



ABSTRACT

The ability to send output to Microsoft Word was the most commonly requested function during the development of the Output Delivery System. Since version 8.1 ODS RTF has been formally available. The Rich Text Format (RTF) specification is a method of encoding formatted text and graphics which is read by Microsoft Word and several other text editors. So now that we have the ability to send output to Word, how can we apply it? What are the advantages over traditional fixed space output? How can we go beyond the documentation to extract some of Microsoft Word's features? This paper will address these questions demonstrating how to define and apply generic appearance to a project and consider the advantages of RTF output to both the programmer and end user. Furthermore it will be shown how to go beyond the documentation by embedding RTF codes to enhance your reports. Finally this paper addresses some practical considerations when generating and distributing RTF files.

INTRODUCTION

For several years as both a statistician and SAS programmer one of the most consistent problems I have experienced is the distribution of reports and graphics generated from SAS. Whilst sending a text file to a recipient is simple enough, for that person to view the text in the intended format requires them to know the page settings, font and font sizes then configure their software before viewing and printing.

Although there are several methods of importing traditional listing output into Microsoft Word via either DDE from within SAS or using Microsoft Word itself, this is a cumbersome task requiring additional processing and time.

Sending reports directly from your reporting procedure or reporting data step to a Word document eliminates these problems.

This paper discusses features available in SAS V8.2.

THE MECHANICS OF RTF

What is "RTF"?

A Rich Text Format file is a text file using defined control words and symbols that preserve the formatting of text. In SAS 8.2 the RTF specification used creates Word 2000 documents. SAS 8.1 created Word 97 documents.

To ODS RTF

Just two additional lines of code are needed to generate a simple RTF file. Consider the following:

```
ods rtf file="report1.rtf";

title "Listing of CLASS Data";
proc print data=sashelp.class;
run;

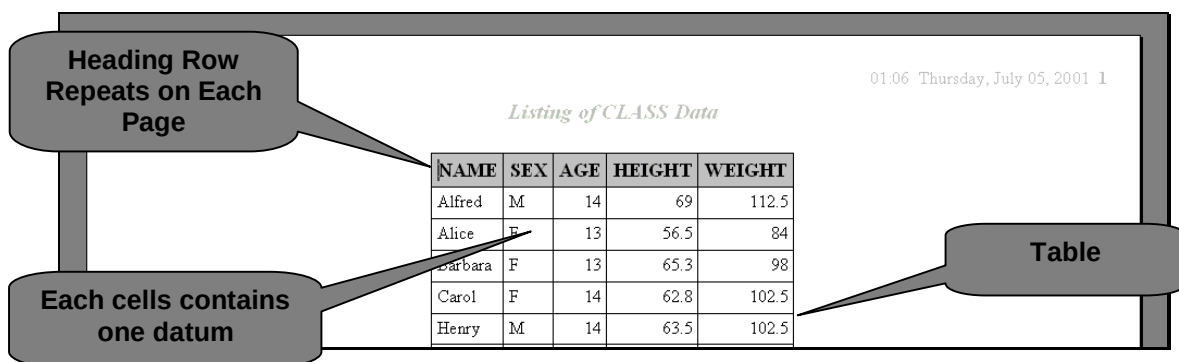
ods rtf close;
```

The first ODS statement opens the RTF destination; all output generated will be sent to the file report1.rtf. If this file already exists it will be overwritten. It is only when the ODS RTF CLOSE statements are passed that the RTF file is saved to disk.

So end of discussion then?

Well if we look at the output created there could be many elements of the default appearance we wish to customise, perhaps enhance specific areas of the report dynamically, or to conform with our companies guidelines, or just to adjust the page settings of the document.

At this point we consider how the output generated is stored inside Word. Taking the partial output from the example above, we can see that a Word table is divided into rows and columns to store the printed output, where each cell contains one element of the report.



01:06 Thursday, July 05, 2001 1

Listing of CLASS Data

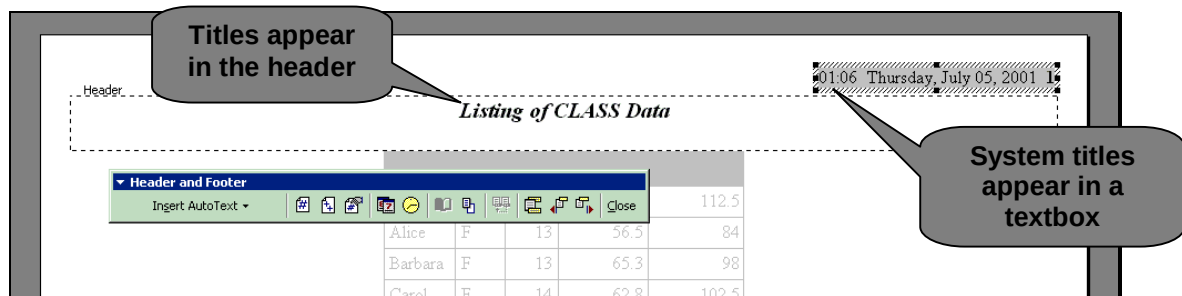
NAME	SEX	AGE	HEIGHT	WEIGHT
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5

Table

Heading Row Repeats on Each Page

Each cells contains one datum

Titles and footnotes are stored in the header and footer sections of the document, respectively:



Header

Titles appear in the header

Listing of CLASS Data

01:06 Thursday, July 05, 2001 1

System titles appear in a textbox

NAME	SEX	AGE	HEIGHT	WEIGHT
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5

▼ Header and Footer
Insert AutoText ▼

The default RTF style shades the cells of the header row grey, and makes the header row text bold. All the cell borders are drawn.

When more than one procedure's output is generated in the same file a section break is added to the document. This also allows the titles to change.

What are the advantages of using RTF?

Compared to the traditional listing output of version 6 the advantages seem quite obvious. However the use of an alternative format for your output may require changes to your standard working practices and therefore you should be able to justify its use.

By using ODS and standard SAS reporting procedures the programmer no longer needs to write additional code to post process output. In particular distributing graphics in a format which is easily viewed and printed has previously been



cumbersome. In SAS 8.2 SAS/Graph output can be sent directly to a Word document. For example:

```
ods rtf file="graph1.rtf" ;

goptions reset=all;

proc gplot data=sashelp.class;
  plot weight*height=sex;
run;
quit;

ods rtf close;
```

Advantages to the reader are most obviously the appearance. Whilst also pleasing to the eye, the ability to change fonts, font sizes, even colouring and shading can be used to enhance various aspects of the report, improve readability and provide an overall professional presentation. As Microsoft Word and other software packages that read Word files are widely used, the end user has a tool to view both text and graphical reports produced from within SAS.

Defining a Generic Appearance for a Project

Whilst we can use the style options within reporting procedures such as proc report and tabulate to define exactly how our Word document will look, it is more efficient and easier to apply generic changes if we define and store a style using proc template.

By using style statements at individual procedure level we can override the generic styles we defined should we need a specific appearance for individual tables.

Furthermore we can define settings such as the paper size and margins, through the options statement:

```
options papersize=A4 topmargin="1in" leftmargin="1in"
      bottommargin="1in" rightmargin="1in";
```

Supposing we wish to define a look which minimises on drawing border cells in the output. We can use proc template to inherit a supplied style which closely resembles what we are aiming for and edit its attributes to suit our needs.

```
proc template ;
  define style custom / store=project.styles;
  parent = styles.printer;
  replace fonts /
    'TitleFont2' = ("Verdana, Arial, Helvetica",10pt)
    'TitleFont' = ("Verdana, Arial, Helvetica",11pt,Bold)
    'StrongFont' = ("Verdana, Arial",10pt,Bold)
    'EmphasisFont' = ("Verdana, Arial",10pt,Medium)
    'FixedEmphasisFont' = ("Courier New, Courier",9pt,Italic)
    'FixedStrongFont' = ("Courier New, Courier",9pt,Bold)
    'FixedHeadingFont' = ("Courier New, Courier",9pt,Bold)
    'BatchFixedFont' = ("SAS Monospace, Courier",7pt)
    'FixedFont' = ("Courier New, Courier",9pt)
    'headingEmphasisFont' = ("Verdana, Arial",11pt,Bold Italic)
    'headingFont' = ("Verdana, Times",11pt,Bold)
    'docFont' = ("Verdana, Times",10pt);
  style Table from output /
    background = _undef_
    frame = HSIDES
```



```
rules = GROUPS
cellpadding = 5pt
cellspacing = 0pt
borderwidth = 1pt;
replace color_list /
  "bg" = white
  "fg" = black
  "bgH" = white
  "link" = blue ;
end;
run;
```

Applying a Generic Appearance to a Project

Style definitions are stored in an “item store” which is located in a SAS library. This can then be made available to our SAS session with the ODS PATH statement.

```
ODS PATH project.styles(read) sashelp.tmplmst(read);
```

To make the style available to multiple users simultaneously locate the item store in a SAS library available to all users on the project and give read-only access to the styles. Specifying the default SAS item store after our customised item store tells SAS to use the default style attributes if there are any missing definitions in our customised style.

Such an ODS PATH statement is ideally located in a project autoexec file. Note, however, if we need to alter our customised style we must change the access to the item store to update mode:

```
ods path project.styles(update);
```

Operating Systems

ODS RTF is available on all operating systems, not just Windows. Note that the RTF is formatted for the environment it is created in. Should you wish to create RTF in one environment and view the RTF in another environment the record separator may need to be altered. For example, if creating RTF files on a mainframe for viewing under Windows then the appropriate record separator syntax for ASCII is:

```
ODS RTF RECORD_SEPARATOR='0D0A'x
```

Record separator takes a hexadecimal number.

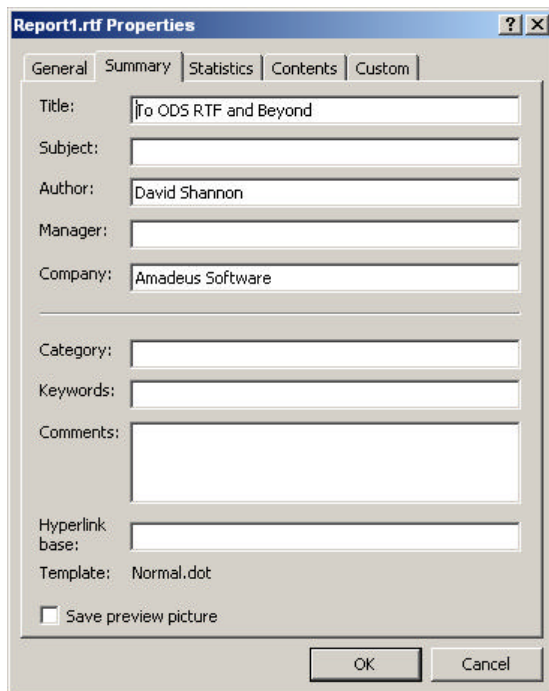
ODS RTF Features

File Properties

There are several options of the ODS RTF statement which we may use to enhance both the attributes of the document and the contents.

We can set the document author and title properties on the ODS RTF statement as follows:

```
ODS RTF Author="David Shannon" Title="To ODS RTF and Beyond"  
File="Report1.rtf";
```



By default SAS creates the document author as "V8 SAS System Output" and Title as "SAS Version 8".

These above options add the title and author information to the document we created. By selecting File – Properties from within Word we can see this information.

Bookmarks

We can make use of is Word's bookmark facility. When SAS creates each new table of output the upper left most cell contains a bookmark named IDX. Graphic objects generated by SAS are also book marked. We can change this for each new table created with the following syntax:

```
ods rtf bookmark="REPORT_1";
```

Page Breaks

The startpage option can be used to control where and when Word places page breaks:

```
ods startpage= now | never | yes | no;
```

Value	Action
Now	Forces a page break now
Never	Prevents all page breaks
Yes	Puts a page break between all outputs
No	Prevents page breaks except around graphics

The startpage option can be used either on its own ods statement or with other ods rtf options. For example:

```
ods rtf file="report.rtf" startpage=never;
  <procedure>
  <procedure>
ods startpage=now;
  <procedure>
ods rtf close;
```

Simultaneously Creating Multiple RTF Files

There are situations when we may wish to send output to multiple RTF files simultaneously. We can achieve this by assigning an ID to the open destination.

```
ods rtf(1) file="report1.rtf";
ods rtf(2) file="report2.rtf";
```

When creating multiple files we can also select or exclude output objects to either or both of these files or change the appearance by using the style option on the ods statement.

For example, consider the output generated by proc glm, we may only be interested in specific parts of the results for conveying information, however we may want to retain all the output for documentation. This can be achieved as follows:

```
ods listing close;
ods noproctitle;

title1 h=11pt "Weight = Sex + Age";

ods rtf(1) file="full glm output.rtf" style=minimal ;
ods rtf(2) file="partial glm output.rtf" style=custom;

ods rtf(2) select modelanova estimates;

proc glm data=sashelp.class;
  class sex;
  model weight = sex age;
  estimate 'Male vs. Female' sex 1 -1;
run;
quit;
ods _all_ close;
```



Escape Sequences

So far all examples of customising text with styles etc. are applied to a complete cell or table. Additionally it is possible to format a section of text within a cell by embedding the appropriate RTF code. Some of these codes are now available via escape sequences which remove the ambiguity of embedding raw RTF codes.

```
data a;
  length text $300;
  text="We can force a soft return here: \-2n Then continue "||
    "on with our text as normal. Or we can add further "||
    "formatting such as \{super superscript text} or \{sub "||
    "subscript text} like that. \S={font_style=italic}There"||
    "\S={} are of course\S={font_weight=bold} several\S={} "||
    "other text formats." ;
run;

ods escapechar='\';

ods listing close;
ods rtf file="escape sequence.rtf" style=custom;
proc report data=a noheader nofs style={asis=yes frame=box
rules=none};
  column text;
  define text / style={cellwidth=10cm};
run;
ods rtf close;
```

produces the following output:

We can force a soft return here: Then continue on with our text as normal. Or we can add further formatting such as ^{superscript text} or _{subscript text} like that. <i>There</i> are of course several other text formats.
--



AND BEYOND

Page Numbering

Inserting footnotes as fields such as the “Page X of Y” auto text entry into your document can be achieved by embedding the RTF fields PAGE and Numpages in a footnote:

```
Footnote1 J=R '{Page \field {\*\fldinst PAGE \*MERGEFORMAT}} {of
\field {\*\fldinst Numpages \*MERGEFORMAT} }';
```

The above footnote statement embeds the necessary RTF code into the Word document to display the Page X of Y field. When the document is opened the page numbers may initially appear wrong, this is because Word has not refreshed the fields. When the document is printed the fields will be corrected.

We can modify the starting page number in our output by using the pgnstartN RTF field, where the suffixed N is the desired start number.

The following sets the start page number of 2:

```
Footnote1 J=R '{Page \field {\*\fldinst {\pgnstart2 PAGE}
\*MERGEFORMAT}}';
```

Tables of Contents

One practical feature would be to insert a table of contents at the beginning of a document, which may contain several tables or graphics. As a table can have embedded hyperlinks we can navigate our way around the document in a similar way to a web page. Whilst Word can create a table of contents based on many styles, it cannot create them based on bookmarks, which are already embedded in our output.

We can however improvise. The following code creates a listing, a summary table and a graphic; where the first data step and proc report insert fields of links to the bookmarks.

```
data t;
  length toc $100 pageno $100;
  toc='{ \field {\*\fldinst HYPERLINK \l "Listing"}}';
  pageno='{ \field {\*\fldinst PAGeref Listing \h}}';
  output;
  toc='{ \field {\*\fldinst HYPERLINK \l "Summary_Table"}}';
  pageno='{ \field {\*\fldinst PAGeref Summary_Table \h}}';
  output;
  toc='{ \field {\*\fldinst HYPERLINK \l "Graphic"}}';
  pageno='{ \field {\*\fldinst PAGeref Graphic \h}}';
  output;
run;

ods rtf file="bookmark.rtf" style=custom
  author="David Shannon" title ="To ODS RTF and Beyond";

proc report data=t nowd
  style={protectspecialchars=off rules=groups frame=hsides};
  column toc pageno;
  define toc / display 'Table' left style={cellwidth=5cm Just=Left};
  define pageno / display 'Page' style={cellwidth=2cm Just=Right};
run;
```



```
ods rtf bookmark="Listing";

proc report data=sashelp.class nowd;
run;

ods rtf bookmark="Summary Table";
proc tabulate data=sashelp.class ;
  class sex;
  var height weight;
  table (height weight)*mean, sex;
run;

ods rtf bookmark='Graphic';
goptions papersize=a4 rotate=landscape device=activex;
proc gplot data=sashelp.class;
  plot height * weight = sex;
run;
quit;

ods rtf close;
```

The table of contents is produced on the first page:

Table	Page
Listing	2
Summary Table	4
Graphic	5

Word must refresh the fields before the page numbers display the correct page numbers in the document. This can be achieved by selecting the output and pressing F9 from within Word, or by printing the document.

Clicking on the blue hyperlink or page number jumps to that output. Word then displays a web toolbar which allows navigation around the document, in a similar way to a web page.



Decimal Alignment

A commonly requested feature is to align decimal values in reports. This can be achieved by specifying the RTF code "tqdec" and the distance to align the decimal from the left border in a unit called twips (there are 1440 twips in an inch or, one twip is 1/20th of a point).

For example within a proc report define statement, decimal alignment can be specified with 400 twips from the left cell margin by adding the following style statement:

```
style(column)={protectspecialchars=off pretext="\tqdec\tx400 "};
```

Producing output such as the following:

Weight (kg)	Female	n	9
		Mean	90.1
		Median	90.0
		SD	19.38

Blinking Text Effects

A further example of enhanced customisation may be to apply a format to given values to make them blink:

```
options orientation=portrait papersize=a4;
```

```
title "Highlight Age 16";
```

```
proc format;
  value age
    16 = '{\animtext2 16}';
run;
```

```
ods rtf file="animated.rtf" style=custom;
```

```
proc report data=sashelp.class nofs style={protectspecialchars=off};
  format age age.;
run;
```

```
ods rtf close;
```

Produces the following output (partial): (however when actually viewed in Word the background of the number 16 is actually blinking.)

John	M	12	59	99.5
Joyce	F	11	51.3	50.5
Judy	F	14	64.3	90
Louise	F	12	56.3	77
Mary	F	15	66.5	112
Philip	M	16	72	150
Robert	M	12	64.8	128
Ronald	M	15	67	133
Thomas	M	11	57.5	85
William	M	15	66.5	112

PRACTICAL CONSIDERATIONS

When creating reports which span several pages, the size of the resulting document should be considered. RTF files become large relative to traditional listing output, hence consideration should be given to storage space and the media used for distributing your files.

Features within Microsoft Word vary between versions, if generating output to ultimately be viewed on older version of Word or other editor, consideration should be given to which features will and will not be honoured.

CONCLUSIONS

Compared to the traditional listing output we can quickly create reports which are both professional in appearance and easy to distribute.

There are benefits to both the programmer in terms of reduced manipulation of output and also to the user for enhanced and professional appearance.

Although initially complex it is possible to extract some of Microsoft Word's features to further enhance RTF output.

The life of ODS RTF is young and there are several areas which could be developed such as improving the ability to use Word's fields, the ability to merge cells and control the drawing of specific cell borders.



TRADEMARK INFORMATION

SAS® is a registered trademark of the SAS Institute Inc. Cary, NC, USA.

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

REFERENCES

Schellenberger, Brian T., "Presentation-Quality Tabular Output via ODS", SAS Institute Inc. 2000.

From the web:

msdn.microsoft.com/library/default.asp?url=/library/en-us/dnrtfspec/html/rftspec.asp

THE AUTHOR

I very much welcome your feedback on this paper or your thoughts on the topic of ODS RTF. Contact me by sending an email to david.shannon@amadeus.co.uk or even by the traditional method:

David Shannon
Amadeus Software Ltd.
Orchard Farm
Witney Lane
Leafield
Oxfordshire
OX29 9PG

www.amadeus.co.uk

Version 1.0, 13th August 2001.