	59 KB
l14.3.2.2-sae 11FEB2022.rtf	3/23/2023 11:19 AM	Rich Text Format	182 KB
t14.1.1.1-disp 11FEB2022.rtf	3/23/2023 11:19 AM	Rich Text Format	123 KB
t14.1.1.2-disp-sc 11FEB2022.rtf	3/23/2023 11:19 AM	Rich Text Format	1,519 KB
t14.1.1.3-disp-sf 11FEB2022.rtf	3/23/2023 11:19 AM	Rich Text Format	61 KB

Figure 2: Our original RTF files we wish to convert to PDF.

Name	Date modified	Type	Size
ahf1.1-alt 11FEB2022.pdf	3/24/2023 11:58 AM	Adobe Acrobat D...	132 KB
f14.4.1.1-bcva 11FEB2022.pdf	3/24/2023 11:58 AM	Adobe Acrobat D...	52 KB
l14.3.2.1-death 11FEB2022.pdf		Adobe Acrobat D...	12 KB
l14.3.2.2-sae 11FEB2022.pdf		Adobe Acrobat D...	50 KB
t14.1.1.1-disp 11FEB2022.pdf		Adobe Acrobat D...	35 KB
t14.1.1.2-disp-sc 11FEB2022.pdf		Adobe Acrobat D...	515 KB
t14.1.1.3-disp-sf 11FEB2022.pdf		Adobe Acrobat D...	52 KB

Windows Script Host

RTFs have been converted to PDFs!

OK

Figure 3: Our newly created PDF files with a popup window notifying us the VBScript has completed its conversions

CONSOLIDATING THE PDFS WITH BOOKMARKS

The most complicated section of SAS® code will be utilized during the latter of this section. We will use SAS® to import our spreadsheet and our list of PDF files in our PDF child folder. Once these two are represented within SAS® datasets, we can use basic manipulation techniques to achieve a merge such that our final dataset has the PDF file names and the new bookmark labels we want to use to rename the PDFs. Renaming the PDFs will involve some nested code and a combination of the %sasexec function, call execute, and return codes. Once that is complete, we can utilize software to combine the renamed PDFs into a consolidated file with the bookmarks retaining the information from the names of the individual PDF files.

RENAME PDF OUTPUTS TO DESIRED BOOKMARK LABELS

This step of the process requires use of the spreadsheet mentioned in the list of prerequisites, and SAS® code for importing, data manipulation, return commands, a data merge, and a slick combination of return commands in a macro to iteratively run a call execute function across all PDF outputs.

Import the Excel Document

You may use any spreadsheet type to house your output numbers, output titles, and bookmark labels. The example in this paper utilizes Excel with the file named `conversionchart.xlsx` that is housed in our specified parent folder, `xpath`. You can import the file into SAS® with the following code:

```
proc import datafile = "&xpath.\conversionchart.xlsx"
  out = outputs
  dbms =xlsx replace;
  getnames = yes;
  sheet = "outputs";
run;
```

	A	B	C
1	Num	Title	Program
2	14.1.1.1	Subject Disposition	t-disp
3	14.1.1.2	Subject Disposition by Study Center	t-disp-sc
4	14.1.1.3	Subject Disposition – Screen Failures	t-disp-sf
5	14.4.1.1	Plot of BVCA ETRRS Letters Read over Time	f-bcva
6	14.3.2.1	Listing of Deaths	l-death
7	14.3.2.2	Listing of Serious Adverse Events	l-sae
8	1.1	Plot of Subjects with ALT ~{unicode 2265}3 x ULN	ahf-alt

Figure 4: Contents of `conversionchart.xlsx` for purposes of this paper

A merge will need to be made between the names of your PDF files and your `conversionchart.xlsx` document later. How you achieve the merge is up to you, but a 1:1 merge is required for the example used in this paper. The expanded use of this tool is not limited to a 1:1 merge. You can manipulate the variable you would like to merge by as needed. The way the files are set up in this example require the following manipulation to achieve a proper merge with the titles of the PDF files:

```
data outputs2 (drop=title num source rename=(title2=title num2=num));
  set outputs;
  length num2 title2 $200;
  num2 = cats(source,num);
  title2 = catx(' ',num,title);
  proc sort;
    by num;
run;
```

`Num` represents the output number often, but not limited to, the CSR-regulated number for the output. `Title` represents the label you wish to apply to the PDF bookmark for each particular output. It can include the output number, but for this example, I use the `CATX` function to concatenate the output number and the bookmark label to get the final bookmark labels I will be using in the consolidated PDF. `Source` represents the type of output in this case and has the following values: “t” for tables, “l” for listings, “f” for figures, “ahl” for ad-hoc listings, etc. This variable will be used as the merge variable with out list of PDF outputs.

Compile a List of the PDF Outputs

Next on the list is gathering the names of the PDF outputs we generated with the VBScript program into a SAS® data set so we can merge in the bookmark labels we wish to apply. We can achieve this with the following SAS® code:

```
filename mypdf "&xpath.\PDF";
data pdfs (keep=filename);
  length filename $200;
  did=dopen("mypdf");
  filecount=dnum(did);
  do i=1 to filecount;
    filename=dread(did,i);
    put filename=;
    output;
  end ;
  rc=dclose(did);
run;
```

I will also need to manipulate the pdfs data set to achieve a proper merge with outputs2:

```
data pdfs2;
  length num $200;
  set pdfs;
  num = scan(filename,1,'-');
  proc sort;
    by num;
run;
```

Num is now in the same structure we achieved in the outputs2 data set, which means we have a SAS® data set with our PDF filenames and a SAS® data set with our filenames and desired bookmarks. We are ready to merge these together.

Merging PDF Output Names and Bookmark Labels

We can merge the two SAS® data sets with the following SAS® code:

```
data outputs3;
  length num $200;
  merge pdfs2 (in=a) outputs2 (in=b);
  by num;
  if a and b;
run;
```

Reviewing or placing warnings in your code to check for improper merges is highly recommended in the above code in the event you need to manipulate your merge variable and a clean merge is not achieved. With this SAS® data set containing our bookmark labels for our PDF outputs, we can begin the process of renaming the PDF outputs with the bookmark labels we want them to have in the consolidated PDF.

Execute Renaming Across All PDF Outputs

We can now make use of call execute and a macro to execute a rename return code for all values of filenames to change their filenames to the bookmark labels via the following code:

```

options noxwait;
%macro file_rename(inname,outname);
data _null_;
    rc= RENAME("&xpath.\PDF\&inname", "&xpath.\PDF\&outname", 'file');
run;
%mend file_rename;
data _null_;
    set outputs3;
    call execute('%file_rename(inname='||strip(filename)||',
        outname='||strip(title)||');');
run;

```

We can finish the process by removing the VBScript program by utilizing the %sysexec macro function:

```
%sysexec del "&xpath\rtftopdf.vbs";
```

What results should be completely renamed PDF outputs in your PDF child folder ready to be consolidated into a single PDF.

Name	Date modified	Type	Size
1.1 Plot of Subjects with ALT GTE 3 x ULN.pdf	3/23/2023 11:19 AM	Adobe Acrobat D...	132 KB
14.1.1.1 Subject Disposition.pdf	3/23/2023 11:19 AM	Adobe Acrobat D...	35 KB
14.1.1.2 Subject Disposition by Study Center.pdf	3/23/2023 11:19 AM	Adobe Acrobat D...	515 KB
14.1.1.3 Subject Disposition – Screen Failures.pdf	3/23/2023 11:19 AM	Adobe Acrobat D...	52 KB
14.3.2.1 Listing of Deaths.pdf	3/23/2023 11:19 AM	Adobe Acrobat D...	12 KB
14.3.2.2 Listing of Serious Adverse Events.pdf	3/23/2023 11:19 AM	Adobe Acrobat D...	50 KB
14.4.1.1 Plot of BVCA ETRRS Letters Read over Time.pdf	3/23/2023 11:19 AM	Adobe Acrobat D...	52 KB

Figure 5: Our newly labeled PDF with bookmark labels we want in the consolidated PDF.

CONSOLIDATE THE PDF OUTPUTS

Now it is time to open Adobe Acrobat Pro or any other PDF editing software which allows for combining PDF files. For purposes of this paper, Adobe Acrobat Pro is used. We will now consolidate the newly renamed PDF outputs in the PDF folder. Click “File > Create > Combine Files into a Single PDF...” to either drag and drop or make file references to all the PDF outputs you wish to consolidate into a single PDF. Click “Combine” in the top-right corner. It may take a while depending on the size and number of files you selected, but the end result should be a single combined PDF with bookmarks automatically generated based on the names of the individual PDF files. Once combined, ensure all bookmarks are populated correctly and edit any that may need special characters that are prohibited from Windows filename references.

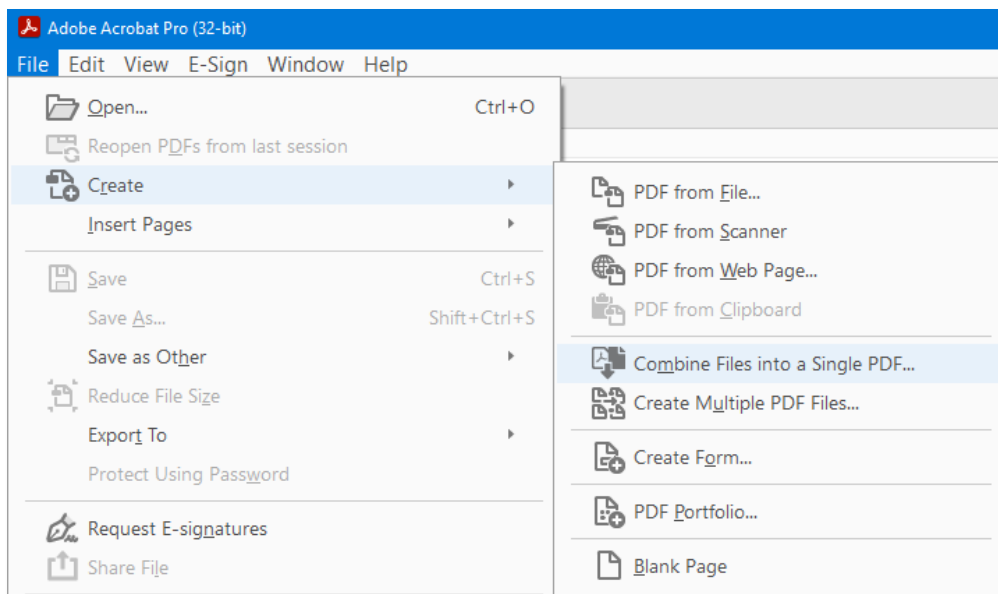


Figure 6: How to combine multiple PDF documents in Adobe Acrobat Pro

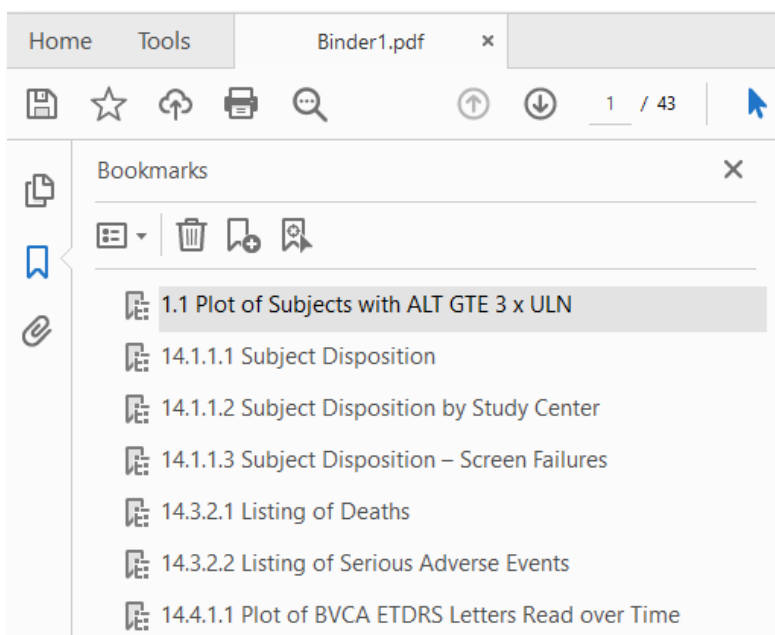


Figure 7: Our final consolidated PDF document with automatically generated bookmarks

CONCLUSION

The goal of this paper is to provide a modifiable process for transforming a folder of RTF outputs into a single consolidated PDF with bookmarks auto-generated from an external spreadsheet. The code provided in this paper is simplified for demonstrative purposes and can be modified in a host of ways. The VBScript code can be modified to perform different tasks across any folder of your choosing, the input spreadsheet can be in any format you wish, the renaming execution code can be utilized in a multitude of ways. Many custom warnings or reports can be created for checking validity of results, accuracy of file names, or problematic bookmark labels.

There are a few things to keep in mind regarding this process:

- Depending on the spreadsheet documents and naming conventions for RTF outputs you use, the way to get titles and output numbers will vary. In that case, the manipulation code in the appendix for the pdf2 and outputs2 data sets should be edited to achieve a proper merge with a title variable matching what is needed for the PDF bookmarks.
- Some symbols are not allowed in Windows file naming conventions. These symbols will need to be manually added into the bookmarks in post. I recommend adding warnings for any conflicting symbols that may appear in your bookmark labels such as “/”, “:”, “;”, “,”, “%”, any special Unicode symbols, and “#”. To see a complete guide of symbols not allowed in Windows file naming conventions, see the link to Microsoft’s documentation.
- Windows also has a limit of 255 characters for an entire filename path. Please keep this in mind as any truncation could result in incorrect application of bookmark labels or break the renaming process. Use of warnings via SAS® put statements is highly recommended to avoid file path length issues.
- This paper relies on Adobe Acrobat Pro for PDF consolidation. There are other options for consolidating PDFs, but it is not clear to me if other methods will function the same in bookmark retention when combining individual PDF outputs.
- Reports can be generated to show which titles contain symbols that cannot be represented in a Windows filename to expedite the manual bookmark-editing process. One example of this is presented in the appendix code to check for symbols not allowed in a Windows file name that needs to be manually updated in the final combined PDF bookmarks.
- Many warnings and force breaks should be placed throughout the code, however, the original RTF output files are never directly manipulated, so there is no danger of corrupting or modifying the original outputs.
- The VBScript does require the saving of the RTF outputs as PDF outputs which updates the “date modified” property of the files. It is recommended to use copies of your RTF outputs for this process rather than the original outputs.

REFERENCES

Website Microsoft, 2023. “Naming Files, Paths, and Namespaces.” Accessed March 23, 2023. <https://learn.microsoft.com/en-us/windows/win32/fileio/naming-a-file>.

Website Microsoft, 2023. “Maximum Path Length Limitation.” Accessed March 23, 2023. <https://learn.microsoft.com/en-us/windows/win32/fileio/maximum-file-path-limitation?tabs=registry>.

Website Microsoft, 2023. “Visual Basic documentation.” Accessed March 23, 2023. <https://learn.microsoft.com/en-us/dotnet/visual-basic/>.

CONTACT INFORMATION

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

Tyler Plevney
 Emanate Biostats, Inc.
 (949) 878-6354
tyler.plevney@emanatebiostats.com

TRADEMARK

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.

APPENDIX

This macro creates a folder with PDF versions of a folder of RTF outputs and renames them according to an Excel spreadsheet stored in the specified filepath:

```
%macro rtftopdf(path);
%let xpath = %str(&path);
%let vbscript=&xpath.\rtftopdf.vbs;
options dlcreatedir;
libname pdf "&xpath.\PDF";
filename vbscript "&vbscript";
data _null_;
  file vbscript;
  put
    / "bRecursive = False"
    / "' Determine script location for VBScript"
    / "Dim oFSO : Set oFSO = CreateObject(""Scripting.FileSystemObject"")"
    / "Dim sScriptDir : sScriptDir =
      FSO.GetParentFolderName(WScript.ScriptFullName) "
    / "sFolder = sScriptDir & ""\RTF\""
    / "Set oRTF = CreateObject(""Scripting.FileSystemObject"")"
    / "Set oWord = CreateObject(""Word.Application"")"
    / "oWord.Visible = False"
    / "bRecursive = False"
    / "Set oFolder = oRTF.GetFolder(sFolder) "
    / "ConvertFolder(oFolder) "
    / "oWord.Quit"
    / "Sub ConvertFolder(oFldr) "
    / "  For Each oFile In oFldr.Files"
    / "    If LCase(oRTF.GetExtensionName(oFile.Name)) = ""rtf"" Then"
    / "      Set oDoc = oWord.Documents.Open(oFile.path) "
    / "      Str = replace(left(oFile,len(oFile)-4),""\RTF\","", "\PDF\") "
    / "      oWord.ActiveDocument.SaveAs Str & ".pdf", 17"
    / "      oDoc.Close"
    / "    End If"
    / "Next"
    / "Wscript.Echo ""RTFs have been converted to PDFs!"" "
    / "If bRecursive Then"
    / "  For Each oSubfolder In oFldr.Subfolders"
    / "    ConvertFolder oSubfolder"
    / "  Next"
    / "End If"
    / "End Sub"
;
run;
filename vbscript;
options nosymbolgen nodlcreatedir;
%sysexec "&path.\rtftopdf.vbs";

proc import datafile = "&xpath.\conversionchart.xlsx"
  out = outputs
  dbms = XLSX replace;
  getnames = yes;
  sheet = "Outputs";
run;
data outputs (keep=num title source);
  set outputs;
```

```

        where ~missing(num);
        source = strip(scan(program,1,'-'));
run;
data outputs2 (drop=title num source rename=(title2=title num2=num));
    set outputs;
    length num2 title2 $200;
    num2 = cats(source,num);
    title2 = catx(' ',num,title);
proc sort;
    by num;
run;
filename mypdf "&xpath.\PDF";
data pdfs (keep=filename);
    length filename $200;
    did=dopen("mypdf");
    filecount=dnum(did);
    do i=1 to filecount;
        filename=dread(did,i);
        put filename=;
        output;
    end;
    rc=dclose(did);
run;
data pdfs2;
    length num $200;
    set pdfs;
    num = scan(filename,1,'-');
proc sort;
    by num;
run;
data outputs3;
    length num $200;
    merge pdfs2 (in=a) outputs2 (in=b);
    by num;
    if a and b;
run;
options noxwait;
%macro file_rename(inname,outname);
data _null_;
    rc= RENAME("&xpath.\PDF\&inname", "&xpath.\PDF\&outname", 'file');
run;
%mend file_rename;
data _null_;
    set outputs3;
    call execute('%file_rename(inname='||strip(filename)||',
        outname='||strip(title)||');');
run;
%sysexec del "&xpath\rtftopdf.vbs";
%mend rtftopdf;
%rtftopdf(%str(C:\ConvertRTFtoPDF));

```