PhUSE 2008

Paper CC06

Styling statistical output
– a CRO perspective –

Annet te Velde, AUTHOR! Data Analysis and Reporting, Soest, The Netherlands

ABSTRACT

This paper describes the use of the SAS® Output Delivery System (ODS) in the daily practice of a CRO specialized in data analysis and medical writing. Because of our focus on these two disciplines and our endpoint being a clinical document in MS WORD®, ODS plays a major role in producing statistical output. Clinical documents are often based on Sponsor's word processor templates and style guidelines. We strive to create statistical output for a clinical document, which adheres to these style guidelines. This, as a result, saves the medical writer time and diminishes the chance of erroneous changes to the contents of a table or listing due to formatting. An introduction to the basics of ODS will be presented with focus on ODS RTF. The creation of templates for different Sponsors, by using PROC TEMPLATE, will be explained, including examples. Enhancements to the output will be described and some tips and tricks are presented.

INTRODUCTION

This paper introduces the ODS RTF basics in the context of providing medical writers statistical output styled according to word processor style templates of different companies. In addition, it gives a basis for creation of standard global style templates. The ODS system can be quite overwhelming for a junior SAS programmer, hopefully with this paper, programmers can make a start to discover the possibilities of the SAS ODS system.

WHY USE ODS RTF

SAS programmers working in close collaboration with medical writers, most probably know from experience that inclusion of statistical output into the core text of a clinical document is often very time-consuming for medical writers. This can have a major impact on the timelines and costs, especially when it concerns a substantial number of tables and listings that need to be included, such as in a summary of clinical safety or a summary of clinical efficacy of the common technical document (CTD). These documents include the complete package of trials performed for registration/submission of an investigational product. Medical writers copy or literally write out and take over each number from the original tables and/or listings into the core text of their document. Furthermore, they have to create and design a table in MS WORD according to Sponsor-defined style guides.

The RTF-destination of the SAS® ODS system makes it possible to produce output that can be incorporated (almost) seamlessly into word processors, such as MS WORD. At AUTHOR!, rich text format (RTF) output is generated so that the medical writer can include the RTF output directly in the core text as an in-text table. When a CRO is requested to perform both data analysis and medical writing for the same project, one is therefore able to perform these tasks more efficiently and as a result save time and money. The creation of RTF-tables can already be prepared with the SAS system, and the layout of tables can be made more appealing and word-processor-friendly during the creation process, right away. If creating RTF statistical output during data analysis of one project already saves time and effort, imagine what it would mean if a CRO is requested to perform more projects for the same Sponsor. Copying of SAS code including style definitions will not be necessary since ODS style templates can be made and archived for future projects.

The ODS system was first introduced in SAS Version 8.1, has developed since then and has proven to be still very useful and, in particular, cost-efficient when data analysis and medical writing services are provided within the same company. Furthermore, details included in an RTF-formatted table or listing and copied into the core text of a report, do not need to be thoroughly QC'd, since the data itself was not written out or typed over from the original SAS output, instead the complete table, including styles, can be copied in.

BASICS OF ODS

Statistical output not created within the ODS system was designed for a traditional line-printer and limited to mono-spaced fonts, like Courier. An example is presented below.

Systolic blood pressure values

PROTOCOL	SUBJECT	PARAM	VALUE
PHUSE2008	101	SBP	123
PHUSE2008	102	SBP	144
PHUSE2008	103	SBP	101
PHUSE2008	104	SBP	128

Just a few simple ODS commands will immediately enhance the looks of your output. First, you need to call to ODS to request RTF OUTPUT and to direct this output to an RTF file:

```
ods rtf file = "ods_1.rtf";
```

The output from every procedure or data step following this statement is now directed to the 'ods_1.rtf' file. To turn the RTF destination off so that the file will be finished and closed you need to add the following statement after you have run all the necessary procedures to be displayed in the RTF file:

```
ods rtf close;
```

Note that in the above example the output is also directed to the output window. If you want only RTF output to be created, add the following statement before the first RTF statement:

```
ods listing close;
```

This turns the listing ODS output destination off. Vice versa, if you want the output to be created in the output window again, the following syntax is to be submitted:

```
ods listing;
```

In conclusion, with the following code,

```
ods rtf file = "ods_1.rtf";

... title and footnote statements...
proc print ..;
...statements ...
run;

ods rtf close;
```

the same listing as presented above can be displayed as below:

Systolic blood pressure values

PROTOCOL	SUBJECT	PARAM	VALUE
PHUSE2008	101	SBP	123
PHUSE2008	102	SBP	144
PHUSE2008	103	SBP	101
PHUSE2008	104	SBP	128

This is just an example of how the output might look, since this depends on your default ODS style template. Within the ODS system it is possible to adjust the layout and style as much as you would like, e.g. fonts, background colors, style of individual cells, style of titles, etc.

ENHANCING YOUR OUTPUT CREATED WITH ODS RTF

To avoid editing statistical output in a word processor and to improve appearance and at the same time create all your statistical output with the same layout and styles, you can create your own style template, including your standard styles. The styles used in the above example are based on the default style template of our SAS system. Style definitions (i.e. style templates) are stored in so-called template stores. Using the ODS PATH statement, you can tell SAS which locations to use for reading from and writing to template stores. For example the following code:

```
libname styles "c:\Clientstyles";
ods path styles.Client1(write) sashelp.tmplmst(read);
```

takes care that the style definitions are created in template store styles.Client1 on directory c:\Clientstyles and that styles used by ODS and PROC TEMPLATE are first looked up in template store styles.Client1 and, if not found, in the template store sashelp.tmplmst, that has been shipped with SAS.

A client style template is to be created with the SAS procedure PROC TEMPLATE. PROC TEMPLATE is used, amongst other things, to list, create, and modify ODS styles. Default style elements and attributes can be redefined by customizing fonts, colors, borders, etc. of existing style templates to suit a Sponsor's template style. Thus, instead of creating a complete new style template by defining hundreds lines of code, it is advised to use existing style templates and make the necessary changes according to the styles and formats requested by your client. To be able to decide what style template will be the best basis for your new client style template, you can view an existing style by submitting the following SAS code:

```
proc template;
   source styles.RTF;
run;
```

The output presents the contents of style template RTF. This style template includes many lines of code and over 100 style elements. A set of style elements makes up a style definition. There are many style elements (please check reference 8 for more information). A few examples of style elements that can be of relevance for RTF output are presented below:

Element	Description
Fonts	Establishes a list of fonts.
Color_list	Establishes a list of color names and their RGB values.
Colors	Associates parts of SAS output with colors from color_list.
Text	Provides text for specific parts of the output.
Cell	Abstract: controls table cells.
Header	Renders the headers of a table.

Style elements include many different style attributes, e.g. the style element 'Header" can consist of the below mentioned attributes font_face, font_weight, font_size. A list of some of the attributes can be found below.

Attribute	Description
Font_Face	font face; specifies the font name
Font_Size	font size
Font_Style	font style (e.g. italic)
Font_Weight	font weight (e.g. bold)
Justify	Justification; specifies left, right, or center.
OutputWidth	width of the table
Cellwidth	Table cell width
CellHeight	Table cell height. A row will have the height of the highest cell.
CellPadding	amount of white space on all four sides of text in a cell
PreImage	Specify an image to be place before the table or cell.
PostImage	Specify an image to be place after the table or cell
PreText	Specify text to place before the table or cell
PostText	Specify text to place after the table or cell
LeftMargin	width of left margin
RightMargin	width of right margin
BottomMargin	width of bottom margin
TopMargin	width of top margin

A list of all predefined style elements and attributes provided by SAS together with their possible values can be found in “the Complete Guide to the SAS Output Delivery System, PROC TEMPLATE, Syntax, Define Style”.

The PARENT statement within PROC TEMPLATE allows us to create a new style based on an existing style. In addition, the REPLACE statement lists the style elements to be adapted. For example, we can adapt the existing style template ‘RTF’ to a new client-based style as follows:

```
proc template;
   define style styles.AUTHOR_clientname1;
   parent = styles.RTF;
   replace –style element-;
   end;
run;
```

An example of a style element is “fonts”. This style element affects the way the fonts are presented. In this example, the style element ‘fonts’ is adapted; each attribute includes three characteristics, separated by commas; i.e. typeface, font size and font weight.

```
proc template;
   define style styles.AUTHOR_clientname1;
   parent = styles.RTF;
      replace fonts /
          'TitleFont' = ("Arial",11pt,Bold Italic)
                   /* Titles from TITLE statements */
          'TitleFont2' = ("Arial",10pt,Bold Italic)
                   /* SAS Procedure titles */
          'headingFont' = ("Arial",10pt,Bold)
                   /* Table column and row headings */
          'docFont' = ("Arial",9pt)
                   /* Data in table cells */
          'footFont' = ("Arial",8pt)
                   /* Footnotes from FOOTNOTE statements */
   end;
run;
```

Another example is to change the output structure, i.e., the looks of a table. ‘Table from Output’ is a style element that controls rules, borders, and spacing for all tables:

```
proc template;
   define style styles. AUTHOR_clientname1;
   parent = styles.RTF;
      replace Table from Output /
         frame = hsides
             /* outside borders: void, box, above/below, vsides/hsides, lhs/rhs */
         rules = groups
             /* internal borders: none, all, cols, rows, groups */
         cellpadding = 5pt
             /* the space between table cell contents and the cell border */
         cellspacing = 0pt
             /* the space between table cells, allows background to show */
         borderwidth = .8pt
             /* the width of the borders and rules */;
   end;
run;
```

And what about changing margins; ‘Body from document’ is a style element that controls the margins of the RTF file:

```
proc template;
   define style styles. AUTHOR_clientname1;
   parent = styles.RTF;
   replace body from document /
         leftmargin=2.5in
```

```
         rightmargin=2.5in
         topmargin=2.5in
         bottommargin=2.5in;
   end;
run;
```

In conclusion, a style definition (style template) contains a set of style elements and a style element is a collection of style attributes including their values, which can be applied, to a specific part of the output.

CREATION OF CLIENT STYLE TEMPLATE

Now you've seen a few basics of ODS, let's create a specific client style template based on a MS WORD template and guideline provided by one of our clients and apply the style template to statistical output. The styles in the MS WORD template are as follows for A4 portrait in-text tables:

- Margins: Top 1 inch, Bottom 1.69 inch, left 1 inch, right 0.77 inch
- Font: Times Roman, black, 10pt,
- Table header font: Times Roman, black, 10 pt, bold
- Table header background gray
- Width of borders and rules of table 2pt
- Links in blue font

The following code can be submitted to create a style template adhering to these definitions:

```
/* assign libname and ods path */
libname styles "C:\Clientstyles";
ods path styles.Client1(write) sashelp.tmplmst(read);

/* create a new client template based on the above definitions */
proc template;
   define style styles.AUTHOR_PHUSE2008;
   parent = styles.RTF;
   replace fonts /
      'TitleFont' = ("Times Roman",10pt,Bold)
      'headingFont' = ("Times Roman",10pt,Bold)
      'docFont' = ("Times Roman",10pt)
      'footFont' = ("Times Roman",10pt);
   replace color_list /
      'link' = blue
      'bgH' = grayBB
      'fg' = black
      'bg' = white;
   replace Body from Document /
      bottommargin = 1.69in
      topmargin = 1in
      rightmargin = 0.77in
      leftmargin = 1in;
   replace Table from Output /
      frame = box
      rules = all
      borderwidth = 2pt;
end;
run;

/* define global options */
options nodate nonumber;

/* remove procedure titles */
ods noptitle;

/* define name of rtf file in which the statistical output is stored */
ods rtf file='C:\AUTHOR\ANALYSIS\PHUSE presentation 2008\DATA\Table1_1.rtf'
style=AUTHOR_PHUSE2008 bodytitle;
```

```
/* define title and footnotes for statistical output */
title 'Systolic and diastolic blood pressure values per treatment group';
footnote 'PHUSE 2008 - example client style template - Oct 2008';

/* create summary table */
proc report data=xxxx headline headskip nowindows;
column –statements- ;
define –statements- ;
define –statements- ;
etc.
run;

/* turn of the ods rtf destination */
ods rtf close;
```

This results in the following table layout:

Systolic and diastolic blood pressure values per treatment group

		Treatment A	Treatment B	Treatment C
Parameter	**Statistic**			
Systolic blood pressure	N	3	3	2
	Mean	123	138	130
	SD	22	20	14
	Minimum	101	123	120
	Maximum	144	160	140
Diastolic blood pressure	N	3	3	2
	Mean	79	89	78
	SD	6	10	3
	Minimum	73	78	76
	Maximum	84	97	80

PHUSE 2008 - example client style template - Oct 2008

To be able for all users within your company to access the style definitions, you should assign a libref to the directory that contains the style definition and putting an appropriate ODS PATH statement in the AUTOEXEC.SAS file of all these users. In addition, you can add a central macro that calls ODS RTF with the correct style definition.

TIPS AND TRICKS

TITLES AND FOOTNOTES

ODS creates RTF files where your specified titles and footnotes are automatically placed as page headers and footers in your word processor. When combining separate RTF files, your page headers and footers might get lost, because it is replaced by the consecutive RTF file. To overcome this problem, you need to include section breaks and tick off the ‘same as previous’ option in your header, before including the next RTF file into your word processor. Another solution is to place the titles and footnotes not into page header or footer by adding the option BODYTITLE as follows:

```
ods rtf file = “ods_1.rtf” bodytitle;
```

Now, the titles and footnotes are part of the main document. Please note that when the output is longer than one page, the title will only be put on the first page and the footer only on the last page.

DECIMAL ALIGNMENT

It is possible to further customize the output by inserting RTF tags. For example, in version 9, SAS added support for decimal alignment in table columns. In version 8 RTF tags can be used to accomplish the same thing. See reference 7 for more information.

PAGE X OF Y

In version 8, you can add raw RTF code in the TITLE or FOOTNOTE statement to produce the page X of Y numbering in your word processor:

```
ods rtf file="table2.rtf";
proc print data=xxx;
footnote j=r
"{\field{\*\fldinst{\b\i PAGE}}}\~{\b\i of}\~{\field{\*\fldinst{\b\i NUMPAGES}}}";
run;
ods rtf close;
```

In SAS 9 and later, the experimental in-line formatting function {PAGEOF}, produces page X of Y numbering with WORD 2000 and higher.

```
ods escapechar="^";
ods rtf file="table2.rtf";
proc print data=xxx;
footnote j=r "^{pageof}";
run;
ods rtf close;
```

PAGE BREAKS

Normally, after each procedure output a page break is added by ODS. These page breaks can be removed (for example if you use small tables that can easily be placed on one page) by applying the following code:

```
ods rtf startpage=never;
```

This code should be placed between the two procedure calls.

TABLE OF CONTENTS

The following example creates a table of contents page that contains embedded table of contents data for some procedures but not for others. The insertion of the table of contents data can be turned on and off in the middle of a program.

```
ods rtf toc_data; /* turn toc on */
ods rtf notoc_data; /* turn toc off */
```

NICE TO KNOW

DEFAULT STYLE TEMPLATE

When a style template is not specified the default style template ‘Default’ is used. By default, ODS RTF uses the RTF style definition, i.e. the Styles.RTF that is in the template store Sashelp.Tmplmst.

HEIGHT AND WIDTH

The number of columns fitting on a page is taken care of by the ODS system. A standard page width is assumed and if a table will not fit in the page width, then ODS will break the table up into smaller tables. The number of rows is not taken care of by ODS but by the word processor, since for example you might want to add headers or footers which influence the height of a table and the place where a table should break onto the next page. This should be kept flexible and not fixed during table creation.

SAS AND MS WORD COMPATIBILITY

RTF files created by ODS in SAS Release 8.1 are readable by MS WORD 97 and later. When created in SAS Release 8.2 and SAS 9, files are readable in MS WORD 2000 and later. When created in SAS 9.1 and later, RTF files are readable in MS WORD 2002 and later.

CHANGES IN VERSION 9.X COMPARED TO 8.X

Over time, PROC TEMPLATE styles have remained almost unchanged. Two enhancements were made with regard to ODS RTF; 1) In version 9.x it is possible to automatically produce a Table of Contents by default and 2) The Minimal style has slightly changed such that the font is now automatically set to SAS Monospace font (for Windows)

and not the default WORD font. For details, see reference 5. Note, that the REPLACE option is not used anymore in SAS Version 9.2. SAS automatically includes the complete style definition by using the PARENT statement and changes to the style template can be made without the REPLACE option, by simply leaving it out (for more detail, see reference 9).

CONCLUSION

Creation of ODS RTF statistical output can be quite complex and overwhelming. This paper is an introduction to ODS RTF and shows that even junior SAS programmers can quite easily create a dictionary of specific client style templates for statistical output to be included in a word processor. This can especially for a CRO, be a cost- and time-saving measure. There are some possibilities and examples mentioned in this paper, however many are left untouched. For further exploration, please have a look at the reference list.

REFERENCES

1. Creating RTF-Output Using ODS and PROC TEMPLATE. Blue Gum Data Analysis, Blackheath, NSW.; SUGA 2001, Melbourne.
2. Output Delivery System, The basics. Lauren E. Haworth, 2001.
3. Guide to the Output Delivery System in SAS Online Doc, version 8, 1999, SAS Institute Inc., Cary, NC, USA.
4. Knowledge based/Samples and SAS notes http://support.sas.com/kb/?nh=100&qm=3&la=en&ct=51012&qt=subject:%22rtf_1%22+
5. A soft landing to ODS 9.1 http://support.sas.com/rnd/base/ods/templateFAQ/ODS91.pdf#pagemode=bookmarks
6. Knowledge based/ Focus areas; SAS Notes and Concepts for ODS; http://support.sas.com/rnd/base/ods/templateFAQ/index.html
7. SAS® with Style: Creating your own ODS Style Template for RTF Output (SUGI 29, Paper 125-29), Lauren Haworth, Genentech, Inc., South San Francisco, CA,; http://www2.sas.com/proceedings/sugi29/125-29.pdf
8. http://support.sas.com/rnd/base/ods/templateFAQ/rtfexplain.html
9. The TEMPLATE Procedure Styles: Evolution and Revolution. Kevin D. Smith, SAS Institute Inc., Cary, NC. SUGI 31, paper 053-31.

RECOMMENDED READING

Output Delivery System, The basics. Lauren E. Haworth, 2001.

CONTACT INFORMATION

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

Annet te Velde
AUTHOR! Data Analysis and Reporting
Vondellaan 32
3768HS Soest
The Netherlands
Work Phone: +31 (0)35 5882814
Fax: +31 (0)35 5882813
Email: atevelde@author.nl
Web: www.author.nl