

Stacking Rich Text Format (RTF) - %SRiT

Duong Tran – Independent Contractor, London, UK

ABSTRACT

Reporting in Rich Text Format (RTF) is gaining more and more popular and as programmers gain more experience on this topic, RTF is set to be a mainstream reporting format. In the paper %RiTEN^[1], the author discussed the techniques for framing cells, merging cells and tables stacking and how the implementation of these techniques gives it an immense powerful tool for reporting. %SRiT (Stack Rich Text) in addition to these techniques gives another level of flexibility to reporting in RTF. %SRiT is a SAS® macro that stacks RTF files. Its key objective is to stack the input RTF files but keep the originality of each individual file and have an option to reset the page counting. Therefore one can either collate the RTF files into a single report or a deliverable document , typically for review.

INTRODUCTION

There are occasions when stacking RTF files together to produce a single report is a much easier / quicker option. Imagine you get an ad hoc request to break down the report you have done, by category - This can often means significant changes to the structure of your program or often can be a challenge to adapt because the program was written by another programmer. In this situation %SRiT is very handy, you can just loops the reporting program per category and stack the results together. Other Scenarios where you want to stack two outputs generated from different standard reporting macros together but do not have a utility to do so. Another function of %SRiT is for stacking RTF outputs into a document for review or delivery purpose.

The content of an RTF file has the following syntax.

```
<File>                {<header><document>}
```

And with some experimenting it is observed that the RTF files could be stacked as follows. That is removes the appropriate braces and insert a separator.

```
<File1>    {<header1><document1>} ← remove }
           {\par} or {\sect} or {\page}

<File2>    {<header2><document2>} ← remove {}

           {\par} or {\sect} or {\page}

<File3>    {<header3><document3>} ← remove {}
           etc
           }
```

An important requirement when stacking the individual files is that each individual file going into the stacked file remains entirety to its originality. That is font types, sizes and margins etc. For this you must apply a section break (i.e. the {\sect} control word) between the files. You might need to experiment and amend some other control words. Things to bear in mind are - inheritance, defaults setting etc.

A PRACTICAL EXAMPLE

Assume you have two outputs as in figure 1 and would like to combine them into a single output as in figure 2. Note a typical requirement is to reset the page counting as indicated.

Figure 1

Table 1 Patients			Table 1 Patients		
Male			Female		
Name	Age	Height	Name	Age	Height
Alfred	14	69	Alice	13	56.5
James	12	57.3	Carol	14	62.8

Page 1 of 1 ← pagination per output → Page 1 of 1

program: demo.sas
output: male.rtf

program: demo.sas
output: female.rtf

Figure 2

Table 1 Patients		
Male		
Name	Age	Height
Alfred	14	69
James	12	57.3

Page 1 of 2

program: demo.sas
output: combined.rtf

.....Page Break.....

Table 1 Patients		
Female		
Name	Age	Height
Alice	13	56.5
Carol	14	62.8

Page 2 of 2

program: demo.sas
output: combined.rtf

re-pagination after combining

```
%SRiT(ifile =      /* input files      */
      ,ofile =      /* output file      */
      ,target =     /* XXX of YYY */
      ,resetpg =    /* reset page (Y|N) */
      ,splitch =    /* split char */
      ,difiles =    /* delete input files */
      );

:
etc

* read in each input file;
%do i = 1 %to &count;
  data rtf&i;
    length fname&i $80;
    infile "%qscan(&ifile, &i, %str(&splitch))" length=linelen;
    filename=fname&i lrecl=32767 recfm=v;
    input linein $varying2000. linelen;
    filename = fname&i;
    seq = &i;
  run;
%end;

data srit;
  * stack input files;
  set rtf1(in=a)
  %do i = 2 %to &count;
    rtf&i
  %end;
  end=eof
  ;
  by seq;
  if a then do;
    * remove last } of first input file;
    if last.seq then linein = substr(linein, 1, length(linein)-1);
  end;
  else do;
    * add page;
    if first.seq then linein = "\page" || linein;
  end;
  if eof then do;
    * close rtf doc;
    linein = trim(left(linein)) || " ";
  end;
end;
```

```
end;
run;
:
Etc

* write file;
data _null_;
  set srit;
  file "&ofile" lrecl = 2000;
  length mofn $20;

if index(linein, "XXX of YYY") then do;
  mofn = "Page " || compress(put(pcount,8.)) || " of " || "&totpages";
  linein = tranwrd(linein, "XXX of YYY", mofn);
end;
linelen =length(linein);
put linein $varying2000. linelen;
run;

%mend;
```

One other main aspect to consider is resetting the page count. If you have the literal text - Page 1 of 3, Page 4 of 5 etc in the output then first you can use a regular expression to change them to a tag say (XXX of YYY) then use this for re-paging. Below is a section of codes to do this. See also Worley^[2] on how to deal with this if the page counting was dynamic.

```
* count page x of y;
data srit;
  set srit end=eof;
  retain rx; drop rx;
  * perl regular expression would be better here;
  if _n_=1 then rx=rxparse(" 'Page ' $d+ ' of ' $d+ to 'XXX of YYY' ");
  call rxchange(rx,99999,linein);
  retain pcount 0;
  if index(linein, "XXX of YYY") > 0 then do;
    pcount = pcount + 1;
  end;
  if eof then call symput('totpages', trim(left(put(pcount, 6.))));
run;
```

So the call for the above result would be something like:

```
%SRiT(ifile = male.rtf {\page} # female.rtf
      ,ofile = combined.rtf
      );
```

And if you have multiple files to combine then you would have something like:

```
ifile = one.rtf {\page} # two.rtf {\par} # three.rtf
```

Note the # is a separator whereas {\page} and {\par} is to specify a page or line break respectively. Also using wild card is allows. For example if (ifile = *.rtf) then all the rtf files would be combined. This option could be used to combine the rtf files for delivery purpose but Visual Basic Application (VBA) is probably a better choice for this job.

CONCLUSION

The ability to stack RTF files gives another level of flexibility to reporting. This functionality can reduce production time and enable you to response quickly to ad hoc request. Also a utility such as %SRiT can be an excellent complimentary tool for patient profile reporting and alike.

REFERENCES

- [1] Tran, Duong “%RiTEN” – PhUSE 2007
[2] Worley, Jim “Combining RTFs While Retaining Each Original Document's Page Numbering” – PharmaSug 2007

CONTACT INFORMATION

Author Name: Duong Tran
Email: trand000@aol.com
Web: www.tranz.co.uk