Creating Word Tables Made Easier
Carey Smoak, Roche Molecular Systems, Inc., Pleasanton, CA

ABSTRACT
Creating Word tables using ODS RTF and PROC REPORT is easier than ever using version 9.1.3 of SAS®. This paper focuses creating Word tables that conform to standards provided by the Medical Writing department at my company for in-text tables (Word tables that are inserted directly into the text of a final report). Specific tasks in creating Word tables that have been made easier with SAS 9.1.3 include: 1) the use of the BODYTITLE option and 2) specifying fonts using PROC FONTREG. Version 9.1.3 of SAS® on a Windows platform was used.

INTRODUCTION
In a previous PharmaSUG paper (Smoak 2004), I showed how to use PROC REPORT and ODS RTF to create Word tables using SAS 8.2. In this current paper, I will show how creating Word tables has been made easier using SAS 9.1.3. Specific items that are now easier include:

- the use of the BODYTITLE option
- specifying fonts with the use of PROC FONTREG

MEDICAL WRITING STANDARDS
The current standards from the Medical Writing department in my company for in-text tables (Word tables that are inserted directly into the text of a final report) are:

- Table title:
  - Should be centered above the table using TimesNewRoman, Bold, 12pt

- Column headers in the table:
  - Arial, Bold, 10 pt

- Text in the table:
  - Arial (not bold), 10 pt
  - Centered (where applicable)
  - 3pt spacing before and after text

- Footnotes:
  - Left justified using TimesNewRoman, Italic, 9pt
  - Paragraph spacing of 0.5pt before & after footnotes

PROGRAMMING ENVIRONMENT
The following SAS code sets up the program environment. Note that in version 9.1.3 of SAS you can use both the NODATE and NONUMBER options with the BODYTITLE statement (see ODS RTF statement below). In prior versions of SAS this was not allowed.

```
/****************************************************************
Note: 1) ls=80 and ps=60 for portrait orientation
   2) nodate and nonumber now work with BODYTITLE option in SAS 9.1.3
****************************************************************/
options missing=' ' nodate nonumber ls=80 ps=60 pageno=1 orientation=portrait
   mprint mlogic symbolgen;
%let dirname = u:\sas\pharmasug2007;
%let pgmname = PharmaSUG_01;
%let user = smoakc;

/****************************************************************
START PROGRAM
****************************************************************/
PROC TEMPLATE
The following PROC TEMPLATE makes changes to the RTF. Specifically the margins are set up and
CELLPADDING and CELLSPACING are set up so that there will be 3pt spacing around the text in the table.

/****************************
Setup New ODS Template Definition
********************************/
ods path work.templat(update) sasuser.templat(read) sashelp.tmplmst(read);
ods path show;
proc template;
define style print.chgRTF;
parent = styles.rtf;
/** RTF automatically has border of 0.25 in /**
style body from body /
  leftmargin = 1.25 in
  rightmargin = 1 in
  topmargin = 1.25 in
  bottommargin = .75 in;
/** For spacing before and after text in the table **/
style table from table /
  cellpadding = 3pt;
end;
run;

PROC FONTREG
PROC FONTREG is a new SAS procedure in version 9.1 (SAS Institute Inc., 2003) of SAS. This procedure will allow
you to use TrueType Fonts with TITLE and FOOTNOTE statements. To find which TrueType Fonts are available to
you, look at c:\windows\fonts or c:\winnt\fonts. Only fonts that have the “.TTF” (TrueType Fonts) are available for use
with PROC FONTREG. In this example, I am registering the TimesNewRoman and Arial fonts.

Since I use both PC SAS and SAS on a Windows Server, I have written the following macro for use with PROC
FONTREG to look in the correct folder for the True Type Fonts. The macro variable &sysscpl is automatically
generated when SAS is invoked.

/****************************
Determine operating system (PC or Server)
- For use with PROC FONTREG
*******************************/
%global winsys;
%macro winsys;
  %if &sysscpl = XP_PRO %then %let winsys = WINDOWS;
  %else %if &sysscpl = WIN_SRV %then %let winsys = WINNT;
%mend winsys;
%winsys;
%put "winsys = " &winsys;

/****************************
Register fonts
- WINNT: Server
- WINDOWS: PC
*******************************/
proc fontreg mode=all msglevel=verbose;
  fontfile "C:\&winsys\Fonts\TIMES.TTF";
  fontfile "C:\&winsys\Fonts\ARIAL.TTF";
run;
DATA USED
In this example I am using data from SASHELP.CLASS. This dataset is a class roster that has 19 observations. I am using the gender, height and weight variables from this dataset and will be producing a summary table of height and weight by gender.

Notice in the data step called FINAL that I am adding superscripts (e.g., \((\text{super 2})\)) to the GENDER variable. The `^` symbol will be defined later in the program (right before the ODS RTF statement) as an escape character to allow the insertion of RTF commands.

Notice also that in the data step called FINAL that I am creating a variable called ORD_NUM. The specific purpose of this variable is for use with the COMPUTER AFTER ORD_NUM in PROC REPORT. A blank line will be inserted after GENDER = 'M' and a text string will be inserted after GENDER = 'T'.

```
/*-----------------------------------------------------------*/
/* Read Data                                               */
/*-----------------------------------------------------------*/
data class;
    set sashelp.class;
run;
ods output Summary = total;
proc means data=class n mean std;
    var height weight;
run;
ods output Summary = means;
proc means data=class n mean std;
    class sex;
    var height weight;
run;

data females males;
    set means;
    if sex = 'F' then output females;
    else if sex = 'M' then output males;
run;

data final(rename=(height_n=ht_n height_mean=ht_mean height_stddev=ht_std
        weight_n=wt_n weight_mean=wt_mean weight_stddev=wt_std));
    set females males total(in=t);
    if t then sex = 'T';
    attrib gender length=$20;
    if sex = 'F' then do;
        ord_num = 1;
        gender = trim(left(sex)) || " ^{\text{super 2}}";
    end;
    else if sex = 'M' then do;
        ord_num = 1;
        gender = trim(left(sex)) || " ^{\text{super 3}}";
    end;
    else if sex = 'T' then do;
        ord_num = 2;
        gender = trim(left(sex)) || " ^{\text{super 4}}";
    end;
run;
proc sort data=final;
    by ord_num sex;
run;
```
ODS RTF

In this example, I am first closing the listing destination so that the report will not appear in the .lst file. Next I am defining the "^" symbol as an escape character that allows the insertion of RTF commands in SAS. Then I issue the ODS RTF command that tells SAS to create an .rtf file in the specified location. Notice also that I am using the style definitions (style=print.chgRTF) from PROC TEMPLATE and the BODYTITLE statement so that the titles and footnotes will stay with the body of the table (i.e., they will not be put into the header and footer of the Word table.

TITLES AND FOOTNOTES

Notice that on the title and footnote statements that I can specify font information. For the titles, I am specifying that I want TimesNewRoman, 12 pt and bold and for the footnotes I am specifying that I want TimesNewRoman, 9 pt and italicized.

Notice also that for the footnotes that I am inserting (using the escape character "^" to indent the footnotes under the table and to add superscripts. For the indentation the "S=(indent=1.0)" is a style attribute (S=style) and the indentation is from the margins set up by PROC TEMPLATE, i.e., the indentation begins at 1.25 inches (the left margin).

Clear all Title and Footnote statements

Include titles and footnotes as applicable

/** Add font information directly on titles / footnotes **/
title1 bold height=12pt f='Times' "Table 1: Template"
title2 bold height=12pt f='Times' "Test"

/** Note use of {super } for superscripts **/
/** Note use of {indent } for indenting the footnotes **/
/** (so that they line-up under the table) **/
footnote1 italic height=9pt f='Times' "\'^S={indent=1.0in}\'
"\'^{super 1}\' Height (in)"

/** Blank superscript for footnote2 is solely for the purpose **/
/** of correctly lining up footnote3 with footnote 2 **/
Now I use PROC REPORT to generate the report. First, I set up the general report style. For example, I center the report and then set up the style attributes for lines, column headers and text in the report. Then I define the columns in the report in the COLUMNS statement. Notice that I can insert (using the escape character "^") superscripts in the column headers (see {super 1}) in the COLUMNS statement.

On the DEFINE statements I am using RTF code to align the text in the cells. This RTF code is particularly useful for decimal alignment of text. This RTF code uses a term called twips which is 1/20 of a point. There are 1440 twips per inch. So, to align the text, you first protectspecialchars=off to all insertion of RTF code using a PRETEXT statement. The PRETEXT statement allows insertion of RTF code directly into each cell in the column. To align text the RTF code is \qj\qdec\tx\(N\). The \qj justifies the text in the cell, the \qdec is a decimal tab and \tx (where \(N\) is the number of twips) is the tab position in twips from the left margin. For example, \qdec\tx500 would begin the decimal tab 500/1440 of inch from the left margin of the cell.

In the compute block, I am using another method to insert RTF code. To use this method you will need to use protectspecialchars=off with the compute block statement. Everything after the slash "/" is RTF code. In this case "qc" centers the text. The text string that you want to print must be contained within brackets "{ }." In this example, I am concatenating three smaller strings into one string within the brackets: 1) "Dataset = ", 2) trim(left("&class")), and 3) "^{super 5}".

```plaintext
proc report nowd data=final split='*' style(report)={just=center}
    /** Set fonts according to medical writing standards **/
    style(lines)=header{font_size=10pt font_face="Arial"
        font_weight=medium background=white just=left}
    /** Header: column header **/
    style(header)=header{font_size=10pt font_face="Arial"
        font_weight=bold background=white}
    /** Column: text in columns **/
    style(column)=header{font_size=10pt font_face="Arial"
        font_weight=medium background=white just=center};

    /** Note use of {super } for superscripting **/
    columns (ord_num sex gender
        ("Body Measurements^{super 1}"
         ("Height" ht_n ht_mean ht_std)
         ("Weight" wt_n wt_mean wt_std)));
    define ord_num / noprint " " order order=data;
    define sex / noprint " " order order=data;
```
1. Width does not work with ODS RTF, use cellwidth instead  
2. protectspecialchars=off allows insertion of RTF commands  
   pretext puts the RTF code in each cell in the column  
   \qj justifies text  
   \tqdec is decimal tab  
   \txN is the tab position in twips from the left margin  

define gender  / display "Gender" flow  
   style(header)={just=center cellwidth=0.75 in}  
   style(column)={protectspecialchars=off  
                  pretext="\qj\tqdec\tx500 "  
                  cellwidth=0.75 in};

define ht_n    / display "N" flow  
   style(header)={just=center cellwidth=0.5 in}  
   style(column)={protectspecialchars=off  
                  pretext="\qj\tqdec\tx350 "  
                  cellwidth=0.5 in};

define ht_mean / display "Mean" flow  
   style(header)={just=center cellwidth=0.75 in}  
   style(column)={protectspecialchars=off  
                  pretext="\qj\tqdec\tx500 "  
                  cellwidth=0.75 in};

define ht_std  / display "Std Dev" flow  
   style(header)={just=center cellwidth=0.75 in}  
   style(column)={protectspecialchars=off  
                  pretext="\qj\tqdec\tx500 "  
                  cellwidth=0.75 in};

define wt_n    / display "N" flow  
   style(header)={just=center cellwidth=0.5 in}  
   style(column)={protectspecialchars=off  
                  pretext="\qj\tqdec\tx350 "  
                  cellwidth=0.5 in};

define wt_mean / display "Mean" flow  
   style(header)={just=center cellwidth=0.75 in}  
   style(column)={protectspecialchars=off  
                  pretext="\qj\tqdec\tx500 "  
                  cellwidth=0.75 in};

define wt_std  / display "Std Dev" flow  
   style(header)={just=center cellwidth=0.75 in}  
   style(column)={protectspecialchars=off  
                  pretext="\qj\tqdec\tx500 "  
                  cellwidth=0.75 in};

/** protectspecialchars=off allows insertion of RTF commands after the **/  
/** \qc centers the text **/  
/** \qc centers the text in the brackets {} **/  
/** inserts a blank line after gender = M **/  
/** inserts text string after gender = T **/  

   if ord_num = 1 then text = "\qc{                                    }
   else if ord_num = 2 then text = "\qc{Dataset = " || trim(left(&class)) ||  
                                  " ^{super 5})"};

   line text $;
endcomp;
run;
ods rtf close;
ods listing;
OUTPUT
Here is the Word table that is produced by this SAS program:

Table 1: Template
Test

<table>
<thead>
<tr>
<th>Gender</th>
<th>N</th>
<th>Height</th>
<th>Mean</th>
<th>Std Dev</th>
<th>N</th>
<th>Weight</th>
<th>Mean</th>
<th>Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>9</td>
<td>60.59</td>
<td>5.02</td>
<td></td>
<td>9</td>
<td>90.11</td>
<td>19.38</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>10</td>
<td>63.91</td>
<td>4.94</td>
<td></td>
<td>10</td>
<td>108.95</td>
<td>22.73</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>19</td>
<td>62.34</td>
<td>5.13</td>
<td></td>
<td>19</td>
<td>100.03</td>
<td>22.77</td>
<td></td>
</tr>
</tbody>
</table>

Dataset = SASHELP.CLASS

1 Height (in)
2 Weight (lb)
3 F = Female
4 M = Male
5 T = Total
6 Dataset Name

This table meets the specifications mentioned at the beginning of the paper:

- The table title is centered above the table using TimesNewRoman, Bold, 12pt
- Column headers in the table are Arial, Bold, 10 pt
- The text in the table is:
  - Arial (not bold), 10 pt
  - Centered (where applicable)
  - 3pt spacing before and after text
- The footnotes are:
  - Left justified using TimesNewRoman, Italic, 9pt
  - Paragraph spacing of 0.5pt before & after footnotes

VARIATION TO SPECIFICATIONS
The following changes were made by our Medical Writing group after this paper was started:

- Text in the table:
  - Arial (not bold), 9pt (instead of 10 pt)
  - 2pt (instead of 3pt) spacing before and after text
- Footnotes:
  - Left justified using Arial (instead of TimesNewRoman), Regular (not Italic), 9pt
  - Inside a box below the report
All other specifications remained the same. These changes were made by:

- Changing the cellpadding in the PROC TEMPLATE to 2pt.

```plaintext
style table from table /
  cellpadding = 2pt;
```

- Changing the font_size in the PROC REPORT to 9pt. In the previous table, I specified a STYLE statement for lines, but did not use lines in the PROC REPORT. In this table the specification of the STYLE for lines is important because of moving the footnotes to a COMPUTE BLOCK (see below).

```plaintext
/** lines: lines in compute block **/
style(lines)=header{font_size=9pt font_face="Arial"
  font_weight=medium background=white just=left}

/** header: column header **/
style(header)=header{font_size=9pt font_face="Arial"
  font_weight=bold background=white}

/** column: text in columns **/
style(column)=header{font_size=9pt font_face="Arial"
  font_weight=medium background=white just=center};
```

- Moving the footnotes to a COMPUTE BLOCK in PROC REPORT. This COMPUTE BLOCK is inserted after the other COMPUTE BLOCK in the presented in the previous PROC REPORT section. The font for these footnotes are controlled by the above STYLE(LINES) statement.

```plaintext
compute after _page_ / style=[protectspecialchars=off];
  /** Note use of {super } for superscripts **/
  line "^\{super 1\} Height (in)"

  /** Blank superscript is solely for the purpose **/
  /** of correctly lining up then footnote **/
  line "^\{super \} Weight (lb)"
  line "^\{super 2\} F = Female"
  line "^\{super 3\} M = Male"
  line "^\{super 4\} T = Total"
  line "^\{super 5\} Dataset Name"
endcomp;
```

The following table is the result of these changes:
**Table 2: Template**

**Test**

<table>
<thead>
<tr>
<th>Gender</th>
<th>N</th>
<th>Mean</th>
<th>Std Dev</th>
<th>N</th>
<th>Mean</th>
<th>Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>9</td>
<td>60.59</td>
<td>5.02</td>
<td>9</td>
<td>90.11</td>
<td>19.38</td>
</tr>
<tr>
<td>M</td>
<td>10</td>
<td>63.91</td>
<td>4.94</td>
<td>10</td>
<td>108.95</td>
<td>22.73</td>
</tr>
<tr>
<td>T</td>
<td>19</td>
<td>62.34</td>
<td>5.13</td>
<td>19</td>
<td>100.03</td>
<td>22.77</td>
</tr>
</tbody>
</table>

Dataset = SASHELP.CLASS

---

**CONCLUSION**

Creating Word tables using PROC REPORT and ODS RTF has been made even easier using SAS 9.1.3. Specifically, the use of the BODYTITLE option and PROC FONTREG has made this task easier. In SAS 8.2, you could not use both the NODATE and NONUMBER options with the BODYTITLE OPTION. This problem has been fixed in SAS 9.1.3. Also, in SAS 8.2, you could not have registered fonts and used them with TITLE and FOOTNOTE statements. Now in SAS 9.1.3 this is possible using the new procedure called FONTREG.

I this example, I have also demonstrated how to create a Word table that conforms to specific standards such as margins, fonts for title, footnotes and text of the table and spacing around the text in the table. Additionally, I have demonstrated how to indent footnotes and add superscripts to the table. I have also shown how to do variations on this table.

**REFERENCES**


**ACKNOWLEDGMENTS**

I would like to thank my fellow SAS programmers at work for their input.

**RECOMMENDED READING**


CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author at:
Carey G. Smoak
Roche Molecular Systems, Inc.
4300 Hacienda Drive
Pleasanton, CA 94588
Tel. 925-730-8033
Fax 925-225-0195
carey.smook@roche.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.