Using SAS$^@$ to Assemble Output Report Files into One PDF File with Bookmarks

Sam Wang, Merrimack Pharmaceuticals, Inc., Cambridge, MA
Kaniz Khalifa, Leaf Research Services, Inc., Southeastern, PA

ABSTRACT

Assembling multiple output report files (such as RTFs or PDFs) into a single PDF file is frequently needed in clinical trial development. The most current practices involve using an adobe product or other third party applications. However these practices are very time consuming for programmers to create or control the bookmarks. Also, default bookmarks by those applications need to use the file names or use certain lines of the titles from the outputs. The use of third party applications will occur additional cost and will require more time to learn and understand the new language syntax. Our approach is to assemble the output report files without using any third party's applications. We only use SAS$^@$ code to assemble all the output files from a specified destination into a single PDF file with bookmarks that are user defined.

INTRODUCTION

PROC REPORT is commonly used to generate RTF or PDF file outputs. After these files are generated, assembling them into a single PDF file becomes more difficult because it needs to accommodate the existing format and the contents of output files. This paper shows how to turn the post-process of file assembling into pre-process of file assembling by adding a few additional SAS codes before and/or into PROC REPORT statements. Before generating the output files, we keep the table/listing/figure numbers and their titles into macro variables and use them as bookmark values. When generating the RFT or PDF output files, we add an ODS DOCUMENT statement to hold the output objects in their original creation structure and format. After all the output files are generated, we use PROC DOCUMENT to replay or rerun those output objects into one designated PDF file. The order of object files and their bookmarks in the PDF file will be based on the stream order controlled by SAS$^@$ code.

UNDERSTANDING ODS PDF OUTPUT

Without any bookmark handling process being applied, the PROC REPORT with ODS PDF output and with BY statement will generate a table having four levels of bookmarks as shown in figure 1 as below.

Figure 1.
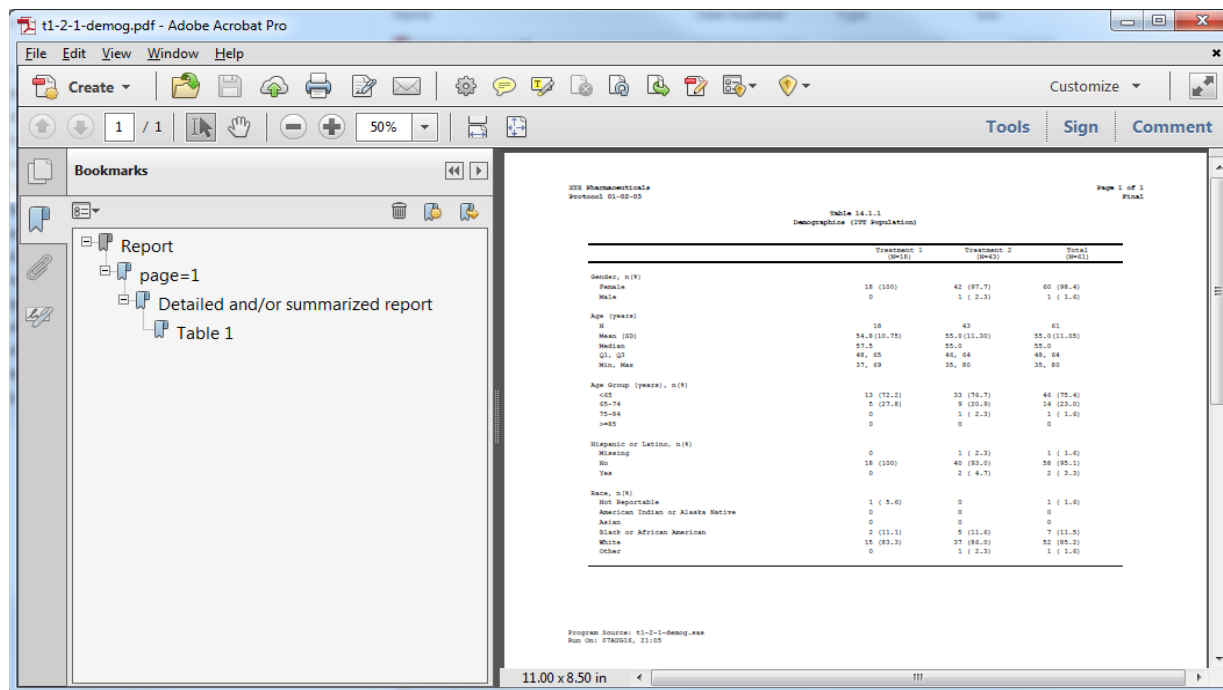
The bookmarks generated by PROC REPORT can be managed and controlled individually based on the needs.

1) To control the first level of bookmark, ODS PROCLABEL statement should be specified. Macro variables can be used to collect the table number and titles and embed them into ODS PROCLABEL statement. For example,

ODS PROCLABEL="&titlenum &titles";

2) To control the second level of bookmark, a BY statement in PROC REPORT should be specified. The BY statement used in PROC REPORT is for the purpose of outputting multiple pages.

3) To control the third level of bookmark, the CONTENTS option should be defined. For example,

PROC REPORT CONTENTS="";

Contents="" will remove this level of bookmark entirely.

4) To control the fourth level of bookmark, a BREAK BEFORE / PAGE statement with CONTENTS="" option should be used.

After above four bookmark control statements are applied, the bookmark content will become two levels as shown in figure 2 as below.

Figure 2.
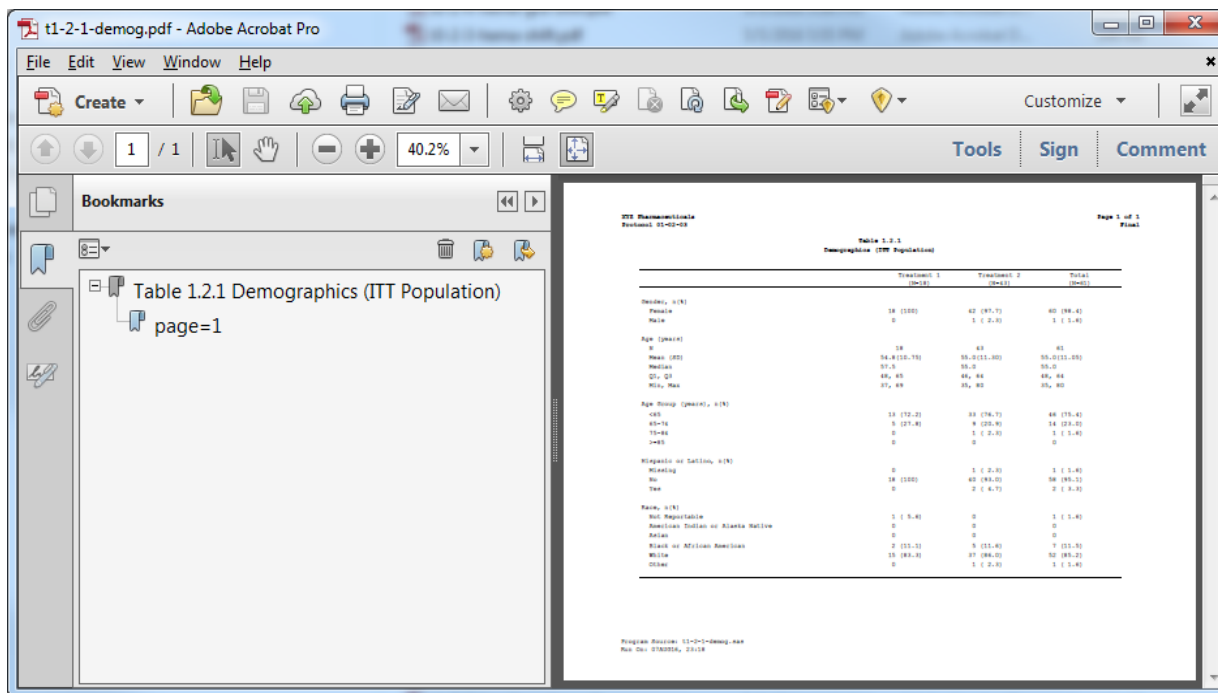
An option of PDFTOC=1 can be used in ODS PDF statement to hide second level of bookmark and make only one level of bookmark shown in each PDF output file.

UTILIZE ODS DOCUMENT AND PROC DOCUMENT FEATURE

The ODS DOCUMENT destination does not create any visual output like RTF and PDF. But it can be used as another layer to store and preserve the format of the output files. It gives a new level of power to manage the outputs.

When each output file such as RTF or PDF is generated, a corresponding document type file can be created at the same time by adding an ODS DOCUMENT statement before PROC REPORT statement. Each document file with extension of sas7bitm is saved at a designated directory for batch access when PROC DOCUMENT is conducted.

After all the induvial document files are generated, an assembling program can be executed. The assembling program consists three sections:

1) Define the destination of directory, then open the directory to get the file id and file name, output the file name by id into a SAS data set for all the files in the directory;

2) Pull all file names from the data set and assign file order from table output numbers. Based on the order number, stack all file names into a macro variable.

3) Loop out the filenames from macro variable and execute PROC DOCUMENT procedure with RELAY statement on each output file into an ODS PDF designated file.

The final assembled pdf file looks like following in figure 3.

Figure 3.
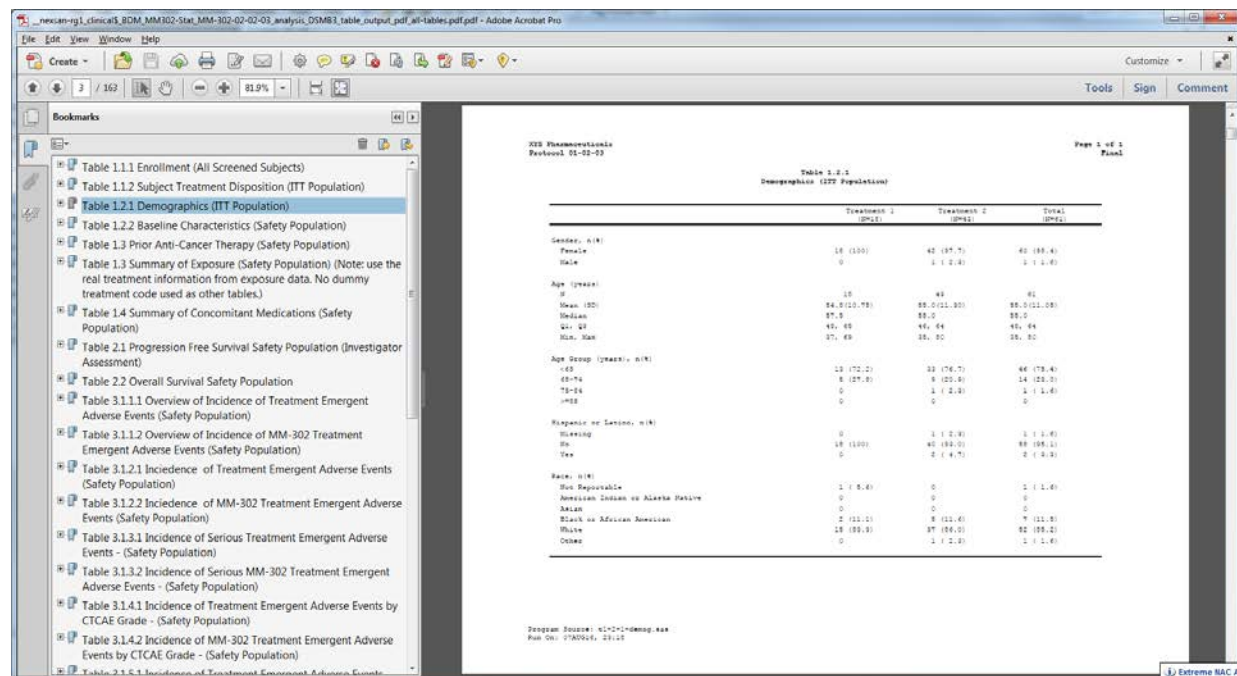
The assembling program code is defined as,

```
data allfile;
  keep filename;
  length fref $8 filename $80 ;
  rc = filename(fref, "&outpath.\document");
  if rc = 0 then do;
    did = dopen(fref);
    rc = filename(fref);
```

```
      end;
    else do;
      length msg $200.;
      msg = sysmsg();
      put msg=;
      did = .;
    end;
    if did <= 0 then putlog 'ERR' 'OR: Unable to open directory.';
    dnum = dnum(did);
    do i = 1 to dnum;
      filename = dread(did, i);
      fid = mopen(did, filename);
      if fid > 0 then output;
    end;
    rc = dclose(did);
  run;

  data allfile1;
    set allfile;
    length ord $20;
    tt=substr(filename,2);
    pos=indexc(tt,'abcdefghijklmnopqrstuvwxyz');
    numstr=substr(tt,1,pos-1);
    n=length(numstr) - length(compress(numstr,'_'));
    ord='';
    do i=1 to n;
      num=scan(numstr,i,'_');
      if length(num)=1 then do;
        num='0'||strip(num);
        ord=strip(ord)||num;
      end;
      else ord=strip(ord)||strip(num);
    end;
  run;

  proc sort data=allfile1 out=allfile2;
    by ord;
  run;

  data _null_;
    set allfile2 end=eof;
    length allname $3000;
    retain nn 0 allname '';
    allname=strip(allname) ||" "||strip(scan(filename, 1, "."));
    nn=nn+1;
    if eof then do;
      call symput("allname", allname);
      call symput("nn", strip(put(nn,4.)));
    end;
  run;

  %macro doit;
    options printerpath=pdf;
    ods pdf  style=mmrtf file="&outpath.\pdf\all-tables.pdf" pdftoc=1;
    %do i=1 %to &nn;
      %let docname=%scan(&allname, &i, " ");
      proc document name=docs.&docname;
```

```
        replay;
      run;
      quit;
    %end;
    ods pdf close;
  %mend;
  %doit;
```

In order to make individual output file in order and at the correct position within the PDF file, it is strongly recommended to use a table number as part of its file name.

CONCLUSION

PDF bookmarks can be generated and managed completely in SAS code. The ODS DOCUMENT destination and DOCUMENT procedure enable the structure and the format of files to be preserved. The files can be stacked together directly without any manipulations. Planning and adding a few more small steps in a new or an existing program can make the PDF file assembling with bookmark becomes much easier.

REFERENCES

Have It Your Way: Rearrange and Replay Your Output with ODS DOCUMENT, Cynthia L. Zender, SAS Institute Inc. Cary, NC

Problem Note 31278: Table 1 node generated by PROC REPORT. Support.sas.com/kb/31/278.html.

RECOMMENDED READING

- SAS® Output Delivery System: User's Guide, PROC DOCUMENT Statement

CONTACT INFORMATION

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

Name: Sam Wang
Enterprise: Merrimack Pharmaceuticals, Inc
Address: One Kendall Square, Suite B7201
City, State ZIP: Cambridge, MA 02139
Work Phone: 617-441-7678
E-mail: swang@merrimack.com

Name: Kaniz Khalifa
Enterprise: L.E.A.F. Research Services
Address: 1000 W. Valley Road, PO Box 505
City, State ZIP: Southeastern, PA 19399
Work Phone: 860-634-1414
E-mail: kkhalifa@leafresearchservices.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.