

## Mimicking NOSEPS in ODS RTF output created with Proc Report and Proc Tabulate

Twan van Berkel, NV Organon, Oss, The Netherlands  
Gertjan van Maaren, NV Organon, Oss, The Netherlands

### ABSTRACT

Within our pharmaceutical company NV Organon clinical trials are being performed and clinical trial reports and appendices are created using the SAS System®.

Clinical trial reports (in Microsoft Word® document format) contain so called report tables which typically have no horizontal gridlines in the table body and no columns in the row table space.

The report tables are created with PROC REPORT or PROC TABULATE and outputted among other things with the ODS RTF facility.

No horizontal gridlines in the table body and no columns in the row table space are typical properties of the NOSEPS option. The NOSEPS option is an option within PROC TABULATE (PROC REPORT does not have such an option) which only works for ODS listing output and not for ODS RTF output.

This paper tries to find answers how to mimic the NOSEPS option for ODS RTF output created with PROC REPORT and PROC TABULATE.

In brief, it is hard to come up with a universal and relative easy and ready to use SAS solution that gives 100% mimicry of the NOSEPS option. We researched the use of ODS RTF style elements, ODS RTF markup and wiping out RTF control words in the ODS RTF output.

For two-dimensional tables (PROC TABULATE output) the use of ODS RTF style elements does not give the desired result. The best result is a box around the table with a gridline between the table header and the table body.

For a one-dimensional table (PROC REPORT output) the use of RTF style elements gives mimicry of the NOSEPS option (although only the horizontal gridlines can be wiped out since there is no row table space).

ODS RTF MARKUP, used within SAS version 9.1.3, does not have a proper RTF tagset that is bugfree for NOSEPS testing. However, in SAS version 9.2 ODS RTF MARKUP seems to be the most promising SAS solution for detailed ODS RTF management (1, 2).

The best solution that gave mimicry of the NOSEPS option is by removing RTF code in the ODS RTF output with SAS/BASE®.

### INTRODUCTION

Within our pharmaceutical company NV Organon clinical trials are being performed and clinical trial reports and statistical appendices are created using the SAS System®.

Clinical trial reports (in Microsoft Word® document format) contain so called report tables which typically have no horizontal gridlines in the table body and no columns in the row table space such as shown in Figure 1.

No horizontal gridlines in the table body and no columns in the row table space are typical properties of the NOSEPS option.

The NOSEPS option is an option within PROC TABULATE (PROC REPORT does not have such an option) which only works for ODS listing output but not for ODS RTF output.

Now our dilemma: we want to dynamically produce report tables with ODS RTF which contain NOSEPS properties such that no additional editing has to be done to wipe out table gridlines in the report tables. The report tables we want to mimic for ODS RTF output are depicted in Figure 1.

The RTF<sup>1)</sup> format is the only 'Microsoft WORD® document like format' which is available with the SAS Output Delivery System (ODS).

First, our research focuses on techniques/possibilities which are already available within the ODS RTF facility. If the 'normal' SAS way does not give a satisfying result, a less obvious possibility is researched.

1) RTF, abbreviation for Rich Text Format, is a vendor neutral, platform independent mark-up language which easy assimilates with a Microsoft Word® document.

**Table naming conventions**

Naming conventions for the spaces in a two-dimensional table are displayed in Figure 1. Notice that in a one-dimensional table, like the PROC REPORT output only the table header and table body are distinguished.

Figure 1: Examples of tables.

One-dimensional table created with PROC REPORT:

Name	Gender	Age	Height	Weight
Alfred	Male	14	69	112.5
Alice	Female	13	56.5	84
Barbara	Female	13	65.3	98
Carol	Female	14	62.8	102.5
Henry	Male	14	63.5	102.5

Two-dimensional table created with PROC TABULATE:

		Treatment group	
		A	B
Female	N	4	5
	Mean	12.50	13.80
	SD	1.29	1.30
	Median	12.50	14.00
	Minimum	11.0	12.0
	Maximum	14.0	15.0
Male	N	4	6
	Mean	13.00	13.67
	SD	1.83	1.63
	Median	13.00	13.50
	Minimum	11.0	12.0
	Maximum	15.0	16.0

Two-dimensional table, the naming conventions:

table box space		→	table header
row table space	table data space	↘	table body

**EXAMPLE DESIGN**

Our research focusses on three different possibilities to mimic the NOSEPS option:

1. ODS RTF style elements that affect the inner borders in the table body and row table space of a table. Can individual style elements that are programmed within the procedures give a close mimicry of the NOSEPS option?
2. ODS RTF markup that directly affects the RTF syntax. ODS RTF markup is a very powerful tool that affects the RTF control words. However, is this technique already mature enough in SAS 9.1.3 to mimic the NOSEPS option?
3. Modifying RTF control words in the RTF output to get control the RTF code with SAS.

The three possibilities are described under header “Experimental”.

The tables in Figure 1 are to be created with PROC REPORT and PROC TABULATE respectively. The ‘raw’ PROC TABULATE and REPORT tables (the RTF output without the NOSEPS features) are shown in Appendix 1. The related syntax and style are displayed in Appendix 2.

The idea is that the one- and two-dimensional tables, as displayed in Figure 1, are produced with ODS RTF style elements, with ODS RTF markup and by modifying RTF control words.

The examples in Figure 1 use data and variable properties displayed in Figure 2. The data comes from the dataset CLASS that was taken from SASHELP.CLASS. The syntax used to derive the dataset CLASS, is shown in Appendix 1.

Figure 2: The dataset CLASS and its variable properties.

	NAME	SEX	AGE	HEIGHT	WEIGHT	TCP
1	Alfred	M	14	69	112.5	A
2	Alice	F	13	56.5	84	A
3	Barbara	F	13	65.3	98	B
4	Carol	F	14	62.8	102.5	B
5	Henry	M	14	63.5	102.5	B
6	James	M	12	57.3	83	A
7	Jane	F	12	59.8	84.5	A
8	Janet	F	15	62.5	112.5	B
9	Jeffrey	M	13	62.5	84	B
10	John	M	12	59	99.5	B
11	Joyce	F	11	51.3	50.5	A
12	Judy	F	14	64.3	90	A
13	Louise	F	12	56.3	77	B
14	Mary	F	15	66.5	112	B
15	Philip	M	16	72	150	B
16	Robert	M	12	64.8	128	B
17	Ronald	M	15	67	133	A
18	Thomas	M	11	57.5	85	A
19	William	M	15	66.5	112	B

  

PROC CONTENTS output:				
#	Variable	Type	Length	Format
1	NAME	Char	8	
2	SEX	Char	1	\$SEXF.
3	AGE	Num	8	
4	HEIGHT	Num	8	
5	WEIGHT	Num	8	
6	TCP	Char	1	

## CONFIGURATION

The following option should minimally be set within the SAS system:

```
OPTIONS ORIENTATION=PORTRAIT;
```

The style RTFNOSEPS, that is used to format all the experimental output, is displayed in Appendix 2 and inherits style elements from the default RTF style that was shipped with SAS version 9.1.3.

Testing was done in a Windows environment (Microsoft Windows XP® SP1.0) with SAS/BASE® version 9.1.3.

## EXPERIMENTAL

In this section the three possibilities, i.e the use of ODS RTF style elements, the use of ODS RTF MARKUP and modification of the RTF code in the ODS RTF output are described.

### POSSIBILITY 1: ODS RTF STYLE ELEMENTS THAT AFFECT THE GRIDLINES IN THE TABLE BODY AND ROW TABLE SPACE OF A TABLE

We examined whether individual style elements within the procedure syntax were able to mimic the NOSEPS option. A style element, a so called style attribute, is the smallest item within an ODS style definition and has a paired syntax:

```
<name style attribute> = <value style attribute>
```

The procedures PROC REPORT and PROC TABULATE allows you to control styles attributes within the procedure via the STYLE=option. The STYLE= option affects different areas in the table.

For PROC REPORT output, the addition of the following code to the procedure syntax gave the desired result as shown in Figure 1:

```
PROC REPORT data=class(obs=5) nowd
  style(report)=[background=_undef_ rules=cols frame=_undef_];
  <procedure syntax>
run;
```

The horizontal grid lines are removed whereas the grid line between the table header and the body of the table is preserved. Also the column grid lines have been preserved.

For PROC TABULATE output, no full mimicry of the NOSEPS options was found. By using the following style definition:

```
PROC TEMPLATE;
  define style test;
  parent=rtfnoseps;
  style table from table/
    outputwidth = 75%
    cellspacing = 0.70pt
    borderwidth = 0.75pt
    rules=groups
    frame=box;
  end;
run;

ods rtf file='<filename.rtf>' style=test;
proc tabulate data=class;
  <procedure syntax>
run;
```

The table shown in Figure 3 was created.

Figure 3: PROC TABULATE output using specific table style definitions.

		Treatment group	
		A	B
Female	N	4	5
	Mean	12.50	13.80
	SD	1.29	1.30
	Median	12.50	14.00
	Minimum	11.0	12.0
	Maximum	14.0	15.0
Male	N	4	6
	Mean	13.00	13.67
	SD	1.83	1.63
	Median	13.00	13.50
	Minimum	11.0	12.0
	Maximum	15.0	16.0

The horizontal grid lines are removed but it is impossible to preserve the column grid lines. No style option is found that could affect the column grid lines in the table body.

It can be notified that style attributes like RULES (responsible for the inner border lines) and FRAME(responsible for the outer border lines) do not seem to be flexible on table cell level and cannot be called with the STYLE= option within the PROC TABULATE syntax.

### **POSSIBILITY 2: ODS RTF MARKUP THAT DIRECTLY AFFECTS THE RTF SYNTAX**

ODS RTF MARKUP is an ODS channel that is available since SAS version 8.2 and enables, in contrast with the normal ODS RTF channel, direct access to detailed table features. ODS RTF MARKUP has its own bag of elements, called a tagset, that control and manage RTF code.

The PROC REPORT and PROC TABULATE example code were tested with the default ODS RTF MARKUP tagset that was shipped with SAS version 9.1.3:

```
ods markup type=tagsets.rtf file='test2.rtf';
<Procedure syntax;
Step boundary;>
ods rtf close;
```

This syntax gives a lot of warnings in the SAS LOG and an empty, corrupted ODS RTF document that cannot be opened. The RTF tagset seems to be experimental and buggy while the other tagsets like HTML are much more mature(3). No tests can be performed to see if the tagset features can be triggered for a NOSEPS mimicry.

The first serious (bugfree) RTF tagsets are shipped with SAS version 9.2 and not commercial available yet.

The RTF tagsets in SAS version 9.2 seem very promising following papers of Wayne Hester(1) and Eric Gebhart(2).

### **POSSIBILITY 3: MODIFYING RTF CONTROL WORDS IN THE RTF OUTPUT TO GET CONTROL THE RTF CODE**

From analytic point of view programmers always look first for simple SAS functions or solutions that do the job. However, looking at the first two possibilities, we cannot directly control the inner table gridlines, so we have to go for a less obvious solution. The last possibility modifies RTF code in ODS RTF output to mimic the NOSEPS option.

Within our department a small application has been written that mimics the NOSEPS option in ODS RTF output. This application consists of two macros, %SAS2RTF and %RTF2SAS which are placed around procedure syntax:

```
%sas2rtf(file=test3.rtf, nosep=Yes);
<Procedure syntax;
Step boundary;>
%rtf2sas;
```

The macro %SAS2RTF passes among other things the ODS RTF output to the %RTF2SAS macro. The %RTF2SAS macro modifies the RTF code to mimic the NOSEPS option if the NOSEP macro parameter is set to Yes.

The one- and two-dimensional table shown in Figure 1 are created using this application.

How does the macro %RTF2SAS mimic the NOSEPS option? Some datasteps with flags and counters do the trick and is explained in the next paragraph.

### **SAS SYNTAX THAT WIPES OUT THE RTF CODE TO MIMIC THE NOSEPS OPTION**

The SAS syntax, coming from the macro %RTF2SAS that wipes out RTF code to mimic the NOSEPS option is shown in Appendix 3. The SAS syntax is captured in another, separate macro called %NOSEPS. Wrapping in macro code is done to manage the inner border thickness and thus to make it universal for use.

The macro %NOSEPS works for ODS RTF tables that were formatted and styled as shown in Figure 1. Furthermore, it works for tables and listings created with the default style with or without titles and footnotes. The inner border thickness is made flexible and can be managed. ODS RTF tables that do not fit on one page are sometimes not processed properly. The macro was further tested with all kinds of tabular designs and gave most of the time a proper result.

The macro does not give an answer to all tabulation designs developed with PROC REPORT or PROC TABULATE but could be used as template for your own table design and style and thus adapt it accordingly.

#### *How does the %NOSEPS macro work?*

In the macro NOSEPS two data steps are used to set counters and to create benchmarks that trigger which RTF control words to remove. An RTF control word is the smallest RTF element within RTF code; see Appendix 4 for further detailed information on RTF control words. The wiped out RTF control words are common in every ODS RTF tabular format created with SAS versions 8.2 and 9.1.3. First the macro makes a distinction in table header and table

body information. In a two-dimensional table, the first inner column position in the table box area is captured, based on the distinction between the table header and body. The first inner column position marks the difference between the inner vertical border lines in the row table space and the rest of the table. As a consequence, vertical gridlines in the row table space can be removed. The horizontal gridlines in the table body can be removed based on the difference between the table header and body.

## *Usage of the macro %NOSEPS*

The macro NOSEPS works somewhat different than the SAS2RTF/RTF2SAS application.

First, create the desired RTF table and refer in the macro call to the RTF table:

```
%noseps (infile='..\Phuse2006\Example_tabulate.rtf',  
         outfile='..\Phuse2006\Example_tab_noseps.rtf',  
         twip=5);
```

The OUTFILE parameter contains the new file name with the NOSEPS mimicry.

The macro parameter TWIP represents the inner border thickness that must be wiped out (a twip is one-twentieth of a point and is the standard thickness element in the ODS RTF output).

The normal RTF default style delivers a normal inner border thickness of five twips. The inner border thickness can be arranged with the style attribute CELLSPACING. The CELLSPACING attribute is processed with PROC TEMPLATE as displayed in Appendix 2. If the value of the attribute is set to for example 0.25pt(=points) the value will appear in the ODS RTF document code as postfix number five in an RTF control word \brdrw5.

So if the cellspacing in the style is changed into 0.50pt, than the macro parameter TWIP gets the value 10. Be aware that the macro only works if there is a difference in border thickness between the inner and the outer borders (this is observed in the default style).

## **CONCLUSION**

For a one-dimensional table (PROC REPORT output) the use of RTF style elements gives mimicry of the NOSEPS option (although only the horizontal gridlines can be wiped out since there is no row table space).

For two-dimensional tables (PROC TABULATE output) the use of ODS RTF style elements does not give the desired result. The best result is a box around the table with a gridline between the table header and the table body.

Individual style elements within the procedure syntax do not have any influence on the grid lines within a table.

ODS RTF markup, used within SAS version 9.1.3, does not have a proper RTF tagset that is bugfree for NOSEPS testing. However, in SAS version 9.2 ODS RTF MARKUP seems to be the most promising SAS solution for detailed ODS RTF management (1, 2).

The best solution that gives mimicry of the NOSEPS option is by removing RTF code in the ODS RTF output with SAS/BASE®. For this purpose, a macro has been created that is able to wipe out the inner horizontal grid lines and the row table column grid lines.

## **REFERENCES**

- (1) Wayne Hester, Teaching Your RTF Tagset to Do Clever Tricks, Paper 067-31, Conference Proceedings, SUGI 31 2006
- (2) Eric Gebhart, The Beginners Guide to ODS MARKUP: Don't panic!, Paper 236-31, Conference Proceedings, SUGI 31 2006
- (3) <http://www.listserv.uga.edu/cgi-bin/wa?A2=ind0503c&L=sas-l&D=1&P=43153>

## **ACKNOWLEDGMENTS**

We would like to thank Kit Roes and Pierre Verweij for the support and the review of this paper. Furthermore, everyone within our Biometrics department who helped or gave tips regarding this topic.

## **CONTACT INFORMATION**

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

Twan van Berkel  
NV Organon  
Department of Clinical Information Oss / Biometrics  
Post-box 20  
Oss, 5340 BH  
The Netherlands  
Work Phone: +31 (0) 412 661205  
Fax: +31 (0) 412 662516  
Email: [t.vanberkel@organon.com](mailto:t.vanberkel@organon.com)

Gertjan van Maaren  
NV Organon

## PhUSE 2006

Department of Clinical Information Oss / Biometrics  
Post-box 20  
Oss, 5340 BH  
The Netherlands  
Work Phone: +31 (0) 412 663475  
Fax: +31 (0) 412 662516  
Email: gertjan.vanmaaren@organon.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.

**APPENDIX 1**

Figure 4: RTF tabular and listing output that were used by the different possibilities. The procedure syntax and style that creates the output is shown in Appendix 2.

PROC TABULATE output:

		Treatment group	
		A	B
Female	N	4	5
	Mean	12.50	13.80
	SD	1.29	1.30
	Median	12.50	14.00
	Minimum	11.0	12.0
	Maximum	14.0	15.0
Male	N	4	6
	Mean	13.00	13.67
	SD	1.83	1.63
	Median	13.00	13.50
	Minimum	11.0	12.0
	Maximum	15.0	16.0

PROC REPORT output:

Name	Gender	Age	Height	Weight
Alfred	Male	14	69	112.5
Alice	Female	13	56.5	84
Barbara	Female	13	65.3	98
Carol	Female	14	62.8	102.5
Henry	Male	14	63.5	102.5

## APPENDIX 2

The procedure syntax and style that creates the output as shown in Appendix 1.

```

proc template;
  define style rtfnoseps;
  parent=styles.rtf;
  style fonts/
    'TitleFont' = ("Courier",11pt,Bold)
    'TitleFont2' = ("Courier",11pt,Bold)
    'headingFont' = ("Courier",9pt)
    'docFont' = ("Courier",9pt);
  replace color_list
    "Colors used in the default style" /
    'link'= blue
    'bgH' = white
    'fg' = black
    'bg' = white;
  style table from table/
    outputwidth = 75%
    cellspacing = 0.70pt /*thickness of inner borders*/
    borderwidth = 0.75pt; /*thickness of outer borders*/
  style header /
    font = ("Courier",9pt, Bold);
  style RowHeader/
    font = Fonts('docFont');
  style Data/
    font=Fonts('docFont');
end;
run;

proc format;
  value $sexf 'M'='Male' 'F'='Female';
run;

data class(drop=r);
  set sashelp.class;
  call ranuni(_n_,r);
  if r < 0.4 then tcp='A'; else if r >= 0.4 then tcp='B';
run;

ods rtf file='example_tabulate.rtf' style=rtfnoseps;
proc tabulate data=class;
  class tcp sex;
  var age;
  tables sex=' *age=' *(n='N'*f=12.
    mean='Mean'*f=12.2
    std='SD'*f=12.2
    median='Median'*f=12.2
    min='Minimum'*f=12.1
    max='Maximum'*f=12.1),
    tcp='Treatment group';
  format sex $sexf.;
run;
ods rtf close;

ods rtf file='example_report.rtf' style=rtfnoseps;
proc report data=class(obs=5) nowd;
  column name sex age height weight;
  define name / 'Name';
  define sex / 'Gender' format=$sexf.;
  define age / 'Age';
  define height / 'Height';
  define weight / 'Weight';
run;
ods rtf close;

```

## APPENDIX 3

SAS syntax that mimic the NOSEPS option wrapped in the NOSEPS macro.

```

%macro noseps(infile=, outfile=, twip=5);
%*****
%*                               The macro NOSEPS
%*****
%*
%* Macro           : noseps.sas
%*
%* Author          : Twan van Berkel
%*
%* Date           : August 2005
%*
%* Purpose        : Mimic the NOSEPS option:
%*                  - Remove inner horizontal gridlines in the table or listing
%*                  body
%*                  - Remove inner vertical gridlines in the row table space in
%*                  a two-dimensional tabular format
%*
%* Parameter(s)  :
%*
%*                infile:  Input RTF file that will be modified
%*                        value: <folder path>\DocumentName.rtf
%*
%*                outfile: Output RTF file
%*                        value:
%*                        <folder path>\DocumentName<other than infile name>.rtf
%*
%*                twip:    Measurements of the inner border thickness. A twip is
%*                        one-twentieth of a point.
%*                        default value(using the default style): 5
%*
%* Examples      :
%*
%*                %noseps(infile='..\example.rtf',
%*                        outfile='..\example.rtf',
%*                        twip=5);
%* Remark(s)     : - No checks are performed on the macro parameters and its values
%*
%*****;

%local lastrownmb rtswidth;

/*
Definition of the flags and counters:

Flags:
i: To divide the general RTF header document information with the table RTF code
j: To divide the RTF table header code and the RTF table body code

Counters:
t: Counts the numbers of table body rows and stores this information in the macro
variable LASTROWNUMB. This is done because in the row table space no difference can
be made between the inner and outer border at certain places in the row table space.
To trigger on the LASTROWNUMB the last outer border is not wiped out
u: Counter that is used to determine the first column position in the box table space.
The value of the position (in twips) is saved in the macro variable RTSWIDTH.

```

## APPENDIX 3 (CONTINUED)

SAS syntax that mimic the NOSEPS option wrapped in the NOSEPS macro.

Visual:

```

i j t   u ==>           1           2

1 0 0
1 0 0 |-----|
1 0 0 |   Table box   |
1 0 0 |             |
1 0 0 |             |
1 0 0 |-----|
1 1 1 |
1 1 1 |
1 1 1 |   Row table space   |   Table body
1 1 2 |-----|
1 1 2 |
1 1 2 |
1 1 2 |

```

\*/

```

data rtfiler;
  retain i 0 j 0 t 0;
  infile &infile length=l end=lst;
  input txtline $varying200. l;

  /*trigger the first row in the header*/
  if index(txtline,'\trowd\trkeep\trhdr')>0 then i=1;

  /*trigger the first non header row: the flag i is necessary because the code
  \trowd\trkeep\trq can also appear in the table header, with i=1 this is
  bypassed*/
  if index(txtline,'\trowd\trkeep\trq')>0 and i=1 then j=1;

  /*count the number of table rows and save the number in the
  macro variable LASTROWNMB*/
  if j=1 and index(txtline,'\trowd\trkeep\trq')>0 then t+1;

  if lst then call symput('lastrownmb', left(t));
run;

data _null ;
  retain t 0;
  set rtfiler;
  file &outfile;

  /*get the cellx information from the header*/
  if j=0 then do;
  /*flag the first cellx in the header */
  if index(txtline,'\cellx')>0 and i=1 and j=0 then do;
    u+1;
    if u=1 then call symput('rtswidth', substr(txtline,index(txtline,'\cellx')+6));
  end;
end;
if j=1 then do;
  if index(txtline,'\cellx')>0 then
cellx=input(trim(substr(txtline,index(txtline,'\cellx')+6)),??best20.);
  if .<cellx<=resolve('&rtswidth') then do;
    /*Do not wipe out the outer border*/
    if t ne &lastrownmb then txtline=tranwrd(txtline,
"\clbrdrb\brdrs\brdrw15\brdrcf1\clbrdr1\brdrs\brdrw&twip", " ");
    else txtline=tranwrd(txtline, "\brdrcf1\clbrdr1\brdrs\brdrw&twip", " ");
    /*wipe out all the horizontal and vertical inner borders in the row table space*/
    txtline=tranwrd(txtline, "\clbrdrb\brdrs\brdrw&twip\brdrcf1\clbrdr1\brdrs\brdrw&twip", " ");
  end;
  /*wipe out all the remaining horizontal inner borders in the table body and row table space*/
  txtline=tranwrd(txtline, "\clbrdrb\brdrs\brdrw&twip\brdrcf1", " ");
end;
put txtline;
run;

%mend noseps;

```

**APPENDIX 4**

**RTF TABLE CODE EXPLAINED**

An RTF table is build up by a sequence of table rows. Some rows represent the table header and the rest the table body. A piece of the RTF code of a table row is displayed below:

```
\clbrdrb\brdrs\brdrw50\brdrf1
```

The example consist of four RTF codes, called control words that are divided by backslashes(\). Every control word gives an instruction on how to manage the RTF document.

The RTF control words instruct applications like Microsoft Word® to draw a cell bottom border (clbrdrb) that has a single line (brdrs) with a thickness of 50 twips (brdrw50; measurements are in twips; a twip is one-twentieth of a point) in black color(brdrf1).

The 'product' of the RTF code sequence is a bold bottom line under the second N statistic displayed in Figure 5.

This RTF example code is the code that is removed to wipe out the inner horizontal borders in an ODS RTF table format. The default thickness of almost all the inner border gridlines (recorded in the default template style) is 5 twips. The 50 twips in the example RTF code is chosen to emphasize the line in Figure 5.

Figure 5: The bold bottom line under the second N statistic, the product of the example RTF code.

		Treatment group	
		A	B
Female	N	4	5
	Mean	12.50	13.80
	SD	1.29	1.30
	Median	12.50	14.00
	Minimum	11.0	12.0
	Maximum	14.0	15.0
Male	N	4	6
	Mean	13.00	13.67
	SD	1.83	1.63
	Median	13.00	13.50
	Minimum	11.0	12.0
	Maximum	15.0	16.0