PharmaSUG China 2024 - Paper CC-10048

Build PDF TFL Package with Table of Contents and Bookmarks – Taking Advantage of Unix Command Line

Shimon Wang, Abby Cai, Everest Clinical Research

ABSTRACT

Packaging table, listing, and figure outputs (TLF) as a single or several combined files is a common request from sponsor in clinical trial industry. Rich Text Format (RTF) is the most popular format used for production of individual reports because SAS provides powerful and flexible control to RTF. Naturally, RTF becomes the option for combining outputs in most cases and plenty of standard programs and macros have been developed for this purpose.

However, limitations are that RTF file may display differently on different computing systems even different version of Microsoft Word application. Combined output in RTF sometimes has thousands of pages, which means the file size is large thus making it time-consuming to open, browse, and transfer. Moreover, if PAGEOF ODS ESCAPECHAR function of SAS is used to generate the page number, it will accumulate through the whole file which is not expected.

Portable Document Format (PDF) doesn't have the limitations above which makes it a better option to convert RTF to PDF and then combine individual PDF files together. This paper introduces a four-step process from individual RTF outputs to combined PDF files with table of contents (TOC) as well as bookmarks. Taking advantage of Unix script and VBA macro, this process can improve efficiency by running in the background. Furthermore, it enables users to modify TOC section in RTF to get a more delicate and well-formatted combined output.

INTRODUCTION

RTF is widely used in the clinical trial industry because SAS provides powerful and flexible control to it. On the one hand, it is easy to create output with desired format using ODS RTF. On the other hand, RTF is just formatted plain text that can be read and modified by SAS which makes post-processing very easy. So, when we need to package the final outputs, it is natural to choose RTF as the format.

However, there are several limitations we met with RTF:

1. RTF files may display differently on different platforms or with Microsoft word application using different authoring language.
2. Combined RTF files are often too large thus takes lots of time to open, browse, and transfer to sponsor.
3. PAGEOF ODS ESCAPECHAR function is usually used to generate the page number in individual outputs. However, the page number accumulates through the whole combined file which is not expected.

An alternative solution is to convert RTF to PDF and then combine individual PDF files together, which takes the advantage of powerful control to RTF of SAS as well as the stable feature of PDF. Providing that it is not easy to combine PDF files using SAS, we decided to utilize the function of several Unix command line tools.

Here we would like to introduce a four-step process to combine individual outputs from RTF to the final PDF package with TOC and bookmarks. The functional module of this process is a bash script (named Packpdf) which is also the main topic we discuss in this paper. Packpdf doesn't require extra configuration files, it takes PDF files from any source as the input as long as they contain required information and creates a combined output and a corresponding 'dummy' TOC RTF file.

Below we introduce the process in detail step by step. We also present key bash script codes and RTF codes related to each procedure.

METHODOLOGY AND DISCUSSION

REQUIREMENTS

The process has the following requirements:

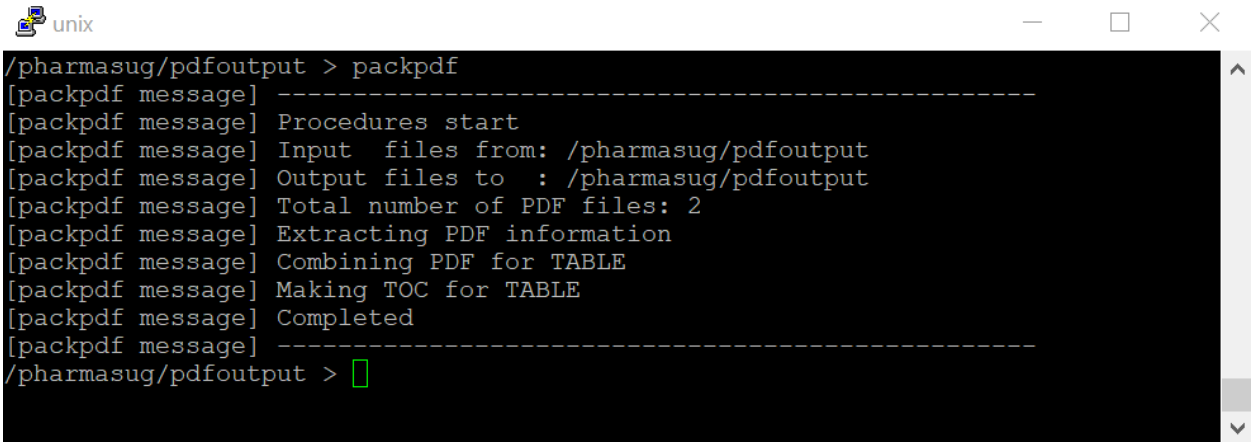
1. Packpdf is a bash script thus can only be used on Unix/Linux system.
2. Unix command line tools Pdftotext, Qpdf, and GhostScript are used within Packpdf to process PDF files.
3. For individual output, the title must exist on the first page and must have a blank line under it.

STEP 1: CREATE INDIVIDUAL PDF FILE

There are plenty of ways to convert RTF to PDF, such as VBA macros (See in Appendix), VBScript, or SAS macros. Packpdf doesn't care about the source of PDF files, thus creating PDF files directly by SAS is also acceptable.

STEP 2: COMBINE PDF FILES AND CREATE DUMMY TOC

Packpdf mainly relies on powerful function of GhostScript, which is an interpreter for the PostScript language and PDF files[1], to perform adding bookmarks and combining PDF files actions. The detailed procedures of using Packpdf are as follows: put the PDF files to be combined under a folder, change directory to the folder, type 'packpdf' and press enter, then Packpdf starts the combining process. In the terminal interface (Display 1), Packpdf presents some information about the process, including the paths of input files and output files for checking purpose, number of files to be combined, and the processing status.



Display 1. Interface for Using Packpdf

How does Packpdf process PDF files

First, Packpdf detects PDF files that have names beginning with the letter 't' or 'l' or 'f' or 'g', to limit the scope of input files.

For each PDF file, Packpdf does the following tasks:

1. Extract the text on the first page using Pdftotext, and then extract the title text. The logic is: a typical output title has the pattern 'Table/Listing/Figure/Graph xx.xx.xx.xx', so based on this pattern we can locate the start line of the title using grep and awk commands, and then use sed command to get the text starting from the start line and ending at the blank line. In bash script codes below, '$1' refers to name of the input PDF file, and 'firstpagetext' refers to a temporary file to store text extracted from the first page.

```
pdftotext -enc UTF-8 -q -layout -f 1 -l 1 $1 firstpagetext

START=$(cat firstpagetext | grep -e -n -i
"(TABLE|LISTING|FIGURE|GRAPH) *([0-9]+(-|\.|\w)?)+" | awk -F:
'NR==1{print $1}')

TITLE=$(cat firstpagetext | sed -n "${START},/^$/p" | sed -e 's/^ *//'
-e 's/ *$//' | tr '\n' ' ')
```

2. Get the number of pages using '--show-npages' option of qpdf command.

```
PAGE=$(qpdf --no-warn --show-npages $1 2>/dev/null)
```

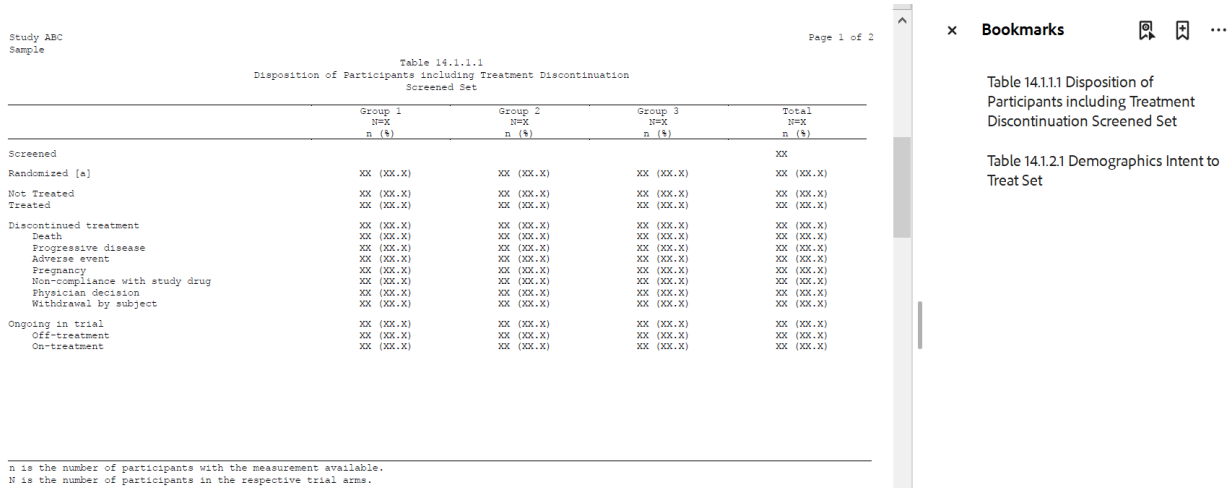
3. Insert the title text as bookmark to the first page of each PDF file using GhostScript. In bash script codes below, '$1' refers to the input PDF file, 'bkmk_$1' refers to the output PDF file in which a bookmark is added on the first page, '${TITLE}' refers to the title text extracted previously (to know more about usage of GhostScript, please refer to reference [1]).

```
gs -q -sDEVICE=pdfwrite -o bkmk_$1 -c "[/Page 1 /Title (${TITLE}) /OUT
pdfmark" -f $1
```

While processing the input PDF files one by one, Packpdf stores the extracted information in a temporary file. After all individual PDF files are processed, Packpdf concatenates all bookmarked individual PDF files using GhostScript. In the bash script codes below, '${OUTDIR}' refers to the output directory, '${PDFLIST}' refers to the list of input PDF files.

```
gs -q -sDEVICE=pdfwrite -o ${OUTDIR}/COMBINED.pdf -f ${PDFLIST}
```

The combined PDF file (Display 2) has bookmarks which can be used in internal review as well as delivery when TOC is not requested.



Study ABC
Sample

Page 1 of 2

Table 14.1.1.1
Disposition of Participants including Treatment Discontinuation
Screened Set

	Group 1 N=X n (%)	Group 2 N=X n (%)	Group 3 N=X n (%)	Total N=X n (%)
Screened				XX
Randomized [a]	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Not Treated	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Treated	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Discontinued treatment	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Death	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Progressive disease	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Adverse event	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Pregnancy	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Non-compliance with study drug	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Physician decision	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Withdrawal by subject	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Ongoing in trial	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Off-treatment	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)
On-treatment	XX (XX.X)	XX (XX.X)	XX (XX.X)	XX (XX.X)

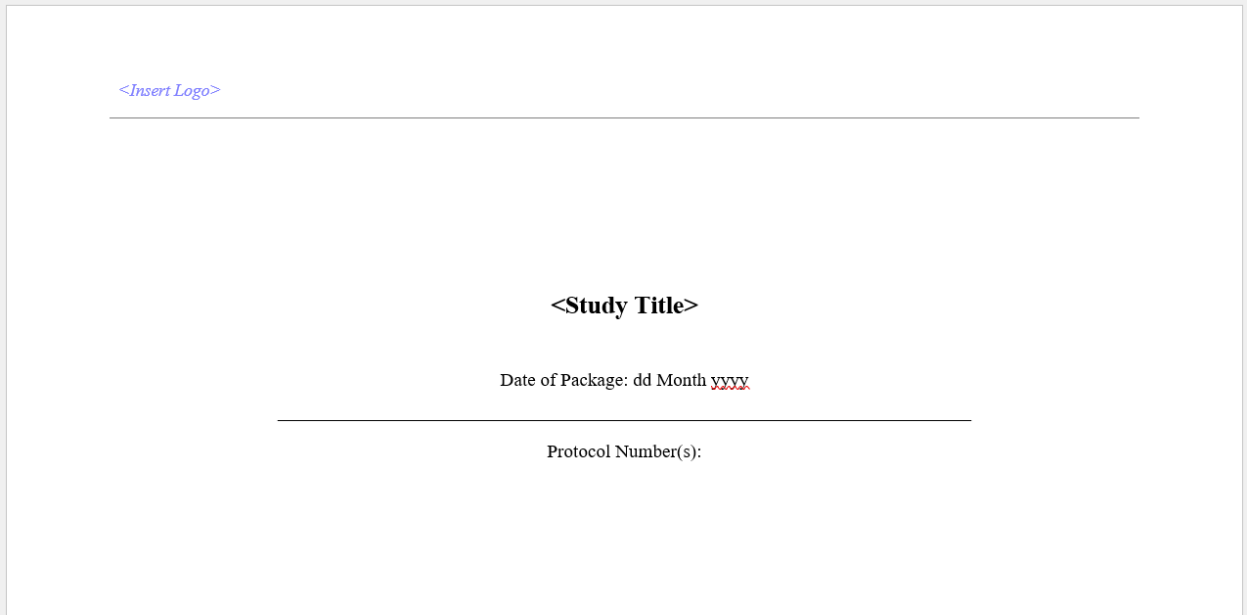
n is the number of participants with the measurement available.
N is the number of participants in the respective trial arms.

Display 2. Combined Output with Contents and Bookmarks

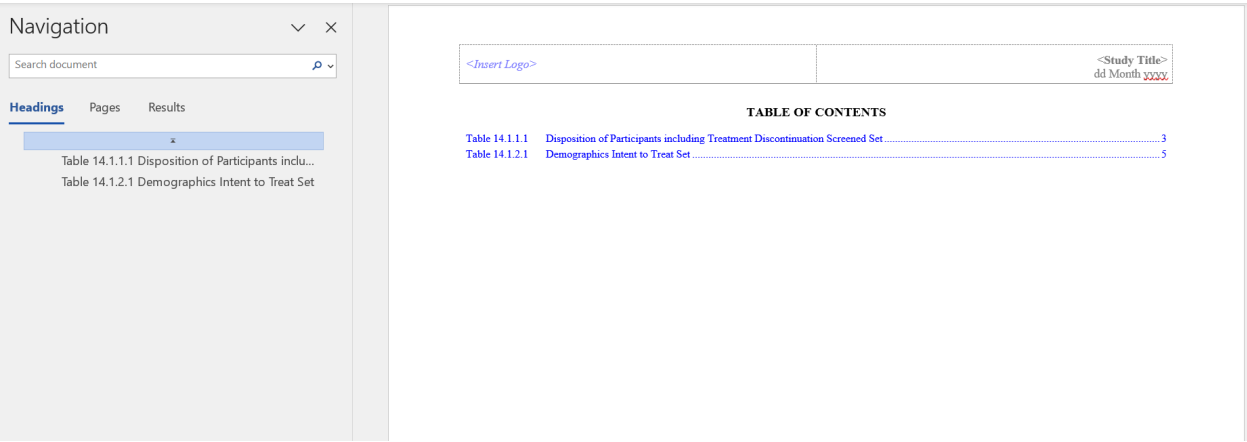
How does Packpdf create dummy TOC RTF files

RTF files are usually 7-bit ASCII plain text, consisting of control words, control symbols, and groups[2]. Since we already have the information of title and number of pages of each individual PDF file, we can construct a TOC using RTF control words. The key point is to create a section of titles with hyperlinks and dummy blank pages of each output to make sure each dummy output has the same number of pages as the original output.

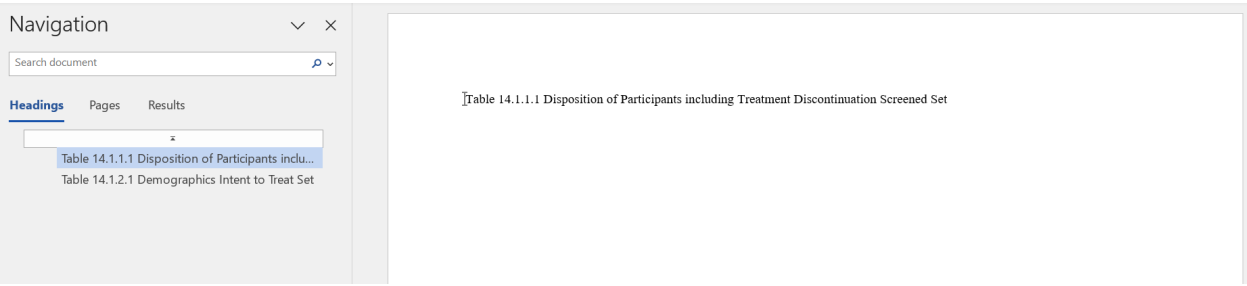
The constructed dummy TOC RTF file contains 3 components: a cover page (Display 3), a table of contents section with hyperlinks (Display 4), and dummy blank pages for each individual output (Display 5).



Display 3. The Cover Page of 'dummy' TOC RTF File



Display 4. The Table of Contents Section of the 'dummy' TOC RTF File



Display 5. The Dummy Page for Table 14.1.1.1 of the 'dummy' TOC RTF File

Below are the key RTF codes for constructing TOC section with hyperlinks to bookmarks. '{\field{*\fldinst{hyperlink \\l "bkmk1"}}{\fldrslt{…}}}' invokes HYPERLINK field[3]. '\tqr\tldot' creates the flushing right dot tab.

```
{\pard\qc\fi0\li0\ri0\sb0\sa240\cf0\f0\b\fs24 TABLE OF CONTENTS\par}

{\pard\ql\sb0\sa60\cf2\f0\fs20{\field{\*\fldinst{hyperlink \\l
"bkmk1"}}{\fldrslt{\fi-1480\li1480\ri720\tx1480\tqr\tldot\tx12960 Table
14.1.1.1\tab Disposition of Participants including Treatment Discontinuation
Screened Set\tab 3\par\pard}}}}

{\pard\ql\sb0\sa60\cf2\f0\fs20{\field{\*\fldinst{hyperlink \\l
"bkmk2"}}{\fldrslt{\fi-1480\li1480\ri720\tx1480\tqr\tldot\tx12960 Table
14.1.2.1\tab Demographics Intent to Treat Set\tab 5\par\pard}}}}
```

Below are the key RTF codes for creating bookmarks and outlines. '{*\bkmkstart …}{*\bkmkend …}' constructs a bookmark on the dummy blank page. '\outlinelevel2' invokes second level outline. '{\page}' adds blank pages to simulate the original output.

```
\sect\sectd{\header{}}{\footer{}}{\*\bkmkstart bkmk1}{\*\bkmkend bkmk1}

{\pard\outlinelevel2{Table 14.1.1.1 Disposition of Participants including
Treatment Discontinuation Screened Set}\par}{\page}

\sect\sectd{\header{}}{\footer{}}{\*\bkmkstart bkmk2}{\*\bkmkend bkmk2}

{\pard\outlinelevel2{Table 14.1.2.1 Demographics Intent to Treat
Set}\par}{\page}
```

STEP 3: MODIFY DUMMY TOC RTF

On the previous dummy TOC RTF file, we can insert logo to the header, fill in the study information, add an additional sub header to group of tables with the same topic, etc., to make it looks better. When the modification is done, we convert it to PDF then we are ready to go to the final step.

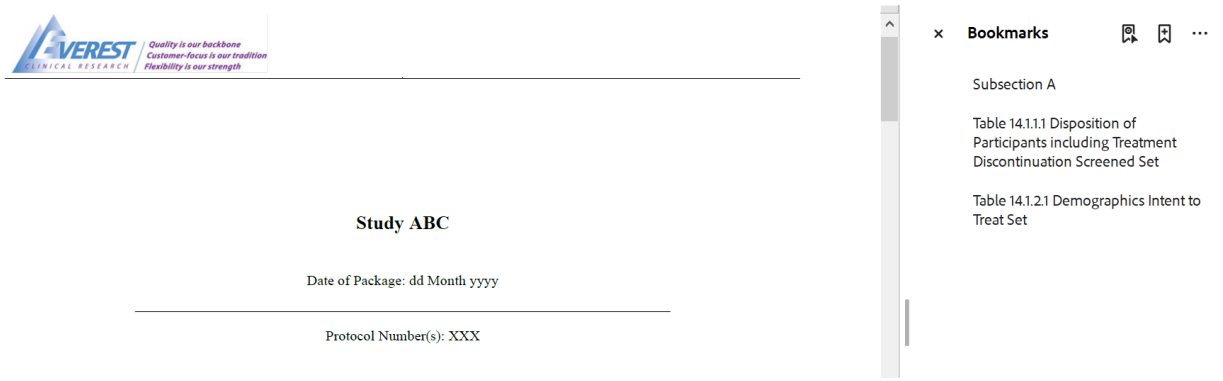
STEP 4: REPLACE BLANK PAGES WITH CONTENTS

Now we have 2 PDF files, one has TOC and dummy blank pages for each individual output, and one has contents of each individual output but no TOC section. The final step is to replace the blank pages in the dummy TOC PDF file with contents using Acrobat. Finally, we get a combined PDF file with well-formatted clickable TOC section and bookmarks.

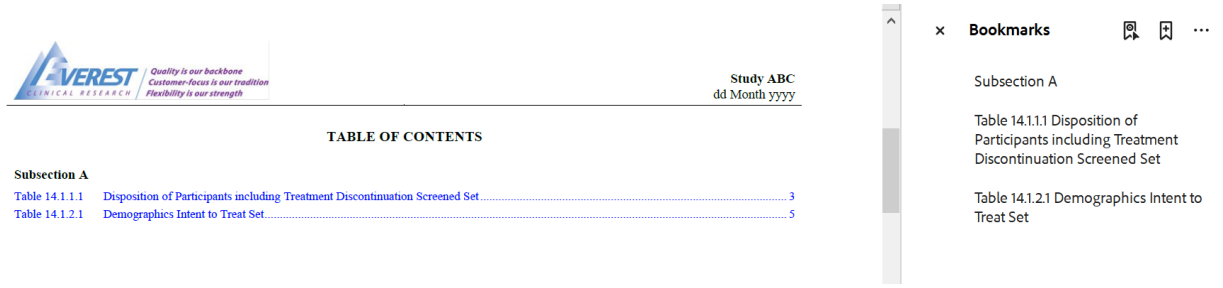
The first page of the final output is a cover page (Display 6) contains information of the study, date of package, and logo.

Table of contents starts from the second page (Display 7) and has a section of hyperlinked titles with page numbers.

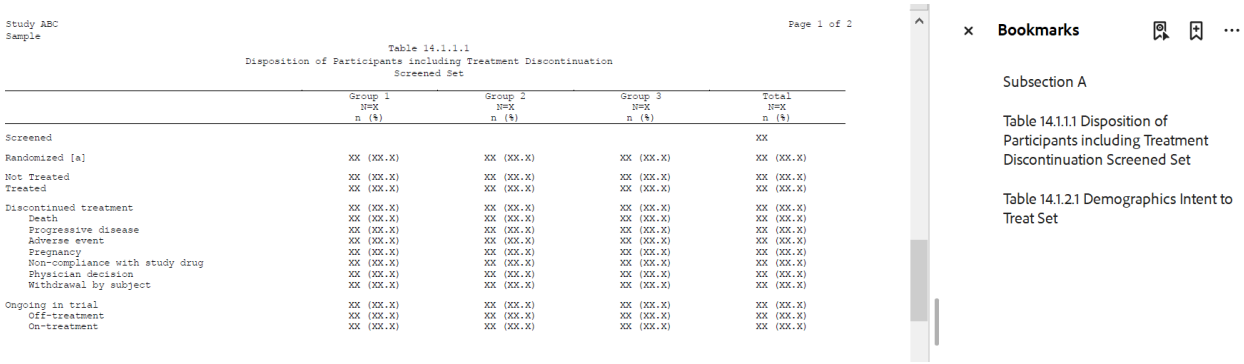
Each of the individual output (Display 8) has a corresponding bookmark and the additional sub header has a clickable bookmark that can be used to browse the file as well.



Display 6. Cover Page of the Final Combined Output



Display 7. TOC Section of the Final Combined Output



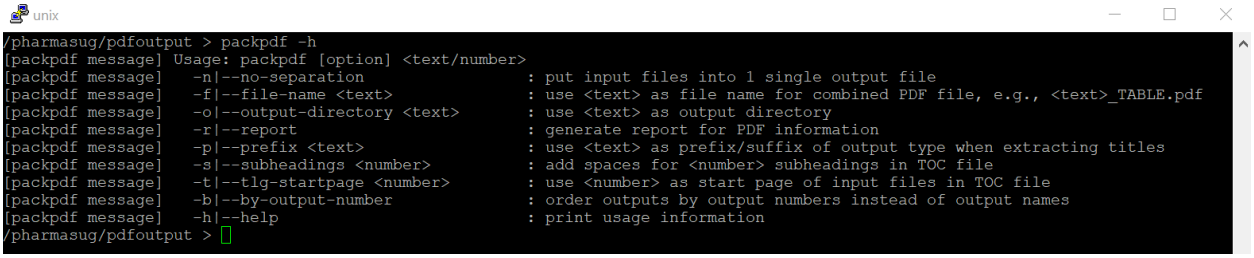
Display 8. Table 14.1.1.1 of the Final Combined Output

MORE FUNCTIONS OF PACKPDF

Packpdf has some options (Display 9) which give user more flexibility.

1. The option '-n' tells Packpdf to combine tables, listings, and figures into 1 single file. By default, Packpdf combines the different kinds of outputs into separate files.
2. The option '-f' tells Packpdf to name the output file with the text you specified.
3. The option '-o' tells Packpdf to output the combined file to the folder you specified. By default, Packpdf outputs files in the current folder.
4. The option '-r' tells Packpdf to put the information extracted from each individual PDF file into a CSV file, which can be used for review.

5. The option '-s' tells Packpdf how many subheadings to be added in the TOC section so that Packpdf can leave extra space in dummy TOC RTF file.
6. The option '-t' tells Packpdf the exact page number from where the individual outputs start, can be used if certain pages of dummy TOC section are needed.
7. The option '-b' tells Packpdf to sort the outputs by the output title numbers. By default, Packpdf sort PDF files by their names.
8. The option '-h' tells Packpdf to print usage information on the screen.



Display 9. Options of Packpdf

CONCLUSION

In this paper we introduce a four-step process based on bash script Packpdf to perform combination from individual RTF outputs to final PDF package with well-formatted TOC section and bookmarks. The process takes both advantage of powerful and flexible control to RTF of SAS and the stable feature of PDF. Most procedures in the process are automated thus can be run in the background to improve efficiency. Moreover, Packpdf has useful options to provide users flexibility in different scenarios.

REFERENCES

[1] "Ghostscript Overview." Available at https://www.ghostscript.com/.

[2] "Rich Text Format (RTF) Specification, Version 1.9.1". Available at https://go.microsoft.com/fwlink/?LinkId=120924.

[3] "List of field codes in Word." Available at https://support.microsoft.com/en-us/office/list-of-field-codes-in-word-1ad6d91a-55a7-4a8d-b535-cf7888659a51.

CONTACT INFORMATION

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

Shimon Wang
Everest Clinical Research
shimon.wang@ecrscorp.com

Abby Cai
Everest Clinical Research
abby.cai@ecrscorp.com

APPENDIX

RTF2PDF

```
Sub RTF2PDF()
  Dim objWord As Object
  Set objWord = CreateObject("Word.Application")
  objWord.Visible = False

  Dim FilePathList As Variant
  Set FilePathList = GetFilePath()
  If FilePathList Is Nothing Then Exit Sub

  Dim objFileSystem As Object
  Set objFileSystem = CreateObject("Scripting.FileSystemObject")

  Dim MsgRes As Integer
  Dim OpenPDF As Boolean
  MsgRes = MsgBox("Open PDF when completed?", vbYesNo, "RTF2PDF")
  If MsgRes = 6 Then
    OpenPDF = True
  Else
    OpenPDF = False
  End If

  Dim objFile As Object
  For Each FilePath In FilePathList
    Set objFile = objFileSystem.GetFile(FilePath)

    With objFile
        Dim FileDrive As String
        Dim FileType As String
        Dim FileName As String

        FileDrive = .Drive
        FileType = .Type
        FileName = .Name
```

```
        If (FileType Like "*Word*" Or FileType = "Rich Text Format") And Not
(FileName Like "*~$*") Then
          Set objWordDoc = objWord.Documents.Open(FilePath)
          With objWordDoc
            .Fields.Update

            Dim PdfPath As String
            Dim PdfName As String
            PdfPath = .Path
            PdfName = .Name
            PdfName = GetFileName(PdfName)
            PdfName = PdfPath & "\" & PdfName & ".pdf"

            .ExportAsFixedFormat OutputFileName:=PdfName,
ExportFormat:=wdExportFormatPDF, OpenAfterExport:=OpenPDF, _
            OptimizeFor:=wdExportOptimizeForPrint, Range:=wdExportFromTo,
From:=1, To:=.Range.Information(wdNumberOfPagesInDocument), _
            Item:=wdExportDocumentContent, IncludeDocProps:=True,
KeepIRM:=True, CreateBookmarks:=wdExportCreateHeadingBookmarks, _
            DocStructureTags:=True, BitmapMissingFonts:=True

            .Close savechanges:=wdDoNotSaveChanges
          End With

        End If
      End With

  Next FilePath

  objWord.Quit
  Set objWord = Nothing
  Set objWordDoc = Nothing
  Set objFile = Nothing
End Sub

Function GetFilePath() As Variant

  With Word.Application.FileDialog(msoFileDialogFilePicker)
    .AllowMultiSelect = True
```

```
    .Title = "Please select one or more files"
    .ButtonName = "Select"
    .Filters.Clear
    .Filters.Add "Word Files", "*.rtf;*.doc;*.docx", 1
    If .Show = -1 Then
      Set GetFilePath = .SelectedItems
    Else
      Set GetFilePath = Nothing
    End If
  End With

End Function

Function GetFileName(ByVal StrName As String)
  Dim StrTemp As String
  StrTemp = StrName

  Position = Len(StrTemp) - VBA.InStr(1, VBA.StrReverse(StrTemp), ".")
  If Position <> 0 Then
    StrTemp = Mid(StrTemp, 1, Position)
  End If

  Position = VBA.InStr(1, StrTemp, "\")
  If Position <> 0 Then
    Position = VBA.InStr(1, VBA.StrReverse(StrTemp), "\")
    StrTemp = Mid(VBA.StrReverse(StrTemp), 1, Position - 1)
    StrTemp = VBA.StrReverse(StrTemp)
  End If

  GetFileName = StrTemp
End Function
```