

%RiTEN

Duong Tran, Independent Consultant, London, Great Britain

ABSTRACT

For years SAS® programmers within the Pharmaceutical industry have been searching for answers to produce tables and listings for clinical study reports to MS Word® format. There have been numerous techniques^[3] but all these, though innovative, either less convincing or requires considerable programming that they have not caught on. The advent of the ODS RTF (Rich Text Format) destination beginning SAS v8.2, in my opinion has and will make these techniques and alike obsolete.

The purposes of this paper are to:

- 1) Introduce the %RiTEN (Rich Text Enhancer) utility and thus hope to inspire you to **fully** move forward to the next generation of reporting formats with the ODS.
- 2) Highlight the three **KEY** functions (**cells framing, cells merging and tables stacking**) that the Proc Report ODS RTF destination still lacks.

INTRODUCTION

%RiTEN is a SAS post-processor macro written to enhance reports generated from Proc Report ODS RTF. That is the macro can add RTF features such as **cells framing, cells merging and tables stacking** to the reports, plus options for styling. The three key functions of %RiTEN gives **immense flexibility** to the type of tables and listings that can be created. So with the aid of this utility, clinical studies can be reported efficiently to the RTF format (see examples in Appendix 1)

FRAMING, MERGING CELLS AND STACKING TABLES

Why these three key functions are so often wished for ?

- 1) Within a table quite often one would like to emphasise result(s) or section(s) etc and framing will highlight these.
- 2) For each Proc Report called, the number of columns defined will persist throughout, but often a table consist of variable number of cells per row so cell merging will achieve this.
- 3) The ability to create separate tables with different Proc Report calls and stack them together gives an immense flexibility to type of tables/listings that can be produced.

For 1 and 2 you can achieve these by tagging the cell(s) value(s) (e.g. "{\cm\t\bl} value" – cell merge, top line, bottom line) and post-process the RTF file accordingly and for 3, this is just one line of SAS code to change ...**\par** to ...**\}** (i.e. assuming you are using the bodytitle option)

So with a combination of these three facilities to enhance the RTF file from Proc Report - the type of tables/listings that can be created is up to ones creativity. Example 1 and its result, Figure 1 shows how these three functions are applied in practice. See Appendix 1 for more practical examples.

Example 1

```
title1      j=c "{\fs30 Proc Report ODS RTF{\line}+{\line}{%}RiTEN}";
footnote;

options orientation=Portrait;

ods listing close;
ods rtf file="&outfile" style = PortraitLetterFS10 bodytitle;
```

```

proc report data = fin1 missing nowindows split = "#"
    style(report) = [rules=groups cellspacing=3pt cellpadding=1pt borderwidth=10]
    ;
    column c2 c1 c3;
    define c2 / display style(column) = {background=cyan cellheight=0.7in cellwidth=20%};
    etc..
run;

title;

proc report data = fin2 missing nowindows split = "#" noheader
    style(report) = [frame=vsides cellspacing=0.75pt cellpadding=1pt borderwidth=10]
    ;
    column c1 c2 c3 c4 c5 c6 c7;
    define c1 / display style(column) = {background=green cellwidth=50 cellheight=1in};
    etc..
run;

footnote1 j=r "D Tran";

proc report data = fin2 missing nowindows split = "#" noheader
    style(report) = [rules = groups cellspacing=3pt cellpadding=1pt borderwidth=10]
    ;
    column c1;
    define c1 / display style(column) = {background=blue cellheight=0.5in};
run;

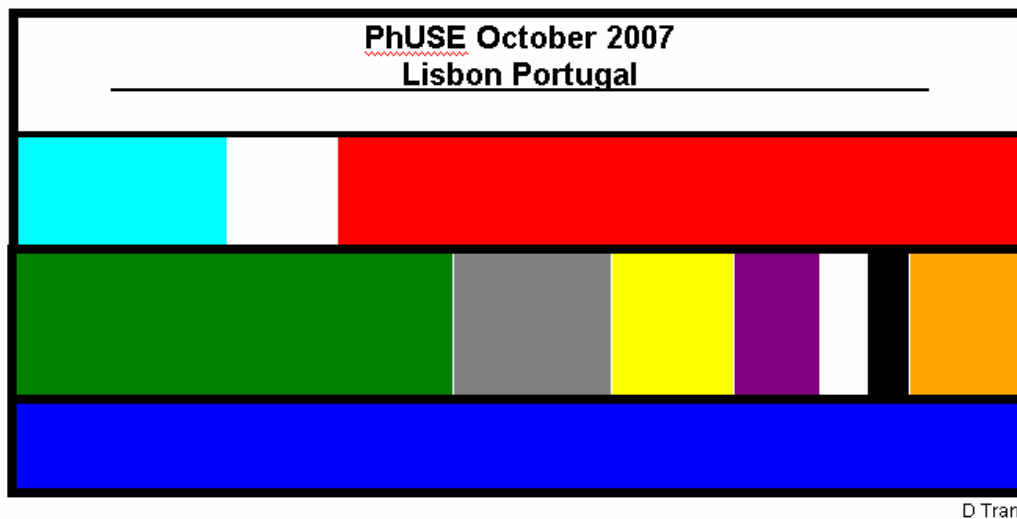
ods rtf close;
ods listing;

%RiTEN(ifile = %str(&outfile), ofile = %str(&outfile), target = %str(XXX of YYY), etc..);

```

Figure 1

**Proc Report ODS RTF
+
%RiTEN**

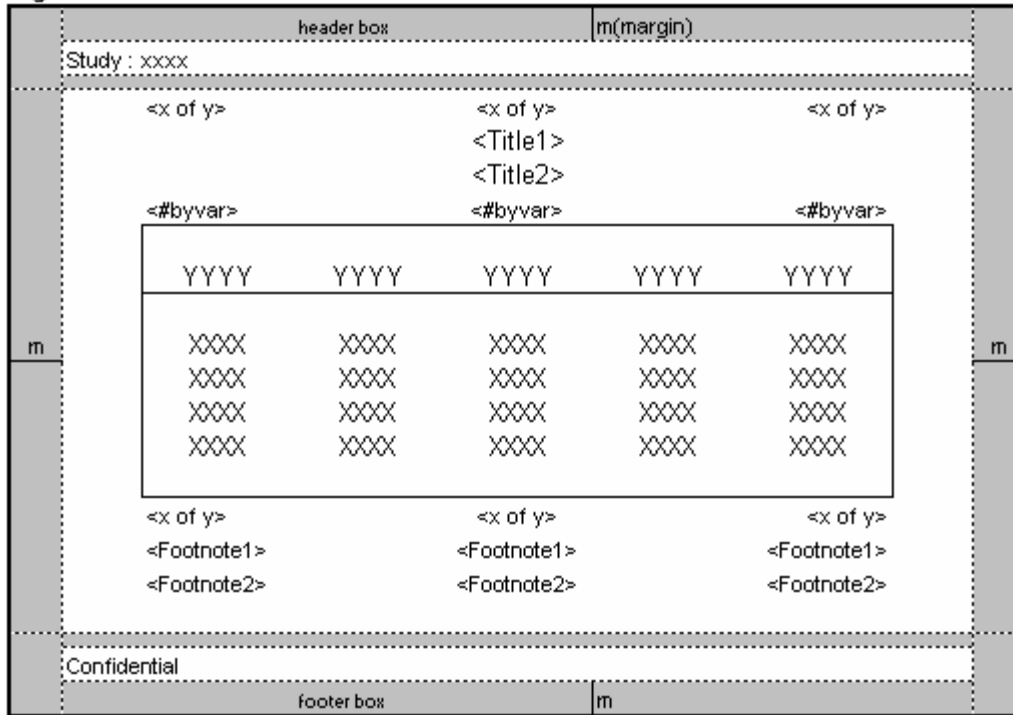


STYLES AND ATTRIBUTES

SAS provides numerous table styles and element attributes but some of these attributes can not be set by SAS at present. For example the header and footer text, the specific alignment of <#byval> and <x of y>. Figure 2 shows typical table styles and attributes that an organisation may adopt. Figures 3.x shows others typical table styles. Note the contents within <> indicates that it is optional. The titles may be left, centre or right justified and span over the width of the page or over the width of the table. The <x of y> can be of any of these styles - <pre-text> <x of y | x / y | x> <post-text> (e.g. Page x of y, Page [x of y], p.[x /y], Page x etc). That is <pre-text> and <post-text> is of your own choice and you have the three page numbering styles <x of y | x / y | x>.

Beside the three key functions mentioned above, %RiTEN can handle all the attributes styling in Figure 2 and table styling in Figure 3.x.

Figure 2



These are some of the others table styles you could chose by setting the rules=groups and farne=hsides | bottom | top | void

Figure 3.1

YYY	YYY	YYY
XXX	XXX	XXX
XXX	XXX	XXX
XXX	XXX	XXX

Figure 3.2

YYY	YYY	YYY
XXX	XXX	XXX
XXX	XXX	XXX
XXX	XXX	XXX

Figure 3.3

YYY	YYY	YYY
XXX	XXX	XXX
XXX	XXX	XXX
XXX	XXX	XXX

Figure 3.4

YYY	YYY	YYY
XXX	XXX	XXX
XXX	XXX	XXX
XXX	XXX	XXX

SUMMARY

The ability to stack tables and merge cells gives you the power to construct a composite (complete) table to however you like. There have been numerous papers in recent past where authors concluded that the ODS RTF destination is complex, unsatisfactory etc. When I first started on this subject area I share the same views, but no longer. In fact my views on this now are the opposite. That is not only I know 'all' tables and listings can be done with Proc Report ODS RTF to the typical pharmaceutical industry layout requirement, in many ways it is even easier than producing them in ASCII format. If RTF is just an ASCII text file of control words (instructions) that render text - then why should post-processing to manipulating instructions that display text be more difficult than manipulating the text itself? 'Surely' it is the other way round, as long as SAS emit RTF text in a consistent structure which it has. In SAS v9.2 you can define your own RTF Tagsets, so possibly your RTF post-processor will be even simpler. There are others proposed solutions, example using VBA etc. For me what ever solutions and technologies are used, they 'must' be a one step process, that is the rendering / post processing 'should' be called within the reporting program. Also the solution(s) must be viable and 'complete' otherwise it is difficult to convince your sponsor to make the transition from ASCII. But if you can show your manager a solution and that the clients can have a much higher quality format than ASCII at the same cost or even less then this transition is inevitable.

The biggest obstacle but the least deal with the ODS RTF in my opinion is pagination but Jiang and Boisvert^[1] has came up with a simple but practical solution.

CONCLUSION

In my opinion it is no longer sensible to generate tables/listings in ASCII then insert them into MS Word or have some kind of utilities to do the conversion when the ODS RTF destination is available. A major advantage of the direct approach is that Statisticians and Clinicians can incorporate the RTF files into their reports, which are most certain to be written in MS Word with minimal changes, if any at all. Another obvious advantage is that the visual quality of the RTF format. There are other ODS destinations for consideration for Clinical Reporting – PDF, XML, HTML etc. so the future is optionally rich – possibly “Rich Text Format”.

I mentioned at the beginning that I believe techniques^[3] will be obsolete – I also believe that the ASCII format is slowly phasing out.

REFERENCES

- [1] PhamaSug 2006: Effective Strategy to Set Page Break for ODS RTF outputs (Songtao Jiang; Daniel Boisvert)
- [2] SUGI 21: %Print Drivers: teaching SAS Software to Speak (Paul Wehr)
- [3] SUGI 27: Output Generating System – A Tool for Creating Tables and Listings in Word: (Jay Zhou)
- [4] PhUSE 2007: Inline Formatting (Duong Tran)
- [5] RTF Pocket Guide: (Sean M. Buke)

CONTACT INFORMATION

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

Duong Tran
Tranz Ltd
Phone: 07752326413
Email: trand000@aol.com
www.tranz.co.uk

Other brand and product names are trademarks of their respective companies.

APPENDIX 1 (more examples for download at www.tranz.co.uk)

Example 1 Framing Cells

Maximum Post-Baseline	Group A			Group B		
	1	2	Total	1	2	Total
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
Total	0	0	0	0	0	0

Page 1 of 1

Program: `t_example1.sas`

Output: `t_example1.rtf` (Date Generated: 08AUG07:22:43:37) Source Data: `xxxx.sas7bdat`

Example 2 Cells Merging

	Placebo (N=xxx)	Drug (N=xxx)
Treatment Effects		
Least squares mean	xx.x	xx.x
SE	xx.x	xx.x
95% CI	(xx.x, xx.x)	(xx.x, xx.x)
Treatment Difference		
Least squares mean	xx.xxxxx	
SE	xx.xxxxx	
95% CI	(xx.xxxxx, xx.xxxxx)	

Page 1 of 1

Note the Treatment Difference lies between the treatments.

Program: `t_example2.sas`

Output: `t_example2.rtf` (Date Generated: 08Aug07:23:35:25) Source

Data: `xxxx.sas7bdat`

Example 3 Stacking Tables

This act like a lid when all the tables are collapsed.

<gap between tables>		
startdate	stopdate	aept
ddmmyyy	ddmmyyy	xxx yyy zzz
<gap between tables>		

ref no

This is a very very long text

<gap between tables>

We can have a lid at the bottom also.

Page 1 of 20

Tables have not been collapsed for demonstration purpose. Set `tgap=0` in `%RiTEN` to do so.

Program: `t_example3.sas`

Output: `t_example3.rtf` (Date Generated: 08AUG07:22:46:26) Source Data: `xxxx.sas7bdat`

APPENDIX 2

%RiTEN parameters

```
%RiTEN(ifile      =          /* in file          */
      ,ofile      =          /* out file         */
      ,target     = %str(XXX of YYY) /* target to be replace by Page x of y */
      ,rgap       = N        /* remove the gap between the titles and table */
      ,tgap       = 0        /* table gap        */
      ,footw      =          /* in twips. -1 to use li/ri tags          */
      ,repfoot    = Y        /* replace footnote  */
      ,reptit     = Y        /* replace title      */
      ,lrcpad     = 60       /* left/right cell padding in twips        */
      ,pnostyle   = Page [1] /* Page numbering style [1] | [2] | [3]    */
      ,headery    =          /* offset from top edge in twips          */
      ,footery    =          /* offset from bottom edge in twips       */
      ,headtxt    = %str()   /* header text       */
      ,foottxt    = %str()   /* footer text       */
      );
```