

Paper 088-29

Perfecting Report Output to RTF

Steven Feder, Federal Reserve Board, Washington, D.C.

ABSTRACT

Output Delivery System (ODS) output to RTF presents possibilities for creating publication-ready final documents directly from the SAS® System. Users can customize many appearance features exactly as needed, with ODS settings for the entire document, or for individual elements within a PROC REPORT table.

INTRODUCTION

In this paper we will look at PROC REPORT output to the RTF destination, as designed to fit specific publication format requirements. In developing a single application to user specifications, the code will include ODS features applying to the entire page, to specific table elements, and to titles and footnotes. Data manipulation will also prepare the formatting for specific RTF features. The application will reside on the MICROSOFT® WINDOWS® system, but SAS can produce similar results on other platforms. Version 9 is required.

BASIC OUTPUT

A PROC REPORT with a simple user-created format presents the basic table content:

```
proc format;
  value now
    1= 'Percent offering'
    1.4= ' '
    1.5='Single-fee account'
    2= 'Percent offering'
    3= ' Monthly fee (low balance)';
run;
proc report nowindows data=in.nowtotal;
  columns rownum col1 col2 col3;
  define rownum/display format=now.
    'Account availability/and fee averages';
  define col1/display "2000";
  define col2/display "2001";
  define col3/display 'Change';
run;
```

GENERAL PAGE APPEARANCE

The basic code directs output to an RTF file, but first specifies the general page appearance. To set fonts and margins PROC TEMPLATE creates a style to which the ODS statement refers. The ODS statement also specifies two-column output, standard for many publications.

```
proc template;
  define style textstyle;
    parent=styles.rtf;
    style usertext from usertext;
    replace fonts /
      'docFont' = ("Times Roman",8pt)
      'headingFont' = ("Times Roman",8pt);
    replace Body from Document
      "Controls the Body file." /
      bottommargin = 0in
      topmargin = 0in
      rightmargin = 1.0in
      leftmargin = 1.0in;
  end;
run;
```

```
options orientation=portrait;
ods verify off;
ods rtf file="rpt1.rtf" columns=2 style=textstyle;
```

The code produces this output:

Account availability and fee averages	2000	2001	Change
Percent offering	96.2	95.6	lm-.6
Percent offering	49.5	40.9	lm-8.6**
Monthly fee (low balance)	8.15	8.71	.56**

SETTING STYLES FOR TABLE ELEMENTS

The output now fits the general page format needed for publication, but specific elements of the table appearance are not quite as needed. Style references added to the PROC REPORT control specific elements of the table appearance:

```
proc report nowindows data=in.nowtotal
  style=[frame=hsides] /*not in effect for header*/
  style(header)=[background=white];
  columns rownum col1 col2 col3;
  define rownum/display style(column)=[just=l]
  'Account availability/and fee averages' format=now.;
  define col1/display style(column)=[just=d] "2000";
  define col2/display style(column)=[just=d] "2001";
  define col3/display style(column)=[just=d] 'Change';
run;
```

The references within the PROC REPORT statement set the header background color to white and remove the outside of the table frame as required. (The last time I checked this was not possible for the header, and in fact this element does not meet the user's specification.) The style reference within the DEFINE statements sets left justification for the label column and decimal justification for each of the results columns, with the resulting output:

Account availability and fee averages	2000	2001	Change
Percent offering	96.2	95.6	lm-.6
Percent offering	49.5	40.9	lm-8.6**
Monthly fee (low balance)	8.15	8.71	.56**

ESCAPECHAR FOR TEXT MODIFICATIONS

Next, the user specifications require that certain lines of the table have footnote numbers and print italicized. To do this, first set a value for ESCAPECHAR, which indicates a character which will be interpreted as preceding an RTF command within braces. By adding the ESCAPECHAR to the PROC FORMAT, the formatted text value will now print with a superscript footnote number, although not yet italicized:

```
ods escapechar='^';
proc format;
value now 1 = 'Percent offering'
          1.4 = ' '
          1.5 = 'Single-fee account^{super 1}'
          2 = 'Percent offering'
          3 = ' Monthly fee (low balance)';
run;
```

To italicize selectively, add a COMPUTE block after the last DEFINE to search for the ESCAPECHAR in the formatted variable and set a style for that row.

```
define col3/display style(column)=[just=d] 'Change';
compute rownum;
  if index(put(rownum,now.),'^') then do;
    call define('rownum',"style","style=[font_style=italic rules=none]");
  end;
endcomp;
```

Lastly, in a DATA STEP before the PROC REPORT, search for the ESCAPECHAR to insert a blank line before the footnoted line:

```
data temp;
  set in.nowtotal;
  output;
  if index(put((rownum-.5),now.),'^') then do;
    col1=' ';
    col2=' ';
    col3=' ';
    rownum=rownum-.5;
    output;
    rownum=rownum-.1;
    output;
  end;
run;

proc sort;
  by rownum;
run;
```

The result now matches the user requirements for the table itself (except for the outer rule on the header):

Account availability and fee averages	2000	2001	Change
Percent offering	96.2	95.6	lm-.6
<hr/>			
<i>Single-fee account¹</i>			
Percent offering	49.5	40.9	lm-8.6**
Monthly fee (low balance)	8.15	8.71	.56**

TITLES AND FOOTNOTES

ODS RTF titles and footnotes by default appear associated with the entire page. ODS provides (at least) two options for associating the titles and footnotes with a particular table of output instead. The method explicitly designed for this purpose is to use the ODS BODYTITLE option. However, the spacing is not quite as required in this case. Instead, the following code illustrates how the ODS TEXT= option writes lines of text before and after the PROC REPORT, plus blank lines, much more closely meeting the exact specifications. The ESCAPECHAR within the text sets the font, margin, and indent. The TITLE statement removes the default SAS title, and the NODATE option removes the date.

```
ods rtf text=
  ^S={font=('Times Roman',10pt,bold)}Noninterest checking accounts at medium-
  sized institutions" columns=2;
ods rtf text=' ' columns=2;
ods rtf text='^S={leftmargin=.25in}Dollars except as noted' columns=2;
```

```
ods rtf text=' ' columns=2;

options nodate;
title;

proc report nowindows data=temp
[]
run;

ods rtf text=' ' columns=2;
ods rtf text=
'^S={indent=.12in}1. A monthly fee for balances below the minimum, no monthly
fee for balances above the minimum, and no other charges.' columns=2;
```

The table now fits the specifications almost exactly, and as closely as was attainable:

Noninterest checking accounts at medium-sized institutions

Dollars except as noted

Account availability and fee averages	2000	2001	Change
Percent offering	96.2	95.6	1m-.6
<i>Single-fee account¹</i>			
Percent offering	49.5	40.9	1m-8.6**
Monthly fee (low balance)	8.15	8.71	.56**

1. A monthly fee for balances below the minimum, no monthly fee for balances above the minimum, and no other charges.

TEMPLATE IN MACRO

A project leader could in theory define a standard for publication-formatted output for use by many users by establishing shared access to an ODS template store. In this case, however, saving the PROC TEMPLATE and PROC REPORT as macros in a shared directory would provide satisfactory access and allow for further customization. But neither of these options has yet been tried.

OS/390® AND UNIX PLATFORMS

We developed this system on the WINDOWS XP® operating system but briefly tested it on the OS/390 system, using an HFS (hierarchical file system) setup, and on a UNIX system. The ODS statements run the same and produce essentially the same RTF files on these two systems. However, the escape character did not resolve correctly when the output was transported and viewed on the WINDOWS platform. If the goal were to use data on one platform and transport the output to another, this topic would require further investigation.

CONCLUSION

Directing SAS output to RTF with ODS can eliminate substantial post-processing of the results to fit exacting user requirements for publication-formatted files. A few minor formatting issues remain, and the coding may be too involved for non-programmers. Getting familiar with coding issues unique to RTF does take time, but the results are worth the effort.

CONTACT INFORMATION

Steven Feder
Federal Reserve Board, Mail Stop 157
Washington, D.C. 20551
202-452-3144
email: steven.h.feder@frb.gov

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 .