

Creating the Perfect Table Using ODS to PDF in SAS 9.4®

Elizabeth Dennis, EMB Statistical Solutions, Overland Park, KS

Maddy Dennis, Independent Contractor, Lawrence, KS

ABSTRACT

Submitting output to the FDA in PDF format has become more common recently. Unfortunately, when RTF files are electronically converted to PDF, unwanted format changes can occur, such as border lines no longer being visible. Creating the table using ODS to PDF directly is a better technique. However, PROC REPORT statements written to create RTF tables produce different results when creating a PDF file. Using SAS 9.4®, this paper will discuss the ODS to PDF statement along with the PROC REPORT statements which will create a perfectly formatted table that conforms to the FDA Portable Document Format Specifications.

INTRODUCTION

It has become more common for publishing groups to request table output in the form of Portable Document Format (PDF) to submit to the FDA. One way to meet this request is to electronically convert a table in Rich Text Format (RTF) to PDF. This technique can result in unwanted format changes, the most common being lines that are no longer visible. These format changes make this conversion process a poor solution.

Many programmers are familiar with the PROC REPORT techniques that produce a table using ODS to RTF. However, these identical statements won't produce identical tables using ODS to PDF; different syntax must be used. For example, the ODS escape character "@" will not work with {last page} for PDF. Also, underlining in PDF requires a different technique than in an RTF.

FDA REQUIREMENTS

The first concern when creating a PDF file is to ensure that it meets the FDA specifications. These are laid out in the Portable Document Format (PDF) Specifications, and the most recent version was published by the FDA September 2014. This applies to both CDER and CBER. This technical document states that the PDF must be in version 1.4 through 1.7, with no security, proper page orientation, and margins of at least ¼ inch. It further specifies that, when the file contains no bookmarks, the Navigation Tab should be set to "Page Only" layout. Standard, suggested fonts are listed, and font size should range from 9 to 12 points. The entire specifications are here:

<http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM163565.pdf>

The appendix to this paper contains the complete code, including PROC TEMPLATE code, for creating a PDF that meets these requirements.

OPTIONS FOR TABLE LAYOUT

SPANNING COLUMN HEADER

Cycle 1 Treatment		
Drug A (N=86)	Drug B (N=96)	Drug C (N=72)

DISPLAY 1. Example of Spanning Column Header

Tables often require a header that spans multiple columns. While producing the column header is standard, underlining it requires that the bottom border line of the cell be visible:

```
Column ("~S={borderbottomcolor=black borderbottomwidth=.5}Cycle 1 Treatment" coll-  
col3);
```

FORCING NEW LINES AND BLANK SPACES

One technique to force blank space or a line break is:

```
ods escapechar='~';  
coll1='~{nbspace 4}Male';  
coll2=AESTDTC || '~{newline 1} || AEENDTC
```

An advantage to using {nbspace} and {newline} is that it is more human-readable than other methods. It's also simpler to insert multiple spaces or lines with a number, rather than repeated commands, such as {~n~n~n} for multiple lines.

PRINTING A BLANK LINE AS THE FIRST FOOTNOTE

The following code will print a solid line as the first footnote. The length of the line will span the entire width of the output.

```
footnote "~{style [outputwidth=100% bordertopcolor=black Bordertopwidth=.1pt]}";
```

PRINTING UNICODE CHARACTERS

Unicode characters print as expected.

```
title5 j=c "Subjects ~{unicode 2265}50 Years of Age";
```

PRINTING SUBSCRIPT AND SUPERSCRIPT

There are a few techniques available to print superscript and subscript characters. A few common ones exist as unicodes, so those could be used. Also, ~{super} and ~{sub} will work, but the characters are printed abnormally small. A third option is to offset the printing of characters, slightly above or below the print area, using a font that is slightly reduced from the normal size.

The following code will add a superscript “a” to the label Age with a font size of 7pt:

```
coll1='Age~-14y~{style[font_size=7pt]a}';
```

This will include a subscript “max” to the label T, again using the font size of 7 pt:

```
coll1='t~7y~{style[font_size=7pt]max}';
```

DECIMAL ALIGNMENT

A difficult issue is decimal alignment. Ideally, the alignment method will work on a table that contains both categorical data (to be summarized by count and percentage), and continuous variable (to be summarized by descriptive statistics). Even more difficult is when a category has counts of zero, and the percentage will not be displayed. In order to make all three scenarios align properly, two separate columns should be used. The first column will contain the descriptive statistics and the categorical counts, and the second column will contain the categorical percentages. When the columns are displayed with zero spacing between them (cellpadding=0 in the template), they appear as one continuous column, and all the decimals are aligned.

Parameter	Drug A (N=86)	
Race - n (%)		
American Indian or Alaska Native	85	(98.8)
Asian	1	(1.2)
Black or African American	0	
Age (years)		
n	86	
Mean	75.2	
Standard Deviation	8.59	
Min	52	
Max	89	

DISPLAY 2. Example of Decimal Alignment

CONCLUSION

Creating PDF output requires different statements than other output types. Using the correct techniques will result in a clear and well-formatted table.

CONTACT INFORMATION

The authors can be contacted at:

Name: Elizabeth Dennis
Enterprise: EMB Statistical Solutions, LLC
Address: 9300 West 110th Street, Suite 550
City, State ZIP: Overland Park, KS 66210
Work Phone: 913-322-6555
E-mail: edennis@embstats.com

Name: Maddy Dennis
Enterprise: Independent Contractor
Address: 1036 New Jersey Street
City, State ZIP: Lawrence, KS 66046
Work Phone: 785-218-7533
E-mail: madison.kansas@gmail.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.

Fake Data, Inc.
Study 123

Table 1.4 Parameter Summaries - Safety Population
Subjects ≥50 Years of Age

Parameter	Overall Treatment		
	Drug A (N=86)	Drug B (N=96)	Drug C (N=72)
Sex - n (%)			
Male	83 (96.5)	41 (42.7)	37 (51.4)
Female	3 (3.5)	55 (57.3)	35 (48.6)
Race - n (%)			
American Indian or Alaska Native	86 (100.0)	0	1 (1.4)
Asian	0	6 (6.3)	9 (12.5)
Black or African American	0	90 (93.8)	62 (86.1)
Age ^a			
n	86	96	72
Mean	75.2	76.4	73.8
Standard Deviation	8.59	8.11	7.94
Min	52	51	56
Max	89	88	101
AUC ₀₋₂₄			
n	86	96	72
Mean	5.22	7.64	73.28
Standard Deviation	8.591	8.111	7.914
Min	5.2	51.7	56.4
Max	8.9	88.3	101.8

^a Age in years.

Source: C:\Fake Data\Study 123\fig\table\spang.sas

27MAR15/15:44

APPENDIX: COMPLETE CODE

```

*****;
** define the style of output;
*****;

ods path (prepend) work.templat(update);

proc template;
  define style testrtf;

    parent = styles.rtf;

    replace Table from Output/
      frame = above
      rules = groups
      cellpadding = 0pt
      cellspacing = 0pt;

    replace color_list/
      'link' = blue
      'fg' = black
      'bg' = _undef_
      'bgH' = white ;

    replace fonts/
      'TitleFont2' = ("Times New Roman",9pt)
      'TitleFont' = ("Times New Roman",9pt)
      'StrongFont' = ("Times New Roman",9pt)
      'EmphasisFont' = ("Times New Roman",9pt)
      'FixedEmphasisFont' = ("Times New Roman, Times New Roman",9pt)
      'FixedStrongFont' = ("Times New Roman, Times New Roman",9pt)
      'FixedHeadingFont' = ("Times New Roman, Times New Roman",9pt)
      'BatchFixedFont' = ("Times New Roman, Times New Roman, Times New Roman",9pt)
      'FixedFont' = ("Times New Roman, Times New Roman",9pt)
      'headingEmphasisFont' = ("Times New Roman",9pt)
      'headingFont' = ("Times New Roman",9pt)
      'docFont' = ("Times New Roman",9pt);

    replace Body /
      rightmargin = 1 in
      leftmargin = 1 in
      bottommargin = 1 in
      topmargin = 1 in;

  end;
run;

*****;
** Create fake data;
*****;

data out;
  row=11;  c1=83;    c2=41;    c3=37;    output;
  row=12;  c1=3;     c2=55;    c3=35;    output;
  row=21;  c1=86;    c2=0;     c3=1;     output;
  row=22;  c1=0;     c2=6;     c3=9;     output;
  row=23;  c1=0;     c2=90;    c3=62;    output;
  row=31;  c1=86;    c2=96;    c3=72;    output;
  row=32;  c1=75.2;  c2=76.4;  c3=73.8;  output;
  row=33;  c1=8.59;  c2=8.11;  c3=7.94;  output;
  row=34;  c1=52;     c2=51;    c3=56;    output;
  row=35;  c1=89;     c2=88;    c3=101;   output;

```

Creating the Perfect Table Using ODS to PDF in SAS 9.4®, continued

```

row=41;  c1=86;    c2=96;    c3=72;    output;
row=42;  c1=5.22; c2=7.64;   c3=73.28; output;
row=43;  c1=8.591; c2=8.111;  c3=7.914; output;
row=44;  c1=5.2;   c2=51.7;   c3=56.4;  output;
row=45;  c1=8.9;   c2=88.3;   c3=101.8; output;
run;

*****;
** get N for column headers and denominators;
*****;

%let N1=86;
%let N2=96;
%let N3=72;

*****;
** format output for table;
*****;

data out;
  set out;
  length coll-col7 $100;

  if row=11 then coll=~{nbspace 4}Male';
  else if row=12 then coll=~{nbspace 4}Female';
  else if row=21 then coll=~{nbspace 4}American Indian or Alaska Native';
  else if row=22 then coll=~{nbspace 4}Asian';
  else if row=23 then coll=~{nbspace 4}Black or African American';
  else if row=24 then coll=~{nbspace 4}White';
  else if row in (31 41) then coll=~{nbspace 4}n';
  else if row in (32 42) then coll=~{nbspace 4}Mean';
  else if row in (33 43) then coll=~{nbspace 4}Standard Deviation';
  else if row in (34 44) then coll=~{nbspace 4}Min';
  else if row in (35 45) then coll=~{nbspace 4}Max';

  if 11 <= row <=24 then do;
    col2=put(c1,4.);
    if c1 > 0 then col3=" (" || put(round(c1/&n1*100,.1),5.1) || ")";
    col4=put(c2,4.);
    if c2 > 0 then col5=" (" || put(round(c2/&n2*100,.1),5.1) || ")";
    col6=put(c3,4.);
    if c3 > 0 then col7=" (" || put(round(c3/&n3*100,.1),5.1) || ")";
  end;
  else do;
    col2=put(c1,best.);
    col4=put(c2,best.);
    col6=put(c3,best.);
  end;

  keep coll-col7 row ;
run;

data headline;
  length coll $100;
  row=10;
  coll='Sex - n (%)';
  output;
  row=20;
  coll='Race - n (%)';
  output;
  row=30;
  coll='Age~-12y~{style[font_size=7pt]a}';

```

Creating the Perfect Table Using ODS to PDF in SAS 9.4®, continued

```

output;
row=40;
coll='AUC~7y~{style[font_size=7pt]0-24}';
output;
run;

data out;
  set out headline;
  if row in (10 11 12) then ord=1;
  else if row < 30 then ord=2;
  else if row < 40 then ord=3;
  else ord=4;
run;

proc sort;
  by ord row;
run;

*****;
*** print output;
*****;

options orientation=landscape nonumber nodate missing=' ' ;
ods listing close;
ods escapechar='~';

ods pdf file="C:\clinlib\CDISC\report\tlg\tables\psug.pdf" style=testtrf notoc
nobookmarkgen uniform;

title1 j=1 "Fake Data, Inc." j=r 'Page ~{thispage} of ~{lastpage}';
title2 j=1 'Study 123';
title3 j=c ;
title4 j=c "Table 1.4 Parameter Summaries - Safety Population";
title5 j=c "Subjects ~{unicode 2265}50 Years of Age";

footnote1 j=1 "~{style [outputwidth=100% bordertopcolor=black Bordertopwidth=.1pt]}";
footnote2 j=1 "~-12y~{style[font_size=7pt]a}~12yAge in years.";
footnote3 ;
footnote4 j=1 "Source: C:\Fake Data\Study 123\tlg\tables\psug.sas~{nbspace
150}&sysdate./&systemtime";

proc report data=out nowindows headskip missing split="" formchar(2)="_" spanrows;
  column ord row coll ("~S={borderbottomcolor=black borderbottomwidth=.5}Cycle 1
Treatment" col2-col7) ;

  define ord      / order order=internal noprint;
  define row      / order order=internal noprint;

  define coll     / display "Parameter"
                  style(header)=[just=left]
                  style(column)=[just=left cellwidth=50%];
  define col2     / display "Drug\ (N="
                  style(header)=[just=right]
                  style(column)=[just=d cellwidth=8% ];
  define col3     / display "~{nbspace 1}A\&n1"
                  style(header)=[just=left]
                  style(column)=[just=d cellwidth=8% rightmargin=.3in];
  define col4     / display "Drug\ (N="
                  style(header)=[just=right]
                  style(column)=[just=d cellwidth=8% ];
  define col5     / display "~{nbspace 1}B\&n2"
                  style(header)=[just=left]

```

Creating the Perfect Table Using ODS to PDF in SAS 9.4®, continued

```

        style(column)=[just=d cellwidth=8% rightmargin=.3in];
define col6      / display "Drug\ (N="
                  style(header)=[just=right]
                  style(column)=[just=d cellwidth=8% ];
define col7      / display "~{nbspspace 1}C\&n3)"
                  style(header)=[just=left]
                  style(column)=[just=d cellwidth=8% rightmargin=.3in];

compute before ord;
  line @1 ' ';
endcomp;

run;

run;

ods pdf close;
ods listing;
proc printto;
run;

*****;
** End of Program
*****;
```