Paper ####

Advanced RTF layout with SAS

Katja Glaß, Bayer Schering Pharma AG, Berlin, GER

ABSTRACT

The Output Delivery System (ODS) from SAS 9 with ODS RTF offers a lot of possibilities to design rich text format (RTF) documents. With additional included RTF commands the layout possibility for designing RTF is nearly unlimited. The purpose of this paper is to point out the different ways of styling RTF with SAS. The focus is put on the advantages of direct RTF commands to create for example auto numbering and a different kind of table of contents. Furthermore, general methods are described how new RTF commands can be found to enrich the output.

INTRODUCTION

The main advantage of the Output Delivery System from SAS is the possibility to create logical tables which are organized in cells. Those tables are much easier to handle, to format and also to read.
The preferred format for clinical study documents is the MS Word format. To create formatable and adjustable tables that fit the current layout requirements, SAS offers ODS RTF which has many other advantages compared to the historically listing (ASCII) format. The design and additional ways to style the RTF file offered by SAS will be introduced in this paper.

BASICS

FUNDAMENT

The most important options for using ODS RTF are the site and paper settings. Those options should be set at the beginning of each SAS session to be taken into account for the RTF creation. For table creation in the European area usually the A4 format is used together with a landscape orientation:

```
OPTIONS PAPERSIZE=A4 ORIENTATION=landscape;
```

BASIS

To use the ODS RTF channel only two simple commands are needed, one for starting and the other for ending:

```
ODS RTF;
ODS RTF CLOSE;
```

All SAS outputs created between those commands are automatically placed within the RTF file. The structure and content is the same as the corresponding listing output. A very basic RTF document can be created in this way:

```
ODS RTF    FILE="C:\temp\class.rtf"
           AUTHOR="Katja Glaß"
           TITLE ="Output of SASHELP.CLASS";
PROC PRINT DATA=sashelp.class; RUN;
ODS RTF CLOSE;
```

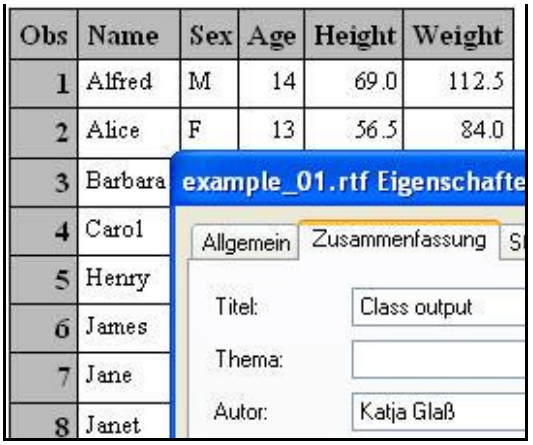

Figure 1: Example for simple RTF output

The "FILE" command redirects the output into a different file then the SAS default. Some document metadata information can be written to the RTF file by using the *AUTHOR* and *TITLE* option of ODS RTF.

A very useful option is *BODYTITLE*. With this the titles and footnotes does not appear in the header and footer of the document, but in the body itself. When new tables should always start on a new page, *STARTPAGE=YES* can be used. The most important ODS RTF option is the style. The general layout, margins , colors and much more can be defined. A complete different layout can be produced by using a different SAS style:

```
ODS RTF    FILE="C:\temp\class.rtf"
BODYTITLE STARPAGE=YES
STYLE=journal;
TITLE1 "Printout of sashelp.class";
FOOTNOTE1 "Note: Created on February 2007";
PROC PRINT DATA=sashelp.class; RUN;
ODS RTF CLOSE;
```

Basics about the ODS RTF styles can be found in many documents, for example in the SUGI paper 125-29 [1].

ODS ESCAPECHAR

To be able to use field functions and other RTF elements, a special character is needed. This prefix has to be registered in SAS, so that the followed code is handled special. To define this special character, the ESCAPECHAR option is used:

```
ODS ESCAPECHAR='^';
```

STYLING WITH ODS RTF

GENERAL LAYOUT OPTIONS

For titles and footnotes some basic styling possibilities are given. For example bold fonts, colors, background colors, font sizes and heights can be used. Additionally justification, underlining and links are available [2]. The following example demonstrate the possibilities:

```
ODS RTF FILE="C:\temp\example.rtf";
TITLE1 BOLD COLOR=blue BCOLOR=yellow "bold blue title on yellow background";
TITLE2 FONT=arial HEIGHT=20 UNDERLIN=1 "tall underlined arial title";
TITLE3 JUSTIFY=R LINK="http://www.sas.com" "left aligned SAS link title";
PROC PRINT DATA=sashelp.class;RUN;
ODS RTF CLOSE;
```

Some standard layout possibilities exists for tables as well. By using mainly PROC REPORT for any kind of table output, only this procedure will be in focus within this paper. Some options are also available for other procedures. The layout options can be applied to different regions of the report. The most used regions are the header,

columns, rows and the complete report. Different options like colors, fonts, alignments and margins can be adapted for those areas. A complete listing can be found in the SAS documentation [3].
Here is a small example to demonstrates miscellaneous possibilities:

```
ODS RTF FILE="C:\temp\example.rtf";
PROC REPORT DATA=sashelp.shoes NOWINDOWS HEADLINE MISSING
           style(header)=[     just=center font_size=10pt
                               font_weight=bold background=white]
                               style(report)=[cellpadding=6pt];
    column region subsidiary product stores sales inventory returns;
    DEFINE region     / display;
    DEFINE stores     / display;
    DEFINE inventory  / display STYLE={FONT_STYLE=italic};
    DEFINE returns    / display STYLE={CELLWIDTH=3cm};
    compute region;
        CALL DEFINE(_col_, "Style", "STYLE=[indent=1cm]");
    endcomp;
    compute stores;
        IF stores < 4
        THEN DO;
            CALL DEFINE(1, "Style", "STYLE=[background=yellow]");
        END;
    endcomp;
RUN;
ODS RTF CLOSE;
```

EMBEDDED LAYOUT OPTIONS

The presented style elements from the former section can also be used in any other area with the use of the escapechar plus an “S” as indicator. In titles and footnotes those elements can be placed as showed in the following example:

```
ODS RTF FILE = "C:\temp\test.rtf" bodytitle;
title1 "standard title";
title2 "^S={indent=.12in} indented title";
title3 "^S={just=right} right aligned title";
title4 "^S={font_size=40pt} tall title";
title5 "^S={font=('ARIAL',20pt,bold)} tall, bold and arial title";
footnote1 "^S={preimage='H:\My Documents\My Pictures\fische-7.jpg'} a picture";
PROC PRINT DATA=sashelp.class;RUN;
ODS RTF CLOSE;
```

Accordingly, those elements can be used by ODS RTF TEXT:

```
ODS RTF FILE="C:\temp\test.rtf" bodytitle;
ODS RTF TEXT = "standard title";
ODS RTF TEXT = "^S={indent=.12in} indented title";
ODS RTF TEXT = "^S={just=right} right aligned title";
ODS RTF TEXT = "^S={font_size=20pt tall title";
ODS RTF TEXT = "^S={font=('ARIAL',20pt,bold)} tall, bold and arial title";
ODS RTF CLOSE;
```

The styles are also available within PROC REPORT tables itself:

```
ODS RTF FILE="C:\temp\example.rtf";
PROC REPORT DATA=sashelp.class NOWINDOWS HEADLINE MISSING;
    COLUMN sex name age height weight;
    DEFINE sex / group noprint;
    COMPUTE BEFORE sex;
        LINE '^S={just=left font_weight=bold}Sex=' sex $1.;
    ENDCOMP;
```

```
RUN;
ODS RTF CLOSE;
```

Name	Age	Height	Weight
Sex=F			
Alice	13	56.5	84
Barbara	13	65.3	98
Sex=M			
Alfred	14	69	112.5
Henry	14	63.5	102.5
James	12	57.3	83

Figure 2: Embedding bold style within PROC REPORT

These layout blocks might also be applied to different positions within the complete block. For example it is possible to only write the first letter of a word in bold but not the rest:

```
ODS RTF FILE="C:\temp\example.rtf";
PROC REPORT DATA=sashelp.class NOWINDOWS HEADLINE MISSING;
    COLUMN sex name age height weight;
    DEFINE sex / group noprint;
    COMPUTE BEFORE sex;
        LINE '^S={just=left font_weight=bold}S^S={font_weight=medium}ex=' sex $1.;
    ENDCOMP;
RUN;
ODS RTF CLOSE;
```

Sometimes the order of the commands is important as well. For example the alignment of the following example is not left aligned as assumed, even though only an additional line has been included. To get a left aligned content, the corresponding command has to be in the first line statement.

```
ODS RTF FILE="C:\temp\example.rtf";
PROC REPORT DATA=sashelp.class NOWINDOWS HEADLINE MISSING;
    column sex name age height weight;
    define sex / group noprint;
    compute before sex;
        line '';
        line '^S={just=left font_weight=bold}Sex=' sex $1.;
    endcomp;
RUN;
ODS RTF CLOSE;
```

STYLING WITH RTF

USE EMBEDDED RTF TAGS

The most interesting and flexible way to design the RTF document is given by using direct RTF commands and field functions. The various RTF commands (tags) can be used in very different ways. When using the RTF syntax with a RTF command in the output, SAS typically masks those special characters. This protect option can be deactivated by the *PROTECTSPECIALCHAR* option. To avoid getting RTF errors accidentally, because of having those special characters in your data (e.g. in free text fields), I recommand only to tell SAS at special places to leave the tags unmasked.

When a text output starts with *{* and ends with *}* SAS does not mask special characters which are in between. Alternatively the *ESCAPECHAR* can be used to avoid the masking. As one sample the tag "\b" is for making a text passage bold.

The following example describes the masking functionality. In the first line the *\b* does appear as text in the output. In the second and third example the masking is removed. For this, the last word appears in bold style and the *\b*

tag is hidden.

```
1)  ODS RTF TEXT = "The third \b word";
2)  ODS RTF TEXT = "{The third \b word}";
3)  ODS RTF TEXT = "The third ^R/RTF'\b 'word";
```

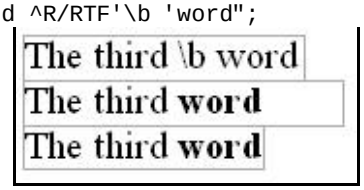

Figure 3: Example for using the bold tag

Common RTF tags are mentioned on the SAS page for ODS FAQS and Concepts [2]. A very detailed description about them all are explained in the RTF specification [4]. A very good way to find special tags is to use a different program which can create RTFs, like Microsoft Word. After saving a file as RTF, the syntax can be analysed by opening the file with a text editor. The corresponding tag and position it of interest. Unfortunately most of the programs creating RTF files use much more tags then needed, so often it is very time consuming to search and test a lot to find the corresponding tag.

FIELD FUNCTIONS

A different but very useful kind of tags are field functions. All functions which are supported by the program with which the file is opened (like Microsoft Word) can be build in the file. Very popular functions are {page} for the current page, {pageof} as number of pages as well as the filename {filename} and dates like {createdate} or {savedate}.

When using Microsoft Word, those field functions can be detected with less effort. Just create a field (Insert -> Field … -> select a field) and afterwards display the field function with pressing [Alt]+[F9]. This method can as well be used when a very difficult table of content with different styles should be created. With the help of Word the corresponding syntax can be seen.

To embed the fields into the document, they have to be included in special tags which indicates that the following text describes a field. The corresponding tags are “{\field{\fldinst <field name> }}”. Those tags have to be placed unmasked into the document. To do so, the prior introduced methods need to be used.

The following example demonstrates a simple use of those different field functions:

```
OPTIONS NODATE NONUMBER;
ODS RTF FILE="C:\temp\example.rtf";

TITLE1 j=r "{page {\field{\fldinst{page}}} of {\field{\fldinst{numpages}}}}";
FOOTNOTE   "file ^R/RTF'{\field{\fldinst{filename}}}' created on
           ^R/RTF'{\field{\fldinst{createdate}}}' saved on
           ^R/RTF'{\field{\fldinst{savedate \\@ ""d.M.yyyy""}}}'";

PROC REPORT DATA=sashelp.class NOWINDOWS HEADLINE MISSING;
    column sex name age height weight;
    compute before _page_;
        line 'Table ^R/RTF"{\field{\fldinst{SEQ table}}}": Listing of class';
    endcomp;
RUN;

ODS RTF CLOSE;
```

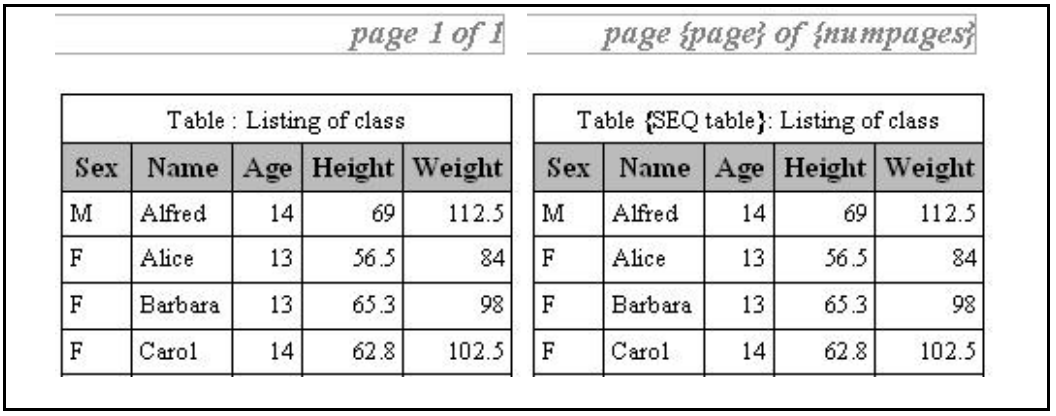

Figure 4: Field function example

The title contains "page x of y" where the footnote includes information about the filename as well as the save and creation date. Different kinds of formattings can be used for dates like showen in the example. For those there are special syntaxes which you can get in the same way as the function with the help of Word. In the table title is a sequence inserted. This can be used for auto numbering.

INDEXES

Experimentally, SAS supports a table of contents by the *CONTENTS* functionality [6]. In a few cases the result is feasible. In most cases a different TOC is needed and has to be build manually.

An index in RTF is also a simple field function. The standard table of contents from Word is *{TOC \o "1-3" \h \z \u}*. Alternatively the styles and level depths can be defined manually with *{TOC \t "heading 1;1;heading 2;2;heading 3;3"}*. This kind of table of contents can be used as well to create any other indexes. When using a style "table" for table heading a table of tables can be created with *{ TOC \h \z \c "Table" }*. Belonging to the language version of Microsoft Word, sometimes "," is expected instead of ";" as seperator within the table of contents.

The assignment of styles to RTF documents can be read in different articles [7] [8].

In the next step the field function has to be embedded in the document. RTF has as special character to indicate a following tag – the backslash. In order to have a backslash as field function specification, those characters can be masked with a double backslash. As a last layout gimmick, a text can be added which appears as long as the field is not updated. We can do this by giving text as temporary result within *{\fldrslt {text}}*.

```
ODS RTF FILE="C:\temp\example.rtf" bodytitle;
ODS RTF TEXT='^R/RTF"{\field{\fldinst{TOC \\o ''1-3'' \\h \\z \\u}}
                 {\fldrslt {table of contents }}}"';
ODS RTF CLOSE;
```

A disadvantage of using ODS RTF TEXT is, that the text is always created within a cell which has the length of the longest text. To display the compiled TOC in a better way, a cell width of 100% should be used. This can be achived with the *OUTPUTWIDTH* style:

```
ODS RTF FILE="C:\temp\example.rtf" bodytitle;
ODS RTF TEXT='^S={outputwidth=100% just=l}^R/RTF"
                 {\field{\fldinst{TOC \\o ''1-3'' \\h \\z \\u}}
                 {\fldrslt {table of contents }}}"';
ODS RTF CLOSE;
```

LINE TAGS

Whenever a very specific layout with special lining is needed, this can not be realized with the ODS styles. For this it is feasible to create a line on special places. Unfortunately the line drawing of a table and single cells can not be influenced by SAS. But by using paragraph lines, it is possible to create lines on different locations. For this topic there exist a very interesting paper [5]. The following example displays the functionality of those tags in short:

```
ODS RTF FILE="C:\temp\example.rtf" style=journal;
PROC REPORT DATA=sashelp.class NOWINDOWS HEADLINE MISSING;
    column sex name age height weight;
```

```
    define sex / group;
    compute before _page_;
        line '^R/RTF"\brdrb\brdrs\brdrw15 "Table: Listing';
    endcomp;
    compute after sex;
        line '^R/RTF"\brdrt\brdrs\brdrw15"';
    endcomp;
RUN;
ODS RTF CLOSE;
```

In the example a line is below the table title and above each grouping by sex. Sometimes those lines are not displayed by the processing program, when printing or using different viewing options, those lines are visible.

INDENTS IN RTF

Another functionality which is often needed, is the use of indents. In ASCII leading spaces are used to create indents. Indeed, SAS supports this with the option AXIS, which does not delete leading spaces on the RTF output. Often indents instead of leading spaces are wished. The ODS style element “INDENT” can be used to create not only simulated but real indents. In most cases this element is not flexible enough for example when due to different leading spaces different indents should be created. The RTF tag for indents is *li* plus a number, which gives the width of the indent.

To display a following example, this dataset is created first, having leading spaces:

```
PROC SORT DATA=sashelp.class OUT=class; BY sex name; RUN;
DATA class2;
    SET class;
    BY sex;
    LENGTH text $20;
    text = '   ' || TRIM(name);
    sort=2;
    OUTPUT;
    IF first.sex
    THEN DO;
        IF sex = "F" THEN text = "Female";
                     ELSE text = "Male";
        sort = 1;
        age=.; height=.; weight=.;
        OUTPUT;
    END;
RUN;
PROC SORT DATA=class2; BY sex sort; RUN;
OPTIONS MISSING='';
```

The use of the “ASIS” option is shown in this example:

```
ODS RTF FILE="C:\temp\example.rtf" style=journal;
PROC REPORT DATA=class2 NOWINDOWS HEADLINE MISSING style(column)={asis=on};
    column text age height weight;
RUN;
ODS RTF CLOSE;
```

To create indents depending on the number of leading spaces, the following code can be used:

```
ODS RTF FILE="C:\temp\example.rtf" style=journal;
PROC REPORT DATA=class2 NOWINDOWS HEADLINE MISSING;
    COLUMN text age height weight;
    DEFINE sex / GROUP noprint;
    COMPUTE text;
        spacediff = rxmatch(rxparse("^' '"), text) * 70;
        IF spacediff > 70
            THEN CALL DEFINE(_col_, "Style",
```

```
  "STYLE=[cellwidth=3cm
                pretext='^R/RTF""\keepn\li" ||
  compress(spacediff) || " ""']");
    ENDCOMP;
RUN;
ODS RTF CLOSE;
```

text	Age	Height	Weight
Female			
Alice	13	56.5	84
Barbara	13	65.3	98
Carol	14	62.8	102.5
Jane	12	59.8	84.5
Janet	15	62.5	112.5
Joyce	11	51.3	50.5

Figure 5: Example for using indents

To explain the last example, at first the number of leading spaces is identified with the help of a regular expression. This number is multiplied by a constant. In this example 70 was selected, because this is the typical width for a space with the Times New Roman font on size 10. Whenever a leading space is available, this line is indented by the calculated value (pretext).
Unfortunately, SAS does not take the indents into account, when calculating the column width. For this the width has to be set manually to avoid unwanted line breaks.

POST PROCESSING

A last possibility to optimize the design of the RTF document is the post processing. All features which could not be realized, can be created in post processing steps. By using auto numbering within a repeated title for example, there is the problem of having also a repeated field function which displays then two numbers for the same table. This can happen when tables are split to different tables, whenever a page variable is used or a table is to width. In a post processing step those repeated titles can be deleted or the auto number field is replaced by a reference to the last field.
Another point is that SAS creates an own heading and footing line for each new table when using *BODYTITLE*. Sometimes this can cause problems. Those can be deleted with a simple data step.
Using page breaks is very frustrating from time to time. The statement *ODS RTF STARTPAGE=NOW* does often not work. Also when tables are too wide and splitted, no page break is inserted by SAS. Creating page breaks is sometimes only possible on post processing.
In which way the post processing is performed does not matter at all. A data step as well as the programming language PEARL provides the needed possibilities. The scripting lanuage Visual Basic for Mircosoft Word is also an alternative. The most important point is performance, knowledge of the used methods and stability of the results.

CONCLUSION

With the mentioned methods of using ODS with a mixture of direct RTF commands and post processing, everything can be realized, which can be done with RTF. The ODS system supports a lot of basic options which are very easy to use. In the course of further developments we are very anxious which new features will be implemented. Then less and less special processing will be needed.
Especially the tag set RTF from ODS MARKUP could be a major benefit to include special tags in the default tag set instead of using those tags in every report. We will see how flexible this will be within the next release of SAS 9.2 [9].
Recapitulate, we see that there are different complexity ranges which can be used within SAS to layout RTF and are very flexible. Initially, as much as possible should be realized with ODS RTF. The source code is easy readable

and understandable. In the next step missing functionality should be tried to realize with direct RTF commands. If this also does not support a solution, then – as a last step – the post processing should be used. Especially the post processing can become very complex and for this should be reduced to a minimum. With those three flexible ways everything can be realized, which is supported by SAS.

REFERENCES

[1] Lauren Haworth [2004] „SAS with Style: Creating your own ODS Style Template for RTF Output", SUGI 29 – Paper 125-29, http:\\www2.sas.com/proceedings/sugi29/125-29.pdf

[2] SAS Support – "ODS FAQ and Concepts – The RTF Destination" - http://support.sas.com/rnd/base/topics/templateFAQ/Template_rtf.html

[3] SAS 9.3.1 Online documentation, "Concepts: REPORT Procedure" - http://support.sas.com/onlinedoc/913/getDoc/de/proc.hlp/a000146851.htm

[4] Rich Text Format (RTF) Specification, version 1.6, Microsoft MSDN - http://msdn2.microsoft.com/en-us/library/aa140277(office.10).aspx

[5] Steve Prust [2005] "Experiences of using ODS : moving from ASCII to RTF output", PHUSE 2005 – Paper TS06 - http://www.lexjansen.com/phuse/2005/ts/ts06.pdf

[6] SAS Support – "Experimental RTF Features in SAS 9.1" - http://support.sas.com/rnd/base/topics/odsrtf/rtf901.html

[7] Lauren Haworth [2005] „Applying Microsoft Word Styles to ODS RTF Output", SUGI 30 – Paper 043-30 - http://www2.sas.com/proceedings/sugi30/043-30.pdf

[8] Elizabeth Axelrod, David Shamlin [2004] „Skinning the Cat This Way and That: Using ODS to Create Word Documents That Work for You", SUGI 29 – Paper 084-29 - http://www2.sas.com/proceedings/sugi29/084-29.pdf

[9] Wayne Hester [2006] "Teaching Your RTF Tagset to Do Clever Tricks" – SUGI 31 – Paper 067 – 31 - http://www2.sas.com/proceedings/sugi31/067-31.pdf

RECOMMENDED READING

- SAS Support – "ODS FAQ and Concepts – The RTF Destination" - http://support.sas.com/rnd/base/topics/templateFAQ/Template_rtf.html
- Steve Prust [2005] "Experiences of using ODS : moving from ASCII to RTF output", PHUSE 2005 – Paper TS06 - http://www.lexjansen.com/phuse/2005/ts/ts06.pdf
- Elizabeth Axelrod, David Shamlin [2004] „Skinning the Cat This Way and That: Using ODS to Create Word Documents That Work for You", SUGI 29 – Paper 084-29 - http://www2.sas.com/proceedings/sugi29/084-29.pdf

CONTACT INFORMATION

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

Katja Glaß
Bayer Schering Pharma AG
Sellerstr. 31
13342 Berlin
Katja.Glass@bayerhealthcare.com