

## Tips and Tricks to Enhance RTF Output

Rohini Rao, Omeros Corporation, Seattle, WA  
Paul Hamilton, Omeros Corporation, Seattle, WA

### ABSTRACT

PROC REPORT has long been a powerful and flexible tool in clinical programming for producing tables and listings. The ODS TAGSETS.RTF destination was introduced in SAS® version 9.2, enabling programmers to produce higher quality output with less effort. This paper demonstrates a few tips and tricks to enhance the aesthetics of the reports.

Topics covered include:

- the use of decimal tabs to properly align numerical output
- the usage of Unicode to insert special characters
- the deletion of excess blank lines at the top and the bottom of a page
- the insertion of page-specific footnotes
- the repetition of a header row when the body of the section spans multiple pages.

SAS® version 9.2 on the Windows platform is used in this paper. It is targeted for an audience with an intermediate level of SAS and ODS knowledge.

### INTRODUCTION

In our paper, we look at techniques to embellish the appearance of our reports. We start with tables and listings that are functional and deliver accurate content and then add 'boutique' touches to enhance their appearance. These techniques allow us to have more control over our output and give us the luxury of crafting more user-friendly reports.

### DECIMAL TABS

Reports in the pharmaceutical industry contain mostly numerical output. Using decimal tabs allows us to properly align numerical output and improve readability. We first look at Table 1.1 created without decimal tabs and revisit it (Table 1.2) after adding the decimal tabs code.

Table 1.1 is a section of a sample vital signs table *without* decimal tabs:

**Table 1.1 Vital Signs**

	<b>Placebo (N=200)</b>	<b>Active (N=200)</b>
<b>Diastolic BP (mmHg)</b>		
Baseline		
N	200	200
Mean(SD)	73.8 (10.3)	74.0 (10.7)
Median	73.0	74.0
Min,Max	43, 107	49, 103
5 Minutes from Start of Procedure		
N	200	200
Mean(SD)	73.1 (10.3)	73.1 (10.3)
Median	72.0	75.0
Min,Max	43, 107	49, 103

Table 1.2 is the same table *with* decimal tabs:

**Table 1.2 Vital Signs**

	Placebo (N=200)	Active (N=200)
<b>Diastolic BP (mmHg)</b>		
Baseline		
N	200	200
Mean(SD)	73.8 (10.3)	74.0 (10.7)
Median	73.0	74.0
Min,Max	43, 107	49, 103
5 Minutes from Start of Procedure		
N	200	200
Mean(SD)	73.1 (10.3)	73.1 (10.3)
Median	72.0	75.0
Min,Max	43, 107	49, 103

SAS styles provide the ability to specify decimal tabs. However, they are not as flexible or attractive as what the native RTF codes allow. The RTF code is embedded within PROC REPORT (highlighted). Col0 is the vital signs category, Col1 is the placebo drug and Col2 is the active drug.

The required syntax to specify a decimal tab is: `pretext = "^R'\q1\tqdec\txNNN"`. The 'q1' code turns on left justification, the 'tqdec' code specifies a decimal tab and the 'txNNN' code determines where the tab is placed within the cell. *NNN* is a number supplied by the SAS programmer and the unit for this number is a twip. A twip is defined as 1/1440 of an inch and in our example, *NNN*=650 twips.

```

OPTIONS nodate nonumber nobyline orientation = landscape missing = ' ';
ODS escapechar='^';
ODS listing close;
ODS tagsets.omsrtf file = "&outpath.\&pgmout" pagePanels = none uniform;

PROC REPORT data = master missing nowindows split = '|' spanrows;
  columns pagenum panel col0 col1 col2;
  define pagenum / order noprint;
  define panel / order noprint;
  define col0/ display " "
    style(column) = [cellWidth = 3.5in
                    just = left];
  define col1/ display "Placebo|(N=&col1)"
    style(column) = [cellWidth = 1.25in
                    just = left
                    pretext = "^R'\q1\tqdec\tx650 '"];
  define col2/ display "Active|(N=&col2)"
    style(column) = [cellWidth = 1.25in
                    just = left
                    pretext = "^R'\q1\tqdec\tx650 '"];

  compute after panel;
    line@1 ' ';
  endcomp;
RUN;

ODS tagsets.omsrtf close;
ODS listing;

```

## INSERTION OF SPECIAL CHARACTERS

The usage of special characters in a report is a perennial problem. The SAS community has been able to produce these characters in the original RTF destination, but with more effort and difficult code. SAS<sup>®</sup> version 9.2 provides direct support of UNICODE characters.

Table 2.1 is a sample table from a clinical trial of a drug to prevent urinary incontinence. Here, we are using Unicode characters to display  $\geq 2$  as  $\geq 2$ . We also look at RTF code to display the superscript characters that appear in the table.

**Table 2.1: Supportive Analysis of Primary Endpoint**

	Placebo (N=200)	Active (N=200)
<b>Subjects with incontinent episodes per day</b>		
<2	43	73
$\geq 2$ to <5	41	30
...		
$\geq 10$	33	22
p-value <sup>a</sup>		0.0316
<sup>a</sup> Chi-square test		

The categories are displayed by embedding Unicode in our SAS code:

```
PROC FORMAT;
  value pan1stat
    0 - <2 = "<2"
    2 - <5 = "^{UNICODE 2265}2 to <5"
    ...
    10 - high = "^{UNICODE 2265}10";

  value pan2stat
    1 = "p-value ^R'\super ' a";
RUN;

DATA pan1B;
  set pan1;
  uiCat = put(uiNum,pan1stat.);
RUN;
```

The Unicode number to display the '≥' symbol is 2265. Defining the Unicode symbol in our PROC FORMAT will yield the results we want.

As well, in our example, the RTF code to display the footnote character 'a' in superscript font is included in the pan2stat format as follows: ^R'\super ' a.

## DELETION OF EXTRA BLANK LINES AND PAGE SPECIFIC FOOTNOTES

Table 3.1 in our example here is the demographics table from the same clinical trial. Col0 is the demographic category, Col1 is the placebo drug, Col2 is the clinical trial drug and Col3 is the total. The categories, or the 'panels' of data are separated by one blank line. One method of inserting a blank line after a panel of data is shown below:

```

OPTIONS nodate nonumber nobyline orientation = landscape missing = ' ';
ODS escapechar='^';

ODS listing close;
ODS tagsets.omsrtf file = "&outpath.\&pgmout" pagePanels = none uniform;

PROC REPORT data = master missing nowindows split = '|';
  columns pagenum panel col0 col1 col2 col3;
  define pagenum / order noprint;
  define panel / order noprint;
  define col0/ display " "
                style(column = [cellWidth = 3in
                               just = left]);
  define col1/ display "Placebo|(N=&col1)|n(%) "
                style(column) = [cellWidth = 1.3in
                               just = left
                               pretext = "^R'\ql\tqdec\tx650 '"];
  define col2/ display "OMS302|(N=&col2)|n(%) "
                style(column) = [cellWidth = 1.3in
                               just = left
                               pretext = "^R'\ql\tqdec\tx650 '"];
  define col3/ display "Total|(N=&col3)|n(%) "
                style(column) = [cellWidth = 1.3in
                               just = left
                               pretext = "^R'\ql\tqdec\tx650 '"];

  COMPUTE after panel;
    line @ 1 ' ';
  ENDCOMP;
RUN;
ODS tagsets.omsrtf close;
ODS listing;

```

The above method inserts a blank line after the last panel on the page as well, which is not wrong but may be unnecessary. In order to make the table more aesthetically pleasing, we can omit the blank line at the bottom of each page by using the \$varying format and the panel numbers.

```

COMPUTE after panel;
  msg=" ";
  if panel in (4,7) then len=0;
  else len=10;
  line @1 msg $varying10. len;
ENDCOMP;

```

In this report, panels four and seven are the last panels on the page. We suppress the output of the blank line after these panels by defining a length of zero to len. The length of ten assigned to the visible blank lines is arbitrary.

This logic can be applied to produce page specific footnotes as well.

```

COMPUTE after pagenum;
  msg2="This footnote will appear only on page 2";
  if pagenum=2 then len=150;
  else len= 0;
  line @1 msg2 $varying150. len;
ENDCOMP;
    
```

Table 3.1 has no extra blank line at the bottom of page one and a footnote appearing only on page two.

**Table 3.1: Demographics and Subject Characteristics**

	<b>Placebo (N=200)</b>	<b>Active (N=200)</b>	<b>Total (N=400)</b>
<b>Age (year)</b>			
N	200	200	400
Mean(Std)	66.7	67.0 (9.6)	69.4 (9.7)
Median	69	69	69
Min,Max	32.0, 89.0	31.0, 88.0	31.0, 89.0
<b>Gender</b>			
Male	22	27	49
Female	178	173	351
<b>Ethnicity</b>			
Hispanic Or Latino	33	17	50
Not Hispanic Or Latino	167	183	350
<b>Race</b>			
American Indian or Alaska Native	2	3	5
Asian	17	14	31
Black or African American	21	22	43
Native Hawaiian or Other Pacific Islander	0	0	0
White	159	161	320
Other	1	0	1

**Page 2...**

	<b>Placebo (N=200)</b>	<b>Active (N=200)</b>	<b>Total (N=400)</b>
<b>Baseline number of incontinent episodes per day</b>			
>2	36	36	72
≥2 to <5	105	100	205
... ≥10	7	7	14
<b>This footnote will appear only on page 2</b>			

## REPETITION OF A HEADER ROW WHEN DATA SPAN MULTIPLE PAGES

An adverse event table often has data displayed for each Preferred Term under a System Organ Classification. Some SOC groups have several Preferred Terms, all of which do not fit on one page. In such cases, where the Preferred Terms span multiple pages, it is useful to have the SOC name repeat on the first row of each page. Here, we look at subject incidence of treatment emergent adverse events.

In table 4.1, the Preferred Terms for Renal and Urinary Disorders are on pages one and two. The SOC name repeats on the top of page two.

**Table 4.1: Subject Incidence of Treatment-Emergent Adverse Events by System Organ Class and Preferred Term (Page 1)**

	<b>Vehicle (N=50)</b>	<b>Active (N=50)</b>
<b>Any Event</b>	47	36
<b>Renal and Urinary Disorders</b>	3	1
Allergic Cystitis	2	1
Anuria	2	1
Bacterial pyelonephritis	7	5
Bladder candidiasis	9	5
Bladder Spasm	3	1
Calculus urinary	1	1
Dysuria	3	0
Fluid retention	12	9
Fungal cystitis	3	1
Haematuria	4	2
Haemorrhage, urinary tract	1	0
Nephrolithiasis	7	2
Pyrexia	7	6

(Page 2)

	<b>Vehicle (N=50)</b>	<b>Active (N=50)</b>
<b><i>Renal and Urinary Disorders (continued)</i></b>		
Renal haematoma	1	0
Renal impairment	5	2
Renal syphilis	3	1
Renal tuberculosis	2	1
Strangury	1	0
Trigonitis	1	0
Urethral pain	2	4
Urinary retention	15	11
Urine flow decreased	7	3

In our master data set, we have defined page, panel and row numbers. Each page contains one or more panels. Each SOC is a separate panel. The SOC title is the first line in the panel and is always assigned row=0. When we come to the top of a new page and the row is not equal to 0, we know that the SOC panel from the previous page is continuing to the current page.

```

DATA master2;
  set master;
  by pagenum panel row;
  *header;
  if first.pagenum and row ne 0 then do;
    row=0;
    col0="^R'\ql\b ' " || strip(propcase(aebodsys)) || "^R'\b0\i 'continued)";
    output;
  end;
  *body;
  set master;
  output;
RUN;

PROC SORT data=master2;
  by panel row;
RUN;

OPTIONS nodate nonumber nobyline orientation = landscape missing = ' ';
ODS escapechar='^';
ODS listing close;
ODS tagsets.omsrtf file = "&outpath.\&pgmout" pagePanels = none uniform;

PROC REPORT data = master2 missing nowindows split = '|';
  columns panel row col0 col1 col2;
  define panel / order noprint;
  define row / order noprint;
  define col0/ display " "
    style(column) = [cellWidth = 3.5in
                     just = left];
  define col1/ display "Vehicle|(N=&col1)"
    style(column) = [cellWidth = 1.25in
                     just = left
                     pretext = "^R'\ql\tqdec\tx650 '"];
  define col2/ display "Active|(N=&col2)"
    style(column) = [cellWidth = 1.25in
                     just = left
                     pretext = "^R'\ql\tqdec\tx650 '"];
RUN;

ODS tagsets.omsrtf close;
ODS listing;

```

## CONCLUSION

We have compiled a list of boutique programming techniques that may be used to enhance the appearance of clinical data reports. We have introduced decimal tabbing with in-line styling to vertically align numerical output. As a result, the user can easily recognize the varying magnitude of numbers within a cell. We have demonstrated the ease of UNICODE usage in inserting special characters in a report. The \$VARYING format can be applied in the LINE statement to not only remove excess blank lines, but also to insert page specific footnotes. In our opinion, the latter method is a simple yet elegant usage of the \$VARYING format. Last but not least, we have shown how to repeat a header row at the top of relevant pages, when a block of data spans multiple pages. The intent here is to avoid flipping pages back and forth while reading the report. This list is by no means intended to be comprehensive. It is merely a discussion of a few methods of enhancing our output or 'gilding our lilies'.

## APPENDIX

The code that creates tagsets.omsrft is below:

```
PROC TEMPLATE;
  define tagset Tagsets.Omsrft;
    notes "This is the RTF_SAMPLE tagset which is inherited from rtf.tpl";
    define event realraw;
      start:
        put value /if value;
      end;
    default_style = "styles.rtf";
    parent = tagsets.rtf;
    uniform;
  end;
RUN;
```

## REFERENCES

Hamilton, Paul. "ODS to RTF: Tips and Tricks". Available at:

<http://www2.sas.com/proceedings/sugi28/024-28.pdf>

Rashleigh-Berry, Roland. "Roland's SAS Macros". Available at:

<http://www.datasavantconsulting.com/roland/condline.html>.

## CONTACT INFORMATION

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

Name: Rohini Rao and Paul Hamilton  
Enterprise: Omeros Corporation  
Address: 201 Elliott Ave W, Suite 500  
City, State ZIP: Seattle, WA 98119  
sWork Phone: (206) 676-5000  
E-mail: [rrao@omeros.com](mailto:rrao@omeros.com); [phamilton@omeros.com](mailto:phamilton@omeros.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.

The content of this paper does not include actual clinical trial data. The opinions expressed herein are those of the authors and not necessarily those of Omeros Corporation.