

PYTHON GUI executable tool to generate TOC using RTF files in various formats (RTF, DOCX, PDF with bookmark)

Sathish Kumar N, Labcorp, Chennai, India

ABSTRACT

In Clinical Industry, statistical reports usually contain huge amounts of tables, listings, and figures. The quantity of the TLFs may vary based on the complexity of the study. Generally, the sponsor asks for the statistical outputs in a single file with a table of contents (TOC) format. To achieve this, we may use various approaches (like a manual, SAS, VBA, etc...) to convert the individual RTF outputs into a combined TOC file. But, there are some limitations and time-consuming factors to the mentioned approach. This paper demonstrates a new, very efficient, and more flexible PYTHON GUI-based tool (single executable application) approach to concatenate the SAS-generated ODS RTF and create a nicely formatted hyperlinked table of contents with various formats (RTF, DOCX & PDF with/without a bookmark). This paper additionally introduces a few key points, and limitations of the PYTHON TOC tool, and also offers some relevant sample codes.

INTRODUCTION

There are numerous ways to create the TOC, but no papers exist that explain how to retrieve the titles from RTF files (if the generated RTFs do not use the RTF's bookmark/hyperlink features). This tool is an independent application that works in both Windows and Unix environments and it uses the keywords "Table," "Listing," "Figure," and "Appendix" (first occurrence) on the header section of the SAS-generated RTF files to extract titles in order to resolve the problem. In accordance with the user's input for the number of lines, the program extracts the text from keywords to the corresponding lines and compresses all RTF tags to obtain the actual title from each output. After all titles have been read from each output, which has been individually chosen, the program combines all the titles and creates the hyperlinked TOC index page at the top of the TOC file. Thus, we don't have to adhere to specific guidelines in order to obtain the TOC index titles.

Also, this tool provides various options to get the generated combined file in RTF, DOCX, PDF without a bookmark, and PDF with a bookmark, so users can get the TOC file for their desired need.

PYTHON TOC TOOL

1. In Figure 1, Output Format is a drop-down box, which contains the final TOC output format (RTF, DOCX, PDF).
2. PDF Bookmark is a check box which uses to create a PDF bookmark in the PDF TOC output, if the checkbox is enabled, the PDF output will get bookmarks which is shown in Figure 3 in the right side area.
3. A drop-down box labelled "No. of Title Lines" with values ranging from 2 to 6. It pulls the corresponding number of title lines from the RTF outputs based on this input.
4. RTF Files Path is the folder selection box, using the RTF Path button, the user can pick the RTF outputs folder path here. Once the path is selected, it reads all the RTF files from the folder and lists them under the white area (selected RTF files). Using the check box, the user may have the provision to unselect the unwanted RTF files in the TOC execution.
5. TOC file name is the final generated TOC file name. The same file name will be used for all types of output format.
6. After selecting the RTF file path, all of the rtf files in the folder are read, and all of the RTF files are listed in chronological order in the white area (chosen RTF files in Figure 1) with the checkbox marked. The user can deselect any file in the list if they do not want it to be included to the TOC file.

Figure 1: RTF TOC Generator - PYTHON tool

Table of Contents Generator

Output Format: RTF **1**

PDF Bookmark: ☐ **2**

No. of Title Lines: 3 **3**

RTF Files Path: C:/Sathish/Projects/X/ RTF Path **4**

TOC File Name: TOC_2023-01-23 (.rtf/.docx/.pdf) **5**

Generate TOC Clear Details

Select	S.No	RTF Files
☒	1	T01_01_demog.rtf
☒	2	T01_02_dispo.rtf
☒	3	T01_03_mh.rtf
☒	4	T01_04_01_react_a.rtf
☒	5	T01_04_02_react_b.rtf
☒	6	T01_04_03_react_c.rtf
☒	7	T01_04_04_01_fever.rtf
☒	8	T01_04_04_02_report1442.rtf
☒	9	T01_04_04_03_report1443.rtf

6

TOC PROCESS

STEP 1: Creating a TOC index Page

In the TOC process, the tool reads the selected RTF outputs to the corresponding RTF file path and extracts title and page number information from the individual RTF outputs by reading them using the codecs module in PYTHON. The selected RTF files will loop one by one to get the individual RTF output titles and page number information. Once all the information is collected, it is combined through a loop, and the TOC page is created along with the hyperlinks to respective outputs.

Code snippet 1: To find the first occurrence of title from RTF outputs

```
if (textline.find("Table ") > -1 or textline.find("Listing ") > -1 or textline.find("Figure ") > -1 or
textline.find("Table\\~") > -1 or textline.find("Listing\\~") > -1 or textline.find("Figure\\~") > -1 or
textline.find("Appendix ") > -1 or textline.find("Appendix\\~") > -1) and titleindex == False:
    nline = 1
    titleindex = True
    title = ""
```

```

nline = nline + 1
if titleindex == True and nline > 0 and nline <= (self.titlelines+1):
    title = title + " " + textline

```

Code snippet 2: To get number of pages of individual RTF file

```

no_of_lines = 0
for textline in f.readlines():
    no_of_lines += 1
    if textline.find("\\sectd") > -1 and totla_no_lines != no_of_lines:
        totalpage = totalpage + 1
self.file_pages[filein] = totalpage

```

Code snippet 3: Sample code to create TOC Index page:

```

#Creating TOC index page based on first RTF file occurrence and Reading first RTF output to get opening block for Index TOC file
def indexTOC(self, fullpath):
    f = codecs.open(fullpath, 'r', encoding='utf-8')
    temp = tempfile.TemporaryFile(mode="w+")
    for textline in f.readlines():
        temp.write(textline)
        if textline.find("\\sectd") > -1:
            sectline = textline
            break
    page_accumulate = 1
    i = 0

    #Concatenating hyperlinked tiles on the TOC index page
    for x in range(1, (self.nooftocpage+1)):
        page_accumulate=page_accumulate+self.nooftocpage
        for file in self.filelist:
            ff = file.split("\\")[1]
            if self.file_cumpage[ff] == x:
                i = i + 1
                rtfline = "\\trowd\\trkeep\\trqc\\n\\
\\pard\\plain\\intb\\lsb0\\sa0\\ql\\f1\\fs16\\cf1\\ri72{%s\\cell}\\n\\
\\pard\\plain\\intb\\lsb0\\sa0\\ql\\f1\\fs16\\cf1{\\field {\\*\\fldinst HYPERLINK \\l\\l \"\"%s\"\" \\lo \"\"%s\"\"} {\\fldrslt
%s}}\\cell}\\n\\pard\\plain\\intb\\lsb0\\sa0\\ql\\f1\\fs16\\cf1{%s\\cell}\\n\\
{\\row}" % (i,f"{self.file_bookmark[ff].upper()}",f"{self.file_titles[ff].strip()}",self.file_btiles[ff].strip()),
                page_accumulate)

                page_accumulate += self.file_pages[ff]
                self.file_pages[ff] = page_accumulate
                temp.write(rtfline)

    temp.write("\\n\\pard\\sect")
    temp.seek(0, 0)
    f.close()
    return temp

```

STEP 2: Combining selected outputs with index TOC Page

RTF file consists of three blocks which are described below.

- **An opening block** is the RTF code until the first “\sectd” (reset to default section properties) text of the file, which includes information on how the file is rendered. e.g. font styles, layout, and colors.
- **The content block** includes the content to be displayed. The RTF code for each page starts with “\sect\sectd” (“\sect” - end of section and paragraph) or “\sectd”.
- **The closing block** is the RTF code after the last “\pard” (resets to default paragraph properties) of the file, typically only a closing curly brace (“}”).

The following approach is used to concatenate all the outputs with the index TOC page.

- Read first RTF output (sorted based on filename) till “\sectd” text to get the opening block for the index TOC page, then the hyperlinked title text of individual RTFs will be concatenated through the loop (see Code snippet 3).
- Remove opening and closing blocks of all RTF outputs and Insert “\sect” text in front of the first page of each RTF file, since the combined TOC, the first page of the individual file is no longer the first page, except the first file, so the “\sect” text needs to be included in the front of all files.
- Remove the opening block of the last RTF file.

Using the above steps, all RTF files will be concatenated to the index TOC page and the final combined RTF file saved into “_toc_” folder in the RTF file path that was chosen.

STEP 3: Converting TOC RTF file into DOCX/PDF format (based on user input)

Once the RTF TOC is generated, based on user input the DOCX file will be created using win32com.client PYTHON package (see Code snippet 4).

Code snippet 4: Creating DOCX/PDF format TOC file

```
#Converting RTF file into DOCX format using win32
def change_word_format(self,file_path, exten, constant):
    word = Dispatch('Word.Application')
    word.Visible = 1
    file_path = os.path.abspath(file_path)
    doc = word.Documents.Open(file_path)
    doc.Activate()

    # Rename path with .docx
    new_file_abs = re.sub(r'\\.lw+$', exten, file_path)

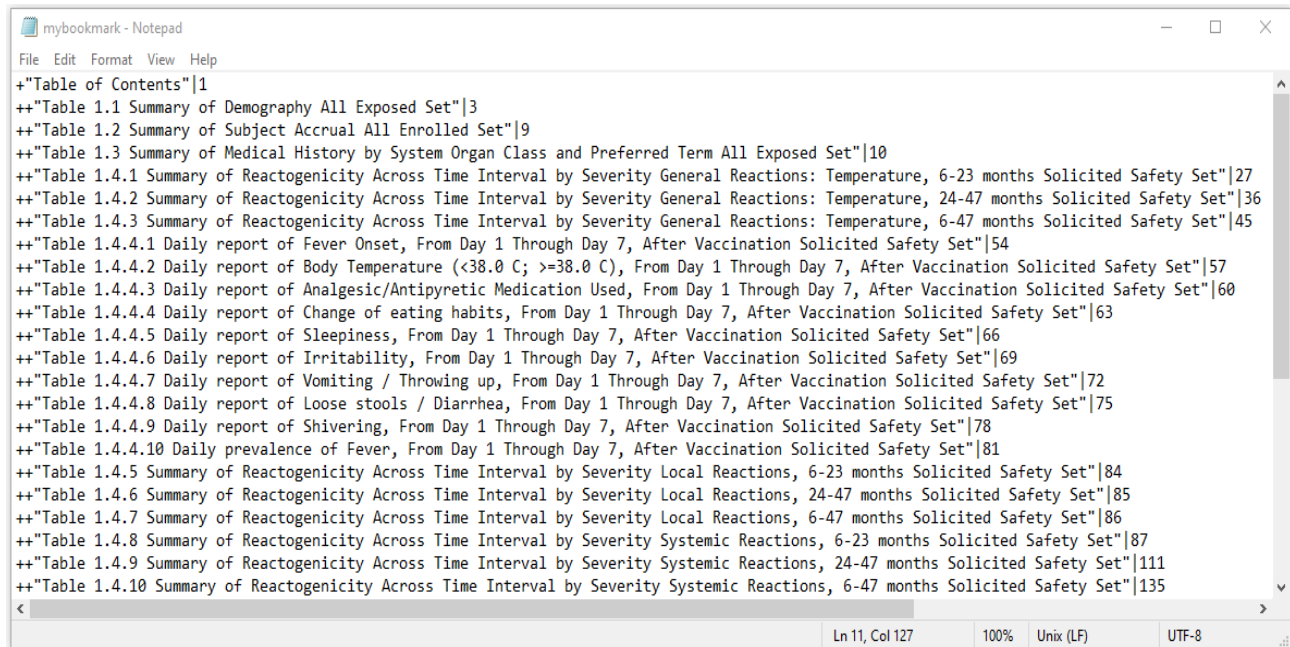
    # Save and Close
    word.ActiveDocument.SaveAs(
        new_file_abs, FileFormat= constant
    )
    doc.Close(False)
```

If the user selects the output type format as DOCX, the constant parameter set as 16 & extension set as ‘.docx’ and if user select the output type as PDF then, the constant set as 17 & extension set as ‘.pdf’.

STEP 4: Creating PDF bookmark (based on user input)

Once the PDF TOC is created, the bookmark.txt will be created using PYTHON function (`self.genPdfBookmark()`) as like shown in Figure 2

Figure 2 – pdfbookmark.txt structure used to create the PDF bookmark with the pdfbookmarker PYTHON package



```
mybookmark - Notepad
File Edit Format View Help
+"Table of Contents"|1
++"Table 1.1 Summary of Demography All Exposed Set"|3
++"Table 1.2 Summary of Subject Accrual All Enrolled Set"|9
++"Table 1.3 Summary of Medical History by System Organ Class and Preferred Term All Exposed Set"|10
++"Table 1.4.1 Summary of Reactogenicity Across Time Interval by Severity General Reactions: Temperature, 6-23 months Solicited Safety Set"|27
++"Table 1.4.2 Summary of Reactogenicity Across Time Interval by Severity General Reactions: Temperature, 24-47 months Solicited Safety Set"|36
++"Table 1.4.3 Summary of Reactogenicity Across Time Interval by Severity General Reactions: Temperature, 6-47 months Solicited Safety Set"|45
++"Table 1.4.4.1 Daily report of Fever Onset, From Day 1 Through Day 7, After Vaccination Solicited Safety Set"|54
++"Table 1.4.4.2 Daily report of Body Temperature (<38.0 C; >=38.0 C), From Day 1 Through Day 7, After Vaccination Solicited Safety Set"|57
++"Table 1.4.4.3 Daily report of Analgesic/Antipyretic Medication Used, From Day 1 Through Day 7, After Vaccination Solicited Safety Set"|60
++"Table 1.4.4.4 Daily report of Change of eating habits, From Day 1 Through Day 7, After Vaccination Solicited Safety Set"|63
++"Table 1.4.4.5 Daily report of Sleepiness, From Day 1 Through Day 7, After Vaccination Solicited Safety Set"|66
++"Table 1.4.4.6 Daily report of Irritability, From Day 1 Through Day 7, After Vaccination Solicited Safety Set"|69
++"Table 1.4.4.7 Daily report of Vomiting / Throwing up, From Day 1 Through Day 7, After Vaccination Solicited Safety Set"|72
++"Table 1.4.4.8 Daily report of Loose stools / Diarrhea, From Day 1 Through Day 7, After Vaccination Solicited Safety Set"|75
++"Table 1.4.4.9 Daily report of Shivering, From Day 1 Through Day 7, After Vaccination Solicited Safety Set"|78
++"Table 1.4.4.10 Daily prevalence of Fever, From Day 1 Through Day 7, After Vaccination Solicited Safety Set"|81
++"Table 1.4.5 Summary of Reactogenicity Across Time Interval by Severity Local Reactions, 6-23 months Solicited Safety Set"|84
++"Table 1.4.6 Summary of Reactogenicity Across Time Interval by Severity Local Reactions, 24-47 months Solicited Safety Set"|85
++"Table 1.4.7 Summary of Reactogenicity Across Time Interval by Severity Local Reactions, 6-47 months Solicited Safety Set"|86
++"Table 1.4.8 Summary of Reactogenicity Across Time Interval by Severity Systemic Reactions, 6-23 months Solicited Safety Set"|87
++"Table 1.4.9 Summary of Reactogenicity Across Time Interval by Severity Systemic Reactions, 24-47 months Solicited Safety Set"|111
++"Table 1.4.10 Summary of Reactogenicity Across Time Interval by Severity Systemic Reactions, 6-47 months Solicited Safety Set"|135
<
Ln 11, Col 127 100% Unix (LF) UTF-8
```

Code snippet 5: Creating PDF bookmark using pdfbookmarker PYTHON package

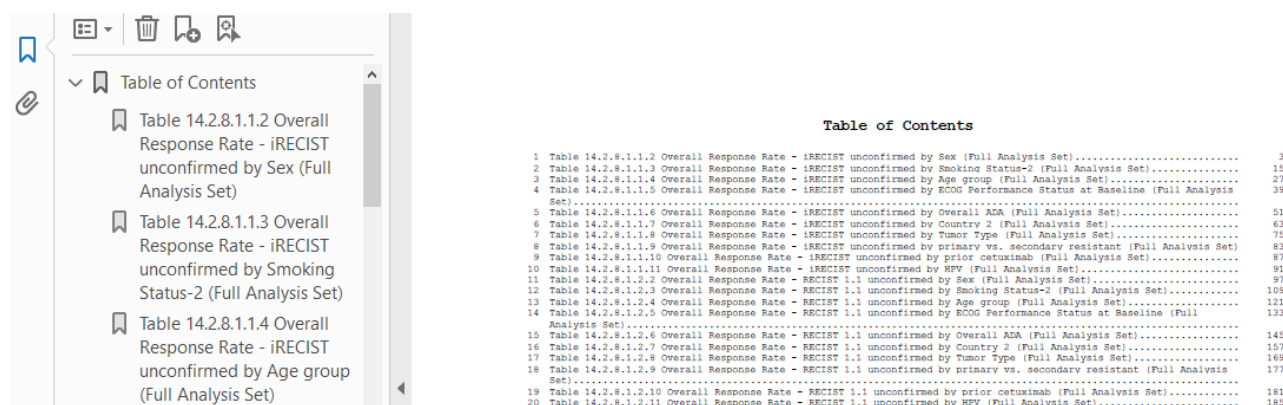
```
#Generate bookmark using pdfbookmarker
if self.pdf_bookmark==1:
    self.genPdfBookmark()
    run_script(self.pathout+re.sub(r'\.rtf$', '.pdf',self.fileout),self.pathout+"mybookmark.txt")
```

KEY PYTHON PACKAGES

- **Tkinter** is used in creation of this tool and is included in all standard PYTHON standard library.
- **Ttkwidgets** supports a variety of widgets as compared to Tkinter widgets and it is used to get the checkbox list for selected RTF files area (Figure 1)
- **Codecs** is encoders and decoders for converting text between different representations.
- **RE** is a regular expression specifies a set of strings that matches it.
- **win32com.client** package is used to convert RTF file into DOCX format
- **Pdfbookmarker** is used to create the PDF bookmarks
- **Pyinstaller** is used to convert PYTHON code to executable file.

FINAL TOC FILE

Figure 3 - PDF TOC File with bookmarks



The image shows a PDF viewer interface. On the left, there is a sidebar with a 'Table of Contents' section containing three bookmarked entries: 'Table 14.2.8.1.1.2 Overall Response Rate - iRECIST unconfirmed by Sex (Full Analysis Set)', 'Table 14.2.8.1.1.3 Overall Response Rate - iRECIST unconfirmed by Smoking Status-2 (Full Analysis Set)', and 'Table 14.2.8.1.1.4 Overall Response Rate - iRECIST unconfirmed by Age group (Full Analysis Set)'. On the right, a full 'Table of Contents' is displayed as a list of 20 items, each with a page number. The items are numbered 1 through 20 and follow a similar naming convention to the bookmarks, including details like 'Overall Response Rate' and 'iRECIST unconfirmed by [criteria] (Full Analysis Set)'. The page numbers range from 3 to 185.

Table of Contents	
1 Table 14.2.8.1.1.2 Overall Response Rate - iRECIST unconfirmed by Sex (Full Analysis Set).....	3
2 Table 14.2.8.1.1.3 Overall Response Rate - iRECIST unconfirmed by Smoking Status-2 (Full Analysis Set).....	15
3 Table 14.2.8.1.1.4 Overall Response Rate - iRECIST unconfirmed by Age group (Full Analysis Set).....	27
4 Table 14.2.8.1.1.5 Overall Response Rate - iRECIST unconfirmed by ECOG Performance Status at Baseline (Full Analysis Set).....	39
5 Table 14.2.8.1.1.6 Overall Response Rate - iRECIST unconfirmed by Overall ADA (Full Analysis Set).....	51
6 Table 14.2.8.1.1.7 Overall Response Rate - iRECIST unconfirmed by Country 2 (Full Analysis Set).....	63
7 Table 14.2.8.1.1.8 Overall Response Rate - iRECIST unconfirmed by Tumor Type (Full Analysis Set).....	75
8 Table 14.2.8.1.1.9 Overall Response Rate - iRECIST unconfirmed by primary vs. secondary resistant (Full Analysis Set).....	83
9 Table 14.2.8.1.1.10 Overall Response Rate - iRECIST unconfirmed by prior cetuximab (Full Analysis Set).....	87
10 Table 14.2.8.1.1.11 Overall Response Rate - iRECIST unconfirmed by HPV (Full Analysis Set).....	91
11 Table 14.2.8.1.2.2 Overall Response Rate - RECIST 1.1 unconfirmed by Sex (Full Analysis Set).....	97
12 Table 14.2.8.1.2.3 Overall Response Rate - RECIST 1.1 unconfirmed by Smoking Status-2 (Full Analysis Set).....	109
13 Table 14.2.8.1.2.4 Overall Response Rate - RECIST 1.1 unconfirmed by Age group (Full Analysis Set).....	121
14 Table 14.2.8.1.2.5 Overall Response Rate - RECIST 1.1 unconfirmed by ECOG Performance Status at Baseline (Full Analysis Set).....	133
15 Table 14.2.8.1.2.6 Overall Response Rate - RECIST 1.1 unconfirmed by Overall ADA (Full Analysis Set).....	145
16 Table 14.2.8.1.2.7 Overall Response Rate - RECIST 1.1 unconfirmed by Country 2 (Full Analysis Set).....	157
17 Table 14.2.8.1.2.8 Overall Response Rate - RECIST 1.1 unconfirmed by Tumor Type (Full Analysis Set).....	169
18 Table 14.2.8.1.2.9 Overall Response Rate - RECIST 1.1 unconfirmed by primary vs. secondary resistant (Full Analysis Set).....	177
19 Table 14.2.8.1.2.10 Overall Response Rate - RECIST 1.1 unconfirmed by prior cetuximab (Full Analysis Set).....	181
20 Table 14.2.8.1.2.11 Overall Response Rate - RECIST 1.1 unconfirmed by HPV (Full Analysis Set).....	185

LIMITATIONS

There are some limitations when generating the TOC file with this tool.

- The RTF outputs should have generated through use of SAS ODS RTF statement.
- SAS-generated RTF outputs should hold the title information in a general format (starting with "Listing/Table/Figure/Appendix" followed by their number and title details.
- The outputs are ordered based on the TLF/Appendix file names, so the user doesn't have the provision to re-order the TLF position. The future version of the tool might have the facility to re-order the TLFs based on the user's needs.

CONCLUSION

This paper presents a method to combine SAS-generated RTF files and create a hyperlinked table of contents using a fully automated process implemented by a PYTHON GUI-based executable file. It can combine listings, figures, tables, and appendixes with very limited requirements for the way individual RTF files are created.

REFERENCES

- Rich Text Format (RTF) Specification, version 1.9.1
- Foundation PYTHON Software. "PYTHON 3.9 Documentation." <https://docs.python.org/3/index.html>.
- Tkinter - <https://docs.python.org/3/library/tkinter.html>

CONTACT INFORMATION

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

Sathish Kumar N

Labcorp Drug Development India Private Limited,

Email: Sathish.kumar2@labcorp.com

Brand and product names are trademarks of their respective companies.