

Paper GH-10

ODS PDF and RTF Application Development

Benno Kurch, Trading and Software Development, Inc., Merrifield, VA
Shirish Nalavade, eClinical Solutions, Inc., Mansfield, MA

ABSTRACT

For an ODS developer at a beginning or advanced level, creation of tailor-made ODS PDF or RTF reports is not an out-of-the-box, turnkey operation. Creation of such reports may require creative thinking and much experimentation. Implementing a larger font size, for example when converting a line-printer ("legacy") report using a monospace font to an ODS report may require report re-design and consultation with business review personnel. This paper will present some challenges encountered in a recent upgrade from legacy reports to more refined, formatted PDF and RTF output, and the recommended solutions to those challenges. Taking time up front to think through potential issues can significantly reduce development time and facilitate the production of properly designed ODS reports that will impress users with their professional look.

INTRODUCTION

Our initial project mandate was to convert 17 pharmaceutical reporting applications producing about 100 legacy reports to ODS PDF and RTF reports. The conversion effort required developing SAS ODS reporting code with UNIX SAS 9.2. The PDF reports represented the initial stage of producing reports for FDA submission. The RTF reports were primarily for our Medical Writing group. The legacy report applications called post-processing programs to manipulate reports that PROC REPORT or PROC TABULATE generated. The post-processing programs added page X of Y information, section headers or desired blank rows.

We elected to produce the RTF reports using ODS measured RTF (also known as the RTF tagset). We just used the ODS TAGSETS.RTF statement and PROC TEMPLATE to produce RTFs. We did not make changes to the default tagset since there was a substantial learning curve involved. For the prototype reports we came close to producing the reports as we expected to see them. However, we could not figure out how to prevent footnotes from going to the next page. We were not knowledgeable of the RTF language so we did not attempt to post-process the RTF. SAS Institute, Inc.® sent us tagsets.rtf code to correct the problem. This code contains many other useful features and provides a great basis for tagsets.rtf development. You can access this code by clicking on the Downloads tab from this SAS support link: <https://support.sas.com/kb/40/327.html>

Our experience with producing a measured RTF report illustrates a classic problem of underestimation. Since we could produce a prototype quickly with the new technology, we did not realize how much time and knowledge it would take to properly implement the whole new methodology. Minimal examples at the SAS web site and on the Web in general compounded the difficulty (Tagsets RTF had just been introduced into SAS when we began using that tagset). By using the ODS TAGSETS RTF statement without tagsets code edits and with few changes to your style template using PROC TEMPLATE, you can produce an almost adequate report. However, because of the inherent complexity of RTF reports, use of tagsets code and a thorough understanding of the RTF language is a requirement to produce a finely tuned report that will meet your users' professional standards. Tagsets RTF is powerful software that promises to make measured RTF creation even easier since SAS Institute is putting more resources into development of the measured RTF destination. Several papers listed in the References section make it easy to understand how to use and create tagsets code.

THE MOST IMPORTANT CONSIDERATIONS WHEN CONVERTING A LEGACY REPORTING SYSTEM TO AN ODS PDF OR RTF BASED SYSTEM

For many companies the reports being converted may have been created 5 years ago or more. The business group probably will be excited when they see a prototypical ODS PDF or RTF report since the report gives them the slick, professional look they are seeking. The business group informed about ODS capability will

realize that maintenance of programs post-processing REPORT output is history since, for one thing, ODS will add Page X of Y information to PDF (and RTF) reports.

Not aware of the potential problems involved in the light year jump technology, the business group may give the green light to convert the reports. This may occur before any consideration is given to major report redesign. Before embarking on the re-programming step, the ODS reports developer needs to stop and alert the business group of the probable need to re-design reports.

Font changes may force the re-design of a whole report. For example, a lab report may have taken up the whole width of the report page when it was printed in 8 point Courier, a small fixed width font. If the report is converted to 10 point Arial, a common proportional font, the report will no longer fit across the page and may overflow to the next page or be truncated. Your users want to display as many columns of information on the page as possible so comparison is optimized. One way to approach the re-design if a row often extends to multiple rows is to combine columns. For example, age, race and sex may occupy 3 columns. You may want to combine age, race and sex into one column (so if age has a value of 46, race white and sex male, you could represent the column label as Age/Race/Sex and the values as 46/White/Male). If enough values for other columns other than Age, Race and Sex extend to multiple rows then you may want to represent the information vertically (For the label, i.e., you might want Age/ on the first line, Race/ on the second line and Sex on the third line).

Legacy reports often have characters all with the same size, a fixed line size and fixed page size. Post-processing programs leverage this functionality to make decisions about keeping a group of information on the same page. ODS PDF reports typically employ proportional fonts. These reports have infinite line size and page size. Breaking to a new page is a logical and no longer physical consideration. To illustrate, let's say you have a group of System Organ Class records that you want to keep together on a page. For a legacy report you can determine how many lines are left for display on the page and how many records are in your group. If there are more records in the group then can fit on the page, your post-processing program forces a page eject, and prints the "System Organ Class" sectional header line (a literal not in a GROUP variable). With ODS PDF your easiest option is to create a page break variable. However, by doing this you most likely will have report pages with lots of wasted space. ODS PDF doesn't have the capability to measure line height size. Developing an algorithm to do this would be time consuming.

The developer armed with knowledge should request a meeting with all parties involved in using the reports. Because programming ODS reports can be labor intensive, a meeting is crucial at the start of the process. The developer needs to ask: "Do the current legacy reports optimally advance business needs? If not, how should the reports be changed to do this?". If reports have not been upgraded for 5 years or more, then most likely the user community will embrace the opportunity to re-vamp the reports. We found in developing a large-scale ODS application that there were many report items users wanted to change. The developer needs to acquire a clear understanding of what the user community desires before thinking about ODS programming details.

When you elicit user feedback about reporting requirements you may find that the report conversion task has become more complicated. Once you start intensively converting and you run into programming issues you may find the task becomes even more challenging. Scope creep, the developer's nemesis, constantly intrudes. To combat this you need to establish regular communication with your business group and to establish limits. Should a change, for example, go in our first version or should it be delayed to a subsequent production version? We recommend weekly technical meetings with representatives from the business and programming groups and general business group meetings for review of reports. Your company may even want to implement ODS reporting in the background of some structured development methodology such as Rational Unified Process (RUP), for example. When formulating possible solutions before engaging in a business group discussion, the developer needs to become as informed as possible about existing or potential problems. The developer can do this by talking to other developers who have already built an ODS system, reading some of the excellent SAS books written about ODS and by looking at SAS blogs.

REPORTING SOFTWARE TO IMPLEMENT FOR CONVERSION OF MULTIPLE REPORTING TYPES: REPORT, TABULATE AND DATA_NULL_:

If legacy programs are written primarily with DATA _null_ logic, converting to the use of PROC REPORT and/or TABULATE will greatly increase your ability to improve the style of your report output. The learning curve for each of these procs is large, so training time for each needs to be factored into your decision. Also you need

to factor in more time for the ODS portion of the conversion since ODS for REPORT and TABULATE sometimes involve different ODS coding techniques.

At the start of our project we decided to convert our TABULATE code to REPORT code. Doing so took less effort than you might think, just a few hours or a day to convert a report. With business approval we re-designed the reports so they would be easier to produce with REPORT (less TABULATE centric). The REPORT COMPUTE block allowed us to produce reports with more of the tailoring that DATA _NULL_ provides. While some programmers and users preferred TABULATE or DATA _NULL_ to REPORT, we found that long-term maintenance and consistency across reporting platforms were simplified by using only REPORT. We strongly suggest selecting one reporting type for all the reports, to reduce re-programming difficulties.

ESTABLISHING THE TEMPLATE (FOUNDATION) FOR ODS PDF DEVELOPMENT

PROC TEMPLATE can appear intimidating at first. You see that the default template style Styles.rtf doesn't give you the report you're looking for. However, we quickly found template examples at the SAS support site and on the web for ODS PDF. After some experimentation we came up with template code that required no modification as we progressed forward.

```
ods path work.templates(update) sashelp.tmplmst(read) sasuser.templat(update);

proc template;
  define style yourstyl / store=templates;
    parent = Styles.rtf;
  class fonts /
    'TitleFont'           = ("Times New Roman, Times Roman, Times" ,10pt,Bold )
    'TitleFont2'          = ("Times New Roman, Times Roman, Times" ,10pt,Bold )
    'footFont'            = ("Times New Roman, Times Roman, Times" ,10pt,Bold )
    'StrongFont'          = ("Times New Roman, Times Roman, Times" ,10pt,Bold)
    'EmphasisFont'        = ("Times New Roman, Times Roman, Times" ,10pt,Italic)
    'headingEmphasisFont' = ("Times New Roman, Times Roman, Times" ,10pt,Bold)
    'headingFont'         = ("Times New Roman, Times Roman, Times" ,10pt,Bold)
    'docFont'             = ("Times New Roman, Times Roman, Times" ,10pt)
    'FixedEmphasisFont'   = ("Courier" ,9pt,italic)
    'FixedStrongFont'     = ("Courier" ,9pt,Bold)
    'FixedHeadingFont'    = ("Courier" ,9pt,Bold)
    'BatchFixedFont'      = ("Courier" ,9pt)
    'FixedFont'           = ("Courier" ,9pt)
  ;
  class color_list /
    'link' = blue
    'bgH'  = white
    'bgT'  = white
    'bgD'  = white
    'fg'   = black
    'bg'   = white
  ;
  style Table from Output /
    rules          = all
    cellpadding    = 4px
    cellspacing    = 0
    bordercolor    = lightgrey
  ;
  class systemtitle /
    protectspecialchars=OFF
    asis=ON
  ;
  class systemfooter /
    font=Fonts('footFont')
    protectspecialchars=OFF
    asis=ON
  ;
end;
```

```

class header /
  protectspecialchars=off
;
class data /
  protectspecialchars=off
;
class rowheader /
  protectspecialchars=off
;
class usertext /
  protectspecialchars=off
;
class byline /
  protectspecialchars=off
;
end;
run;

```

Depending on your business preferences you will need to do some tailoring of the template code. You may want to: Change the type size and font for the various elements, place more padding around the text of the cell, and place more space between cells. We built the templates dynamically; we added macro code around the template code, for example, to allow users at run-time to produce a template with no grid lines within the table body (rules=groups). Your site may want to create permanent templates if your reports never have grid lines within the table body or you never want to change the bolding attribute.

This information from the SAS on-line documentation for BASE SAS: ODS fonts is very useful:

The fonts associated with the following three strings are the most commonly modified:

The string 'DocFont' controls the fonts for the majority of the output, such as the data values and the items in the table of contents.

The string 'headingFont' controls the fonts for the headers.

The string 'TitleFont' controls the fonts for the titles and footnotes.

Regarding style color_list:

'fg' = black is specified. In the parent style Styles.RTF 'fg' = black is also specified.

If 'fg' = black removed from color_list warning message will appear in log,
i.e., Warning: Could not locate style reference 'std.colors("bylinefg")'.

'fg' - color specified affects color of text in cells throughout the table.

'bg' - color specified affects background color of entire table except for title, footer and column header areas.

'bgH' - color specified just affects background color of header area.

You will also need to set your margins. We chose margins suitable for a pharmaceutical application:

```

options
  topmargin      = 0.75in
  bottommargin  = 0.75in
  leftmargin    = 0.75in
  rightmargin   = 0.75in;

```

You can also do this in the template:

```

Style body from Document /
  topmargin      = 0.75in
  bottommargin  = 0.75in
  leftmargin    = 0.75in
  rightmargin   = 0.75in

```

PRODUCING YOUR PDF

Example code:

```
ods listing close;
options pdfcopy pdfaccess pdfassembly pdfcomment nopdfcontent;

ods pdf file = "./yourfile.pdf"
      compress = 6
      newfile    = NONE
      pdftoc     = 1
      startpage  = YES
      style      = yourstyle
      uniform
      nobookmarkgen
;

```

Finally, after adding your PROC REPORT code, issue the command:

```
ods PDF close;
```

LINE SUPPRESSION BETWEEN REPORT CELLS

Your client may request that you suppress the grid line between the count and percent columns:

n	(%)	n	(%)
1186	93	973	80
223	18	214	18
2	0		
1	0		
1	0	2	0

Figure 1: PDF output with the gridline between n and %.

You may think when specifying RULES=ALL that you can set the BORDERRIGHTWIDTH=0 for the N column and set BORDERLEFTWIDTH=0 for the percent column. You will find that setting these values for PDF or RTF does not work; the line between N and percent still appears. You can white out the line with BORDERRIGHTCOLOR=WHITE or BORDERLEFTCOLOR=WHITE. That may work depending on your version of Adobe Acrobat or the template you are using. To remove the gap we recommend adding a BORDERTOPWIDTH= and BORDERBOTTOMWIDTH= specification:

```
define count      / analysis sum "Count"

  %if ( &produce_grid_lines= 1 ) %then %do; /* count - grid lines */

    %if ( &produce_percentages = 1 ) %then %do; /* percentages */

      style(column)={just=center width=15mm
                     borderrightcolor=white
                     bordertopwidth=1px
                     borderbottomwidth=1px
                     }
      style(header)={just=center}

    %end; /* percentages */
  %end;

```

```

%else %do; /* no percentages */

        style(column)={just= center  width=15mm}
        style(header)={just=center}

%end; /* no percentages */

%end; /*count - grid lines */

%else %do; /* count - no grid lines */

        style(column)={just=center  width=15mm}
        style(header)={just=center}

%end; /* count - no grid lines */
;

%if ( &produce_percentages = 1 ) %then %do; /* display percentages */

define _pct_      /  "Percentages"

        %if ( &produce_grid_lines = 1 ) %then %do; /* percent - grid lines */

                style(column)={just=center  width=15mm
                                borderleftcolor=white
                                bordertopwidth=1px
                                borderbottomwidth=1px
                                }
                style(header)={just=center}

        %end; /* percent - grid lines */

        %else %do; /* percent - no grid lines */

                style(column)={ just=center  width=15mm }
                style(header)={just=center}

        %end; /* percent - no grid lines */
;

%end; /* display percentages */

```

n	(%)	n	(%)
1186	93	973	80
223	18	214	18
2	0		
1	0		

Figure 2: PDF output with no gridline between n and %.

CENTER ALIGNMENT WITHIN A CELL OF DECIMAL VALUES

For legacy reports with fixed character sizes (a non-proportional font) center aligning decimal values is easy. For SAS 9.2 producing ODS PDF files with a proportional font this is not easy. When producing a PDF file SAS software as of 9.2 does not allow you to center a value in a cell and align on the decimal. You can accomplish this to some extent by specifying `style(column)={just=center asis=on}` and assigning a PICTURE format to the value. This technique is not perfect, however, and you will find with different combinations of column numbers that sometimes the alignment will noticeably not appear centered. We don't recommend assigning a PICTURE format. We recommend using `JUST=D` (`ASIS=ON` not necessary) to right align on the decimal.

n	(%)
1186	93.238994
223	17.531447
2	0.1572327
1	0.0786164

Figure 3: PDF output center aligned

If producing an ODS RTF file you can easily center align on the decimal using in-line RTF commands with the 'R' destination code. An example of this code for your REPORT DEFINE statement:

```
define _count_ / "Count"
               Style(column) = [pretext='^R"\ql\tqdec\tx250 "' ];
define _pct_   / "Percentages"
               Style(column) = [pretext='^R"\ql\tqdec\tx250 "' ];
```

The 'ql' means to left justify, 'tqdec' set the decimal tab and 'tx250' set the space for the decimal tab at 250 twips. A twip is 1/1440 of an inch.

Part of the output file is shown below

223	17.53	214	17.53
2	0.16	.	.
1	0.08	.	.
1	0.08	2	0.16
.	.	1	0.08

Figure 4: RTF output center aligned

PRODUCING ONE PDF BOOKMARK NODE WITH NO SUB-NODES

To suppress sub-nodes there are some tacks you can take (reference SAS support problem note 31278: <http://support.sas.com/kb/31/278.html>). 3 nodes appear by default. The ODS PROCLABEL statement controls the production of the first (primary) node. By default the value for this node is "The Report Procedure". Via the PROCLABEL statement you replace this value with your own such as "Table 1". To suppress the production of the 2nd node (a sub-node), code the `CONTENTS=` option with no space between the quotes on your REPORT statement:

```
Proc report data=x nowd list contents=''
```

To suppress the 3rd node, also a sub-node, you have to counteract a SAS change to REPORT's table of contents entry in SAS 9.2: You have to create a dummy variable, reference it within REPORT and specify the `CONTENTS=` option for the variable within the BREAK statement:

```
Data x;
  Set x;
  ... ;
```

```

dummy_var = 1;
run;

proc report data=x nowd list contents='';
  columns dummy_var ... ;
  define dummy_var / group noprint;
  break before dummy_var / contents='' page;

```

Even after you implement this code, sub-nodes will appear in your Table of Contents (bookmarks) if your PROC REPORT code contains a BY statement.

You may realize that PROC DOCUMENT can be used to remove/manipulate sub-nodes as well. This solution will work but becomes a bit complicated when a page by report involved or you are stacking output from multiple procedures (REPORT, FREQ, for example) in one PDF. A solution that avoids the use of DOCUMENT and is quite simple is the following code. The solution involves creating a dummy report page with no visible character (we tested this code extensively and found no problem):

```

ods pdf file="./example.pdf"
      startpage=YES nobookmarkgen;

proc report data=x nowd;
  column a;
  define a ...
run;
ods pdf bookmarkgen startpage=no;

data test;
  x="00"x;
  output;
run;
options label;
ods proclabel = "Table X";

title;
footnote;

proc report data=test nowd contents=''
      style={font_size=.1pt foreground=white cellpadding=0
      rules=none frame=void}
      ;
  column x ;

  define x / group noprint
      style={font_size=.1pt foreground=white cellpadding=0
      rules=none frame=void};
      break before x / contents='' page;
run;
ods pdf close;

```

Thank you to Bari Lawhorn of SAS Institute, Inc. for providing this very creative solution!

USING '@' AS THE ESCAPECHAR WITH LASTPAGE WILL NOT WORK IN PDF OUTPUT

For our legacy reports, post processing programs added Page X of Y information. To create ODS PDF files, in our reporting programs we produced this information using the THISPAGE and LASTPAGE functions. However, while experimenting with various escape characters, we realized that the LASTPAGE function does not work when '@' is used as the ODS escape character. Please refer to below link for usage note and the possible workaround. <http://support.sas.com/kb/38/614.html>

INSERTING BREAKS IN TITLES AND FOOTNOTES

Sometimes we needed to add more titles or footnotes than the limit of 10 SAS allows. With ODS, by using line breaks it is easy to have more than 10 titles or footnotes. For more information on how to achieve this, please refer to the following paper from Cynthia Zender:

<http://www2.sas.com/proceedings/forum2007/099-2007.pdf>

RECOMMENDED SAS SOFTWARE UPGRADE

More than likely you may find that the solution to a problem may involve a hot fix or application of a maintenance pack. You may find as we often did SAS support recommending this. You may also come across a new problem that you bring to SAS support's attention. Before undertaking the ODS project, if feasible, upgrade your software to the latest maintenance pack and apply all relevant hot fixes. By doing so you will greatly facilitate your conversion effort.

CONCLUSION

Foremost you want to remember that converting your legacy reports to PDF and/or RTF reports presents a marvelous opportunity to advance the goals of the business. Enlist the support of your user community and put in a lot of effort in creating reports that package data presentation most efficiently. Establish report consistency and minimize your programming effort by building reports based on report templates whenever possible. To further reduce such effort settle on one reporting type of software such as PROC REPORT. Give your development group an advantage and access to more capability by upgrading to the latest SAS release. Spend a lot of time visiting the SAS support and blog sites, and researching on the Web so you can keep abreast of recommended solutions. There is a vast amount of information on the Web about the powerful ODS software and more appears every day. Remember countless others have gone down the same road you are on. Remember if you are stymied and the formulation of your solution is taking too long, contact a fellow developer or give SAS Support a call.

ACKNOWLEDGMENTS

Many thanks to Bari Lawhorn at SAS Institute, Inc. for the many discussions we had with her, all the numerous solutions she provided us, and her editing of this paper. Many thanks to Al Pasquino at Pfizer, Inc. for all his recommendations in producing optimal re-designed reports, for his assistance with analysis and design and for his programming suggestions.

REFERENCES

Gebhart, Eric. 2007. "Paper 225-007: ODS Markup, Tagsets, and Styles! Taming ODS Styles and Tagsets" Proceedings of the SAS Global Forum 2007 Conference.

Available at: http://support.sas.com/rnd/base/ods/odsmarkup/Paper_225-2007_tagsets_styles.pdf

Haworth, Lauren. 2005 "132-30: SAS with Style: Creating Your Own ODS Style Template for ODS PDF Output" Proceedings of the SAS Global Forum 2005 Conference.

Available at: <http://www2.sas.com/proceedings/sugi30/132-30.pdf>

Lawhorn, Bari. 2011. "Let's Give 'Em Something to TOC about: Transforming the Table of Contents of Your PDF File", Proceedings of the SAS Global Forum 2011 Conference

Available at: <http://support.sas.com/resources/papers/proceedings11/252-2011.pdf>

Lawhorn, Bari and Huntley, Scott. 2010. "035-2010: Getting the Right Report (Again): Your Compatibility Guide for ODS PDF 9.2", Proceedings of the SAS Global Forum 2010 Conference.

Available at: <http://support.sas.com/resources/papers/proceedings10/035-2010.pdf>

Zender, Cynthia. 2007. "Funny ^Stuff~ in My Code: Using ODS ESCAPECHAR". Technical Papers and Presentations. Cary, NC:SAS Institute Inc.

Available at <http://www2.sas.com/proceedings/forum2007/099-2007.pdf>

Zender, Cynthia. 2007. "Paper 2603-29 Markup 101: Markup Basics" , SAS Institute, Inc., Cary, NC
Proceedings of the SAS Global Forum 2010 Conference.

Available at: <http://support.sas.com/rnd/papers/sugi29/markup-basics.pdf>

Carpenter, Arthur L., 2007, Carpenter's Complete Guide to the SAS REPORT Procedure, Cary, NC: SAS
Institute Inc. Available at: <https://support.sas.com/pubscat/bookdetails.jsp?pc=60966>

SAS Institute Inc. 2009. SAS® 9.2 Output Delivery System: User's Guide. Cary, NC: SAS Institute Inc.
Available at support.sas.com/documentation/cdl/en/odsug/61723/PDF/default/odsug.pdf.

SAS Discussion Forums <http://communities.sas.com/index.jspx>

CONTACT INFORMATION

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

Benno Kurch
Trading and Software Development, Inc.
P.O. Box 868
Merrifield, Va. 22116-0868
Email: bennfran17@yahoo.com

Shirish Nalavade
eClinical Solutions, A Division of Eliassen Group
603 West St
Mansfield, MA 02048
Work Phone: 508-594-6337
E-mail: snalavade@eclinicalsol.com or at shirish.nalavade@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.