

## Generate Customized Table in RTF Format by Using SAS Without ODS RTF – RTF Table File Demystified

Kai Koo, Abbott Medical Devices, Santa Clara, CA

### ABSTRACT

Rich Text Format (RTF), developed and maintained by Microsoft, is a popular document file format supported by many word processors in different operating systems. Because RTF data file is human-readable, experienced programmers can write their own RTF document, including tables and graphs directly based on the RTF specification. Although skilled SAS users can generate high quality RTF tables through ODS RTF and related procedures (e.g., PROC REPORT/TEMPLATE), tables in RTF format can also be produced by SAS DATA step without any ODS RTF codes involved. A SAS DATA step-based macro for RTF table preparation will be introduced in this article. The basic concept of coding tables in RTF format will be explained. The advanced techniques used in this macro for setting margins, header, footer, title, footnote, font types, highlight, border, column width, cell merging, and different approaches of inserting/embedding image file will also be demonstrated. SAS users can use those presented concepts and logics to develop their own SAS codes or even macro as an ODS RTF alternative to prepare customized tables for thesis writing, presentation, publication, and regulatory submission.

### INTRODUCTION

Many SAS users output their statistical analysis results in the formats of tables and listings through SAS ODS RTF system. Those RTF files can be opened, reviewed and further edited by many popular RTF-friendly word processors, such as Microsoft Word or LibreOffice, in different operating systems. The content of RTF file looks complicate through a plain text editor, even it is called “human readable”. Fortunately, the specification of RTF is well documented and publicly available, users are able to understand the file content fully by following the instructions and statements of this specification.

The RTF specification describes different commands to fine tune the document style, page orientation, margins, font type/size and paragraph for generating well-formatted documents. In this article, we are not going to discuss all those commands in detail; only essential commands, control words and required SAS coding concepts for preparation of high quality table/listing will be demonstrated step by step. Through those concepts, readers will find out it is not difficult to build customized tables in RTF through SAS directly, even without the help of ODS RTF.

### CONCEPT 1 – A very simple table in RTF

RTF file structure for table is simple. A simple 2 x 2 table in RTF can be built by using 13 short lines of RTF statements (code and comments listed below).

4	3
2	6

**Table 1.** A very simple 2 x 2 table

	<b>RTF code</b>	<b>Comment</b>
1.	<code>{\rtf1</code>	The is header statement. It tells the RTF compatible word processor this is a RTF file based on the specification version 1.x.
2.	<code>\trowd</code>	It indicates the 1 <sup>st</sup> table row started
3.	<code>\cellx2000</code>	To define the width of 1 <sup>st</sup> column: “\cellx2000” will be interpreted as the 2000 twips = 1.39 inch [ <i>i.e.</i> , 1440 ( <b>tw</b> entieth of an <b>in</b> ch <b>po</b> int) = 1 inch]

- |     |            |   |
|-----|------------|---|
| 4.  | \cellx4000 | To define the total width of 1 <sup>st</sup> and 2 <sup>nd</sup> column: “\cellx4000” will be interpreted as 4000/1440 = 3.78 inches ( <i>i.e.</i> , the width of 2 <sup>nd</sup> column is also 1.39 inches) |
| 5.  | {4}\cell   | Content of cell (column 1 of 1 <sup>st</sup> row): 4  |
| 6.  | {3}\cell   | Content of cell (column 2 of 1 <sup>st</sup> row): 3  |
| 7.  | \row       | To tell the word processor the 1 <sup>st</sup> table row ended  |
| 8.  | \trowd     | Indicate the 2 <sup>nd</sup> table row started  |
| 9.  | \cellx2000 | As same as line 3   |
| 10. | \cellx4000 | As same as line 4   |
| 11. | {2}\cell   | Content of cell (column 1 of 2 <sup>nd</sup> row): 2  |
| 12. | {6}\cell   | Content of cell (column 2 of 2 <sup>nd</sup> row): 6  |
| 13. | \row}      | To tell the word processor this table row and RTF file ended  |

For SAS users, the following sample code will generate the table displayed in this section through the FILE and PUT statements through a \_NULL\_ DATA step.

```

data _null_;
  file "C:\TEMP\PharmaSug 2019\example1.rtf" lrecl = 1000;
  put '{\rtf1';
  put '\trowd';
  put '\cellx2000';
  put '\cellx4000';
  put '{4}\cell';
  put '{3}\cell';
  put '\row';
  put '\trowd';
  put '\cellx2000';
  put '\cellx4000';
  put '{2}\cell';
  put '{6}\cell';
  put '\row}';
run;

```

### CONCEPT 2 – A 2 x 2 table with different border settings

(a)

4	3
2	6

(b)

4	3
2	6

**Table 2.** (a) Table cell with all borders, (b) table with top and bottom borders only

Those two tables with additional boarder settings still can be generated by reusing those 13-line RTF statements described in CONCEPT 1, except including extra border control words in lines 3,4, 9 and 10 (see below).

```
\clbrdrb\brdrs\clbrdrt\brdrs\clbrdr1\brdrs\clbrdrx\brdrs\cellx2000 [table 2(a)]
```

```
\clbrdrt\brdrs\cellx2000 [row 1 of table 2(b) – only the top border line displayed]
```

```
\clbrdrb\brdrs\cellx2000 [row 2 of table 2(b) – only the bottom border line displayed]
```

In this updated RTF code, those control words, \clbrdrb, \clbrdrt, \clbrdr1, \clbrdrx, tell word processor to add bottom, top, left and right borders to the data cell, respectively. The additional \brdrs keyword further defines the thickness of border line is “single-thickness”.

### CONCEPT 3 – Change font type, size, alignment, page margin and orientation

4	3
<b>2</b>	6

**Table 3.** Table with different font settings

Changing the default font setting (pre-determined by the word processor) to user defined front type and size can be done by adding font definition statement after the “{\rtf1” RTF header.

```
{\rtf1\ansi\deff0{\fonttbl{\f0 Arial Narrow;}}
```

The “\ansi” defines the character set as ANSI and the following “\deff0” defines the document font group as font #0. The font type of this font #0 will be further described by the font table statement {\fonttbl{\f0 Arial Narrow;}}. In this example, the user-defined font type of this document will be “Arial Narrow”.

As mentioned in CONCEPT 1, RTF uses “twip” as measurement unit. The following page information statement describes the paper width (\paperw) as 12240/1440 = 8.5 (inches) and paper height is 15840/1440 = 11 (inches) and margins for left, right, top, and bottom of this document are all 1 inch.

```
\paperw12240\paperh15840\margl1440\margr1440\margt1440\marginb1440
```

If users switch the control words from “\paperw12240 \paperh15840” to “\paperw15840\paperh12240”, the page orientation will be altered from portrait to landscape.

The alignment of cell content can be adjusted individually by adding \qc (centered), \ql (left-justified) or \qr (right-justified) before the cell content. The alignment of cell content can be changed from center (default) to left by modification of the original RTF description from {4}\cell to \ql{4}\cell.

To further change the font size of each row, RTF uses \fsN as the control word to define font size (e.g., N=24 “half-points” will be interpreted as 24/2=12 pt). We can either put \fsN ahead (\fs24\trowd) or behind (\trowd\fs24) the row starting statement. The word processor will not change the font size until another new \fsN command be detected.

### CONCEPT 4 – Highlighting table row and merging table cells

	Treatment 1	Treatment 2
<b>Outcome 1</b>	4	3
<b>Outcome 2</b>	2	6
p-value <sup>1</sup>	0.3147	

**Table 4.** Table with highlighted background, superscript font and merged cells

In comparison with table 4 shown here with tables presented in previous sections, readers will find out that one additional column and two more rows are added.

Adding extra rows and column is straightforward by using the cell control words demonstrated in CONCEPT 1 and CONCEPT 2 sections. For generating “highlighted” cells, users need to attach a color table to the RTF header after font setting. If the “light gray” will be used, the color should be presented by using the combination of 8-bit RGB values (211, 211, 211).

```
{\rtf1\ansi\deff0{\fonttbl{\f0 Arial Narrow;}}  
{\colortbl;\red211\green211\blue211;}
```

To highlight a cell with pre-defined color or pattern, the cell background control word, \clcbpatN should be attached to the other cell formatting commands. In the example listed below, the word processor will display the background color defined in color table #1 of a specific cell.

```
\clcbpat1\clbrdrb\brdrs\clbrdrt\brdrs\clbrdr1\brdrs\clbrdrr\brdrs\cellx2000
```

In cell merging, the following sample RTF control word segments (a) and (b) demonstrated the way of cell merging. To merge 2<sup>nd</sup> and 3<sup>rd</sup> cells (in the row), no extra cell-merging command is needed; user can just simply make it happen by removing the cell control words of the 2<sup>nd</sup> table cell without any coding changes for the 3<sup>rd</sup> cell.

**(a) a row contains 3 data cells – total table width: 5000 twips**

```
\clbrdrb\brdrs\clbrdrt\brdrs\clbrdr1\brdrs\clbrdrr\brdrs\cellx2000
\clbrdrb\brdrs\clbrdrt\brdrs\clbrdr1\brdrs\clbrdrr\brdrs\cellx3500
\clbrdrb\brdrs\clbrdrt\brdrs\clbrdr1\brdrs\clbrdrr\brdrs\cellx5000
```

**(b) a row contains 2 data cells – total table width: 5000 twips**

```
\clbrdrb\brdrs\clbrdrt\brdrs\clbrdr1\brdrs\clbrdrr\brdrs\cellx2000
\clbrdrb\brdrs\clbrdrt\brdrs\clbrdr1\brdrs\clbrdrr\brdrs\cellx5000 [cell 2 and cell 3 defined in (a) will be merged
into 1 cell]
```

**CONCEPT 5 – Customized table title, footnote, header and footer**



**Table 1. A 2 x 2 Contingency Table**

	Treatment 1	Treatment 2
Outcome 1	4	3
Outcome 2	2	6
p-value <sup>1</sup>	0.3147	

<sup>1</sup>Fisher's exact test.

Demonstration

**Table 5.** A well-formatted table with header, footer, title, and footnote.

The table 5 presented in this section is further formatted with additional table title, footnote, header, and footer by inserting following 4 lines control words below color table setting described in CONCEPT 4. Users can use the basic structure of a paragraph in RTF, “\pard\other settings + content of the paragraph \par”, to separate different text sections for header, footer, title and footnote. For further adjustment of font position, RTF provides the \sub for subscript, \super for superscript, and \nosupersub to reset the font position back to the normal.

```
{\header\pard\fs20 \qc{PharmaSug 2019}\par} (define header)
{\footer\pard\fs20 Demonstration \par} (define footer)

\pard\fs24\ql \b Table 1. A 2 x 2 Contingency Table\b0 (define title)

\par\ql\fs18 \super 1 \nosupersub Fisher's exact test. (define footnote)
```

If users can digest and use those essential concepts described from CONCEPT 1 to CONCEPT 5, it should not be difficult to generate well-formatted tables in RTF by writing short SAS code. Coders can even code their own SAS macro to prepare similar tables in RTF from SAS dataset based on those described RTF rules. A sample SAS macro code used to produce the table 5 from a SAS dataset (table 6) is listed below to serve as an example in summarizing those coding concepts discussed in these five sections. This sample code also takes advantage of the design that RTF allow users adding non-executable “comment” (*i.e.*, word processor will ignore it) in the format of `{*\ comment string;}`, so the users can write a customized comment, such as `{*\ Image Inserting Point;}`, to instruct SAS code to preserve the location as the inserting point of an embedded image file.

	c0	c1	c2
1	\b Outcome 1\b0	4	3
2	\b Outcome 2\b0	2	6
3	p-Value \super 1 \nosupersub		0.3147

**Table 6.** Sample SAS dataset used for RTF generation

```
%macro RTF_DEMO1(in=, outfn=, title=, footnote=, hdr=, ftr=);

/* Define the general cell format - border lines */
%let cfmt=\clbrdrb\brdrs\clbrdrt\brdrs\clbrdl\brdrs\clbrdr\brdrs;

data _rtftab_;
  set &in end=p;
  file "&outfn" lrecl = 1000;

  length tmp $1000.;

  if _n_=1 then do;
    /* Part 1 of RTF - Global setup file header, font type and color table */
    put '{\rtf1\ansi\deff0{\fonttbl {\f0 Arial Narrow;}}';
    put '{\colortbl;\red211\green211\blue211;}' ;
    put '\paperw12240\paperh15840\margl2880\margr2880\margt1440\margb1440';

    /* Part 2 of RTF - Table title, header and footer */
    put "{\header\pard\fs20 &hdr\par}";
    put "{\footer\pard\fs20 &ftr\par}";
    put "\pard\fs24\&title";
    put '\par\fs10\par';

    /* Part 3 of RTF - column labels */
    put '{\trowd\fs20';
    put "&cfmt\cellx2000";
    put "\clcbpat1&cfmt\cellx3500";
    put "\clcbpat1&cfmt\cellx5000";
    put '\qc{ }\cell';
    put '\qc{\b Treatment 1 \b0}\cell';
    put '\qc{\b Treatment 2 \b0}\cell';
    put '\row';
  end;
end;
```

```

/* Part 4 of RTF - Data section and row labels */
put '\trowd';
put "&cfmt\cellx2000";
if c1^='' then put "&cfmt\cellx3500";
put "&cfmt\cellx5000";

tmp='\qc{||strip(c0)||'}\cell';
put tmp;
if c1^='' then do;
  tmp='\qc{||strip(c1)||'}\cell';
  put tmp;
end;
tmp='\qc{||strip(c2)||'}\cell';
put tmp;
put '\row';

if p then do;
  put '}';
/* Part 5 of RTF - footnote */
put "\par\ql\fs18 &footnote";
put '\par {\*\Image Inserting Point;}; /* Preserved for image inserting
*/
  put '}'; /* end of RTF file */
end;
run;
%mend RTF_DEMO1;

%RTF_DEMO1(in=tabset, outfn=%str(C:\TEMP\PharmaSug 2019\example4.rtf),
  title=%str(ql\b Table 1. A 2 x 2 Contingency Table\b0),
  footnote=%str(\super 1 \nosupersub Fisher%'s Exact test.),
  hdr=%str(\qc{PharmaSug 2019}), ftr=%str(Demonstration));

```

## EMBEDDING IMAGE WITH TABLE

To better demonstrate or interpret output of statistical analysis, combining table and graph (bar/pie chart, histogram, or scatter plot etc.) is an efficient method to present data. RTF supports the displaying text, tables and images in the same document, and has a set of control words to insert/embed pictures.

There are two essential steps to include picture (e.g., image file) in the document. First, users should declare the starting of a picture/image, type of image file [e.g. \pict\pngblip (PNG file), \pict\jpegblip (JPEG file), or \pict\emfblip (EMF - enhanced metafile file)], scale of image (\picscalex $N$ , horizontal scaling:  $N\%$  of original image size, \picscaley $M$ , vertical scaling:  $M\%$  of original image size), and insert those control words at the point of image will be placed. The second step is to convert the binary image data to hexadecimal text and write this data stream after the picture control words declared in step 1. By the way, the location of image can be further aligned by adding \qc, \ql, or \qr commands.

The sample SAS macro code %READ\_IMG demonstrates the coding method to read binary data (INPUT ib informat) of an image file and convert the binary data stream to an image data block in hexadecimal format for RTF.

```

%macro READ_IMG(gfn=, gtyp=, gscl=);
  data _imgbin(keep=bt); /* read binary data */
    infile "&gfn" recfm=n;
    input byte ib1.;
    length bt $2.;

```

```

tm='0123456789ABCDEF';

if byte<0 then byte=byte+256; /* Convert binary data to hexadecimal */
bt=strip(substr(tm,int(byte/16)+1,1)||strip(substr(tm,mod(byte,16)+1,1));
run;

data _imghex(keep=hexdat); /* Output hexadecimal data for image embedding */
set _imgbin end=p;

length hexdat $80.;
retain hexdat;

if _n_=1 then do;
hexdat="\pard\par\qc{\pict\&gtyp.blip\picscalex&gscl\picscaley&gscl";
output; /* Output RTF picture control words and put image in the center */
end;

if mod(_n_, 40)=1 then hexdat='';
hexdat=strip(hexdat)||strip(bt);
if mod(_n_,40)=0 or p then output; /* Output data in Hexadecimal format */

if p then do;
hexdat='}\par\pard\par';
/* end of image data and add one empty line after the image */
output;
end;
run;
%mend READ_IMG;

%READ_IMG(gfn=%str(C:\TEMP\PharmaSug 2019\Histogram_Fisher.png),
gtyp=png, gscl=60);

```

The following SAS macro %RTF\_DEMO2 combines %RTF\_DEMO1 and %READ\_IMG into a new version. The displayed code segment shows the way to insert a processed image data block to the existing RTF file with an image insertion tag. In this sample code, it inserts a histogram generated by SAS PROC UNIVARIATE after the footnote section of the formatted table shown in the CONCEPT 5. Using this arrangement, we can archive the goal (Figure 1) successfully and generate a nice-looking table with graph.

```

%macro RTF_DEMO2(in=, outfn=, title=, footnote=, hdr=, ftr=,
outgf=, gftyp=, gfscl=, gfjst=);

-----
/* Same code displayed in %RTF_DEMO1 */
-----

%if %length(&outgf)>0 %then %do;
%read_img(gfn=&outgf, gtyp=&gftyp, gscl=&gfscl, gloc=&gfjst);

data rtf_part1 rtf_part2; /* RTF file splitting */
infile "&outfn" pad lrecl=1000;
input tx $ 1-1000;

retain pm;

if _n_=1 then pm=1;
if pm=1 then output rtf_part1; /* RTF file 1: before image data */
if index(tx,{'*\Image Inserting Point;'})>0 then pm=2;

```

```

        else if pm=2 then output rtf_part2; /* RTF file 2: after image data */
run;

data _null_;
    set rtf_part1 _imghex(in=imgx) rtf_part2;
    file "&outfn" lrecl = 1000;

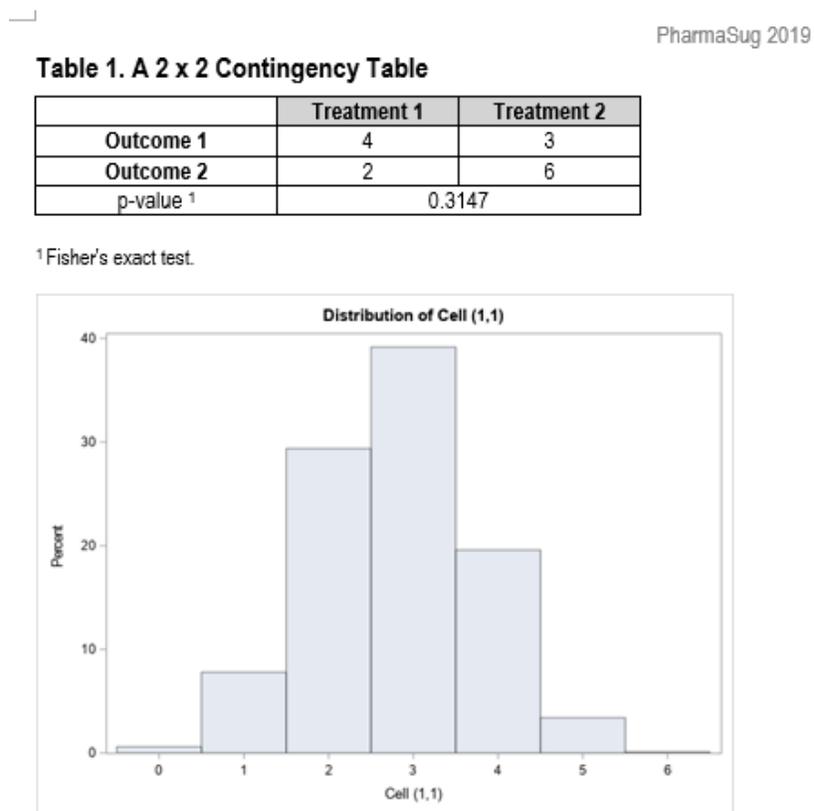
    if imgx then put hexdat;
    else put tx;
run;
%end;

%mend RTF_DEMO2;

%RTF_DEMO2(in=tabset, outfn=%str(C:\TEMP\PharmaSug 2019\example8.rtf),
    title=%str(q1\b Table 1. A 2 x 2 Contingency Table\b0),
    footnote=%str(\super 1 \nosupersub Fisher%'s Exact test.),
    hdr=%str(\qc{PharmaSug 2019}), ftr=%str(Demonstration),
    outgf=%str(C:\TEMP\PharmaSug 2019\Histogram_Fisher.png),
    gftyp=png, gfscl=60, gfjst=1);

```

The SAS macros %RTF\_DEMO2 listed here has only total 80 lines of SAS code; however, it is already powerful enough to generate formatted table with embedded image in RTF. Every experienced SAS users can add more user-defined parameters and RTF control words easily. These demonstrated macros will serve as basic code units to further develop a comprehensive application for more complicated tables with images inserted at different locations.



**Figure 1.** The result (screenshot) of RTF file containing table and embedded histogram.

## APPLICATION DEVELOPMENT

It is more realistic to generate tables in RTF from SAS datasets directly by using a general-purpose SAS macro (or package) than coding numerous SAS programs for different table types. Based on the sample SAS macro, %RTF\_DEMO2, the same concepts and logics can be used to develop a general-purpose macro for building different tables in RTF.

The coding flow of the proposed general-purpose macro for RTF table generation can be summarized in (1) global settings in page size, margins, font types, titles, footnotes, header and footer, and (2) line-/cell-specific settings for borders, highlight, cell merging. There are different strategies to program this macro. This section will only focus on the approach by using macro variables to pass those setting values.

Using following macro %RTF\_DEMO3 as an example, it collects basic settings, such as page orientation, margins, font type, title, footnote, header, footer, name of input SAS dataset and output RTF file name etc. through macro variables for processing. Those macro variable values will be further converted and output as standard RTF header, font table, color table and header, footer sections.

```
%macro RTF_DEMO3 (
  in=, /* name of input SAS dataset */
  outfn=, /* name of output RTF file */

  fontyp=%str(Times New Roman), /* font type */
  tfts=9, bfts=8, /* font sizes of title, table body */
  ffths=7, sfts=7, /* font sizes of footnote, source line */
  hdrfts=10, ftrfts=9, /* font sizes of header, footer */

  topmrg=1.0, btmrg=1.0, /* page: top margin and bottom margin */
  lfmrg=1.0, rtmrg=1.0, /* page: left margin and right margin */
  pgl=Portrait, /* page orientation */

  clwidth=, /* columns width per table column (%) */
  clnm=, /* variable name of each table column */
  psize=, /* rows per page */

  title=, footnote=, scr=, /* table: contents of title, footnote and source line
*/
  hdr=, ftr=, /* table: contents of header and footer */

  outgf=, gftyp=, /* name and type of graph/image file */
  gfjst=c, /* alignment of graph */
  gfsclx=100, gfscly=100, /* scale of image (%) */
  insertgf=1 /* point of insert: 1 (before), 2 (after) table */
);
```

In this %RTF\_DEMO3 macro, the part 4 described in the previous %RTF\_DEMO1/%RTF\_DEMO2 macros is replaced by a RTF syntax parser which reads and translates user-defined formats of each data cell to RTF control words without re-program the SAS code for different table styles. In this design, each table cell contains the original value plus a set of user-defined formatting commands (e.g., "cell data" @\_fmt[ cell format string]).

For example, a SAS dataset derived from demographic data through PROC FREQ and DATA step is the data source of a demographic analysis table. This raw dataset contains only numbers or converted text.

C1	C2	C3
143 (81%)	150 (79%)	0.6028
33 (19%)	40 (21%)	
88 (50%)	80 (42%)	0.1423
168 (95%)	178 (94%)	
8 (5%)	11 (6%)	0.7254
0 (0%)	1 (1%)	

Based on the mockup table, the raw dataset will be further modified to a "RTF-formatted" dataset for table generation.

C0	C1	C2	C3
{\enspace } \b Characteristic \b0 @_fmt[BT/jc]	\b Group 1 \line (N=176) \b0 @_fmt[BT/jc]	\b Group 2 \line (N=190) \b0 @_fmt[BT/jc]	\b p-value* \b0 @_fmt[BT/jc]
{\enspace } \b Gender \b0			
\tab Males @_fmt[/jl]	143 (81%) @_fmt[/jc]	150 (79%) @_fmt[/jc]	0.6028 @_fmt[/jc]
\tab Females @_fmt[/jl]	33 (19%) @_fmt[/jc]	40 (21%) @_fmt[/jc]	
{\enspace } \b Risk Factor \b0			
\tab Factor 1	88 (50%) @_fmt[/jc]	80 (42%) @_fmt[/jc]	0.1423 @_fmt[/jc]
{\enspace } \b Ethnicity \b0 @_fmt[/jl]			
\tab Non-Hispanic @_fmt[/jl]	168 (95%) @_fmt[/jc]	178 (94%) @_fmt[/jc]	
\tab Hispanic @_fmt[/jl]	8 (5%) @_fmt[/jc]	11 (6%) @_fmt[/jc]	0.7254 @_fmt[/jc]
\tab Subject Refused to Provide @_fmt[B/j]	0 (0%) @_fmt[/jc]	1 (1%) @_fmt[/jc]	

Using SAS INDEX and SUBSTR functions, the RTF format information of this "modified" dataset will be extracted and processed/interpreted by the parser and output the final RTF formatted data through SAS PUT statement. The syntax parser for each data row (SAS observation) provides the flexibility for further expansion to add more control words or functions (e.g., the command of cell merging).

The pseudo code of this RTF command parser is listed below.

"If commend string contains 'H' (highlight), program will add '\clcbpat1' in the beginning of the row control string."

"If commend string contains 'B' (bottom border), program will add '\clbrdrb\brds' control words to draw the border at the bottom of the cell."

"If commend string contains 'T' (top border), program will add '\clbrdr\brds' control words to draw the border at the top of the cell."

"If commend string contains 'R' (right border), program will add '\clbrdr\brds' control words to draw the border at the right side of the cell."

"If commend string contains 'L' (left border), program will add '\clbrdr\brds' control words to draw the border at the left side of the cell."

"If commend string contains '/jc', program will add '\qc' (centered) control word ahead of table data."

"If commend string contains '/jl', program will add '\ql' (align left) control word ahead of table data."

"If commend string contains '/jr', program will add '\qr' (align right) control word ahead of table data."

Furthermore, for "long" table/listing which will uses more than one page, to insert a \page control word at the point of the page-break will generate a new page.

The %RTF\_DEMO3 macro derived from previous %RTF\_DEMO2 with command parser will process formatted SAS dataset containing user-defined parameters and generate the final RTF (Figure 2 and 3) by using the macro call listed below.

```

%macro RTF_DEMO3(in=tabset, outfn=%str(C:\TEMP\PharmaSug 2019\example5.rtf));
fontyp=%str(Arial Narrow), tfts=9, bfts=8, ffts=7, sfts=7,
hdrfts=10, ftrfts=9,
topmrg=1.0, btmrg=1.0, lfmrg=1.0, rtmrg=1.0, pglt=Portrait,
clnm=%str(c0 c1 c2 c3), clwidth=%str(40 20 20 20), psize=10,
title= %str(\b Table 7. Baseline Demographics: Gender, Risk Factor and
Ethnicity \b0),
footnote=%str(*Fisher%'s exact test.),
scr=%str(C:\TEMP\PharmaSug 2019\Demol.sas (March 20, 2019 (12:16)));

```

The final well-formatted dataset for RTF (Table 7a) can be further modified by adding cell-highlighting and cell-merging commands to produce customized table (Table 7b) with different styles, or adding graph (Figure 2) according to the design criteria of different table mockups. The quality of those output files should be good enough to meet the requirements for thesis writing, presentation, publication, and regulatory submission.

(a)

**Table 7a. Baseline Demographics: Gender, Risk Factor and Ethnicity**

Characteristic	Group 1 (N=176)	Group 2 (N=190)	p-value*
<b>Gender</b>			
Males	143 (81%)	150 (79%)	0.6028
Females	33 (19%)	40 (21%)	
<b>Risk Factor</b>			
Factor 1	88 (50%)	80 (42%)	0.1423
<b>Ethnicity</b>			
Non-Hispanic	168 (95%)	178 (94%)	0.7254
Hispanic	8 (5%)	11 (6%)	
Subject Refused to Provide	0 (0%)	1 (1%)	

Page 1/1

Source: C:\TEMP\PharmaSug 2019\M3\_BaseDM2x.sas (March 20, 2019 (12:16))

\*Fisher's exact test.

(b)

**Table 7b. Baseline Demographics: Gender, Risk Factor and Ethnicity**

Characteristic	Group 1 (n=176)	Group 2 (n=190)	p-value*
	n (%)	n (%)	
<b>Gender</b>			
Males	143 (81%)	150 (79%)	0.6028
Females	33 (19%)	40 (21%)	
<b>Risk Factor</b>			
Factor 1	88 (50%)	80 (42%)	0.1423
<b>Ethnicity</b>			
Non-Hispanic	168 (95%)	178 (94%)	0.7254
Hispanic	8 (5%)	11 (6%)	
Subject Refused to Provide	0 (0%)	1 (1%)	

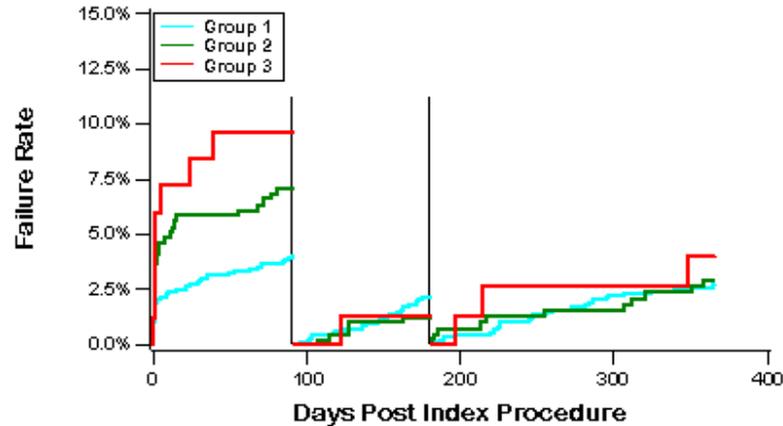
Page 1/1

Source: C:\Temp\PharmaSug 2019\Tab\_BaseDM.sas (February 19, 2019 (16:58))

\*Fisher's exact test.

**Table 7.** Sample tables in RTF generated by SAS macro %RTF\_DEMO3 with different format settings.

**Figure 1. Kaplan-Meier Curve Representing the Estimated Cumulative Incidence Rates of All Complications through 1 Year (Landmark Analysis)**



Time After Index Procedure (Days)						
	0	30	90	180	270	365
Group 1 # At Risk	1301	1247	1222	1215	1170	1053
# Event	14	38	52	27	19	32
Group 2 # At Risk	410	382	373	383	369	331
# Event	5	24	29	5	6	11
Group 3 # At Risk	83	75	74	76	74	64
# Event	1	7	8	1	2	3

Page 1/1

Source: C:\TEMP\PharmaSug 2019\KM\_dLMK.sas (March 18, 2019 (11:26))

Note: Only early terminated subjects are excluded from the landmark analysis.  
 Note: Includes only each subject's first occurrence of related events.

**Figure 2.** Screenshots of sample tables with graph in RTF generated by SAS macro %RTF\_DEMO3.

## CONCLUSIONS

RTF is capable to handle complex formatting task for a document containing text and image; it is a reasonable choice to generate well-formatted documents. The concepts and coding methods presented in this article should not be seen as a replacement to the SAS ODS RTF system; however, it is an alternative approach to generate customized output tables without using any ODS RTF syntax. Furthermore, the knowledge of RTF structure and skill of image embedding shared in this article should help experienced SAS users to better understand how to generate clean and concise RTF file from SAS datasets with images through DATA step. This approach will be a good starting point to further develop some light-weighted RTF add-on tools for SAS users. It should benefit users even without any experience in ODS RTF to customize their output tables for regulatory submissions, journal paper publications or conference presentations.

## REFERENCES

RTF Rich Text Format (RTF) Specification, version 1.5 ([http://www.biblioscape.com/rtf15\\_spec.htm](http://www.biblioscape.com/rtf15_spec.htm))

Haworth, Lauren. 2011. ODS RTF: the Basics and Beyond. SASGF 2012, Paper 263-2011.

## CONTACT INFORMATION

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

Name: Kai Koo  
Enterprise: Abbott Medical Devices  
Address: 3200 Lakeside Dr.  
City, State ZIP: Santa Clara, CA 95054  
E-mail: [kai.koo@abbott.com](mailto:kai.koo@abbott.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 registered trademarks or trademarks of their respective companies.