How to Add a Little Spice to Your PDF Output

ABSTRACT

The ODS PRINTER statement has enabled SAS users to create PDF output from their SAS jobs since Release 8.2 of SAS. Now, with SAS 9.2, PDF output has more spice. Topics in this paper include how to apply new styles, the new SAS/GRAPH interaction, how to scale background images, how to use new fonts and Unicode characters, and much more, including a brief discussion of ODS layout. Tips show you how easy it is to add some spice to your PDF output.

INTRODUCTION

Adding spice to an existing recipe can make it better. Adding some spice to your ODS PRINTER output can help you not only demonstrate results, but also dazzle your audience. ODS PRINTER in SAS 9.2 adds some serious spice to your output. The ODS PRINTER statement (which includes the Printer, PCL, PostScript, and PDF destinations) is always trying to stay on top of changes and requests that users want. By staying on top, ODS can provide you the best possible output.

This paper highlights several new features that have been implemented in SAS 9.2. It explains the new features and how best to use them. It provides examples and code that you can use.

SPICE UP YOUR APPEARANCE WITH STYLES AND NEW FUNCTIONS

INLINE STYLES

Styles enable SAS programmers to control the overall appearance of ODS tables and graphs. Using PROC TEMPLATE is one of the best ways to use styles in your ODS output. Creating or using a style template supplied by SAS is a great way to control the overall look. Sometimes you need to use a style just once in your output, and you don't want it to affect your entire output in the process. This is when using inline styles is the best option. Using the ODS ESCAPECHAR statement and the inline syntax, you can modify the styles of TITLE or TEXT statements. Using the SAS 9.2 inline syntax enables you to set several style attributes for one text string, without having to reset the previously used style. Let's look at the inline syntax and how it works:

```
^{style <style-element-name><[style-attribute-specification(s)]> formatted text}
```

The ^ symbol is defined when you issue an ODS ESCAPECHAR statement at the beginning of your SAS program. After the escape character, begin the syntax with the left curly bracket {. Then, use the STYLE function. Next, you can use a style element like SYSTEMTITLE, HEADERFIXED, SYSTEMFOOTER, and so on. A style element is a collection of style attributes that apply to a feature or aspect of the output. Values are specified for each attribute in a style definition. After the style element, you can set the style attributes. Some common style attributes are color, font, and size. You can set as many style attributes as needed within square brackets. The syntax ends with the text you want to be formatted.

The example shows how to use the SAS 9.2 inline syntax:

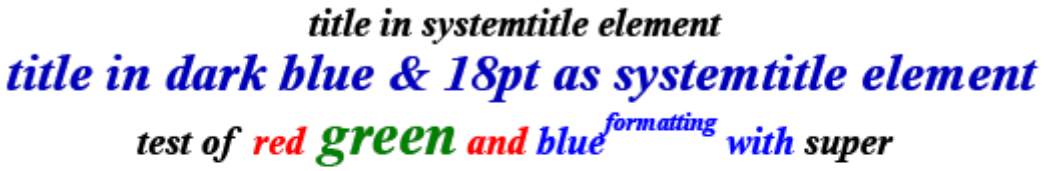
```
options nodate nonumber;
ods escapechar="^" ;
ods pdf file="test.pdf";

title1 "^{style systemtitle title in systemtitle element}";
title2 "^{style systemtitle[color = dark blue fontsize=18pt] title in dark blue
& 18pt as systemtitle element}";

title3 "test of ^{style [color=red] red ^{style [color=green fontsize=20pt]
green} and ^{style [color=blue] blue ^{super formatting} with}} super" ;

proc print data=sashelp.class(obs=1) ; run;
ods _all_ close ;
```

Here is the output:



Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5

In the code, three TITLE statements are set. The first TITLE statement creates a title using the SYSTEMTITLE style element. The second TITLE statement creates a title using the SYSTEMTITLE style element and sets two style attributes. The title is dark blue with an 18-point font size. The third TITLE statement shows off the power of inline syntax. This syntax allows nesting or combining styles within the same statement. Look at the TITLE3 line. The color changes three times and a superscript function is used. Any style declared within another style inherits the traits of the outer style. When a current style declaration ends using a right curly bracket, the style reverts to the previously defined style, or the default style, if none is available.

Inline styles can be used in a DATA step. You use the same syntax as you use in TITLE or TEXT statements. This example shows the inline syntax with the DATA _NULL_ step:

```
options ls=130 nonumber nodate nocenter;
ods pdf file='temp.pdf';
title;
ods escapechar='^';

data _null_;
  file print;
  put 'text with ^{style [foreground=red]red} color';
  put 'text with ^{style [font_size=18pt color=green font_weight=bold]color and}
font size and weight';
  put 'text with ^{style [font_face=arial font_style=italic font_size=20pt
color=blue]blue and} font style';
run;
ods pdf close;
```

Here is the output:

You can control the border of each cell using a new style element. Every cell has four borders: top, bottom, left, and right. Using the STYLE function and the new border style elements, you can assign a color or width

to each border. The example uses PROC REPORT and the new border style elements. By using an IF statement in the code, you can highlight a cell:

```
options nodate nonumber;
ods listing close;
ods escapechar="*" ;
ods pdf file="borders.pdf" ;

title "*{style [borderleftwidth=2pt borderleftcolor=red borderbottomwidth=2pt
borderbottomcolor=blue] Border Control w/Height over 63 } ";

proc report data=sashelp.class(obs=5) nowd;
  column name age height=height2 height;
  define Name / "Name";
  define age / "Age";
  define height2 / noprint;
  define height / "Height";
  compute height;
     if height2 > 63 then
     call define(_col_, "style", "style=[borderleftcolor=violet
borderleftwidth=2pt borderrightcolor=rose borderrightwidth=2pt
bordertopcolor=lime bordertopwidth=2pt borderbottomcolor=gold
borderbottomwidth=2pt]");
  endcomp;
run;
ods _all_ close;
```

Here is the output:

Name	Age	Height
Alfred	14	69
Alice	13	56.5
Barbara	13	65.3
Carol	14	62.8
Henry	14	63.5

INLINE FUNCTIONS

In SAS 9.2, new functions give you more options in your output. The syntax for these functions is the same for all functions.

^{*<function name> <function parameter>*}

The first two new functions are NBSPACE and NEWLINE. NBSPACE enables you to create a number of nonbreaking spaces before the next text character. NEWLINE creates new blank lines before the next text character. The example uses both of these functions:

```
options nodate nonumber;
ods escapechar="^";
ods pdf file='test.pdf';
title 'test of nbspace^{nbspace 5}now test newline ^{newline 3} okay all done';

proc print data=sashelp.class (obs=1);run;
ods pdf close;
```

Here is the output:

test of nbspace now test newline

okay all done

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5

You can see the use of the two functions in the TITLE statement. The function NBSPACE has a parameter of 5, and there are five nonbreaking spaces between the words “nbspace” and “now.” The function NEWLINE has a parameter of 3, and there are three line breaks before the phrase “okay all done.”

Another new function is TEXTDECORATION. This function enables you to overline, underline, or strike-through your title or footnote. The example shows each:

Here is an overlined title.

Here is a underline title.

Here is a title with a line through it.

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5

A picture speaks a thousand words in this example. You can see how each title using the TEXTDECORATION function appears in the SAS output.

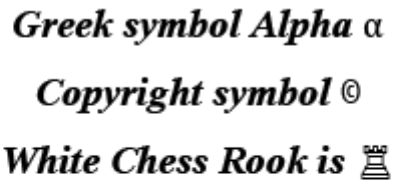
The last new function is UNICODE. This function enables you to designate Unicode symbols in output. You can set the parameter to be either a 4-digit Unicode value, or a value from a predefined list stored as a tagset template. The example uses the UNICODE function:

```
options nonumber nodate;
ods escapechar='^';
ods pdf file="test.pdf" notoc;
```

```
title  'Greek symbol Alpha ^{unicode alpha}';
title2 'Copyright symbol ^{unicode 00A9}';
title3 'White Chess Rook is ^{unicode 2656}';

proc print data=sashelp.class(obs=1); run;
ods pdf close;
```

Here is the output:



Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5

The first TITLE statement uses the keyword “Alpha” from the tag set. The next two TITLE statements use the 4-digit Unicode values to designate what characters to use. To find out what Unicode values are available, you can look at a character map of the font. The character map contains all the glyphs for that font. Each glyph has a unique 4-digit Unicode value. To access the character map in Microsoft Windows XP, click **Start→Programs→Accessories→System Tools→Character Map**.

See the SAS 9.2 documentation for other inline functions.

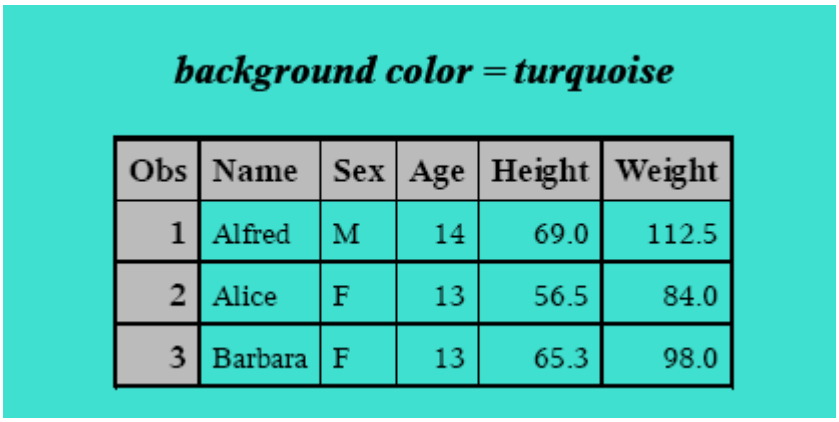
SPICE UP YOUR APPEARANCE WITH BACKGROUND COLOR AND IMAGES

BACKGROUND COLORS

You can customize the background of a document. Output can look powerful with color or a background image. The example uses PROC TEMPLATE and adds a background color to the entire page:

```
proc template;
   define style bkgdcolor;
   parent=Styles.Printer;
      class Body
         "Controls the Body file." /
              backgroundcolor=turquoise;
         end;
              run;
title 'background color = turquoise';
ods pdf file="backgnd_color.pdf" style=bkgdcolor;
proc print data=sashelp.class(obs=3);run;
ods _all_ close;
```

Here is the output:



background color = turquoise

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0

Using PROC TEMPLATE, you can define a new style for the document. You set the BACKGROUNDCOLOR element in the BODY section of the template. Setting the color here covers the entire page. It's important to choose the correct color so that you can still read your output.

BACKGROUND IMAGES

Another quick way to add some spice to your output is with a background image. Think about how a watermark or a company logo would look in your output. The background image appears on every page of your output. It's important that you create a proper size of the image so that it won't interfere with your output. You might have to use an image editor to get the size you need. ODS PRINTER accepts many different kinds of image formats. Some accepted formats are JPG, GIF, PNG, BMP, and SVG.

It is easy to create a simple PROC TEMPLATE that instructs PDF to use an image in the background of your tables. You use the BODY section of the template to set the background image. In the example, the word "DRAFT" is used as a background image in a draft copy of a proposal. The image text is muted so that it does not overshadow the data. The image has the same background color as the table. If the background color of the image were different, you would have to change the table's background color to be consistent.

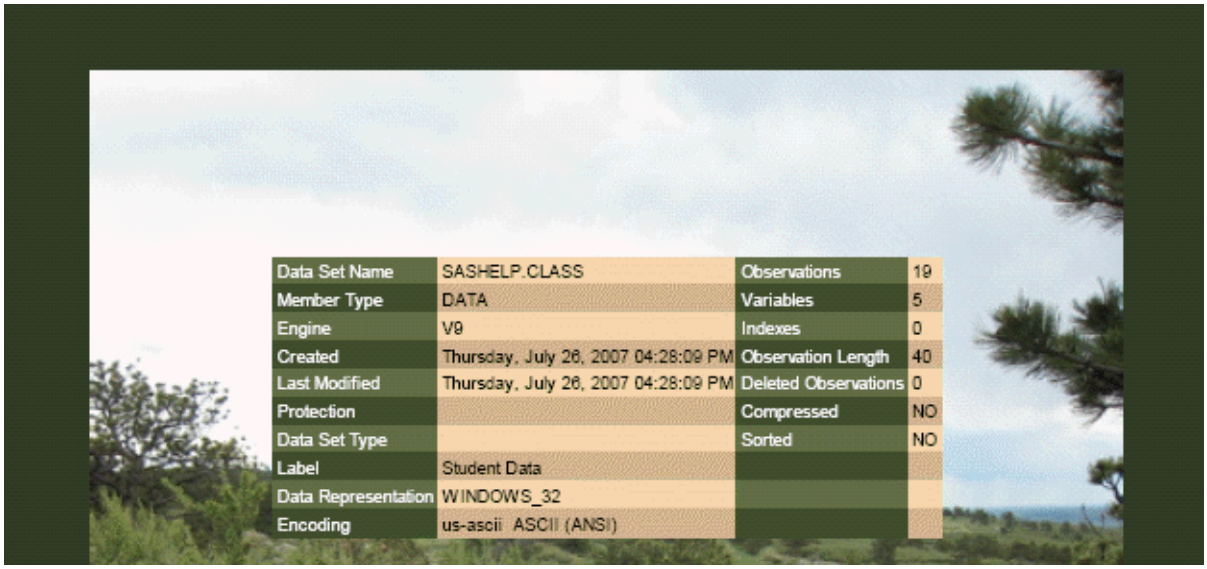
```
proc template;
   define style bkgdimage;
   parent=Styles.Printer;
      class Body
         "Controls the Body file." /
               backgroundimage='u:\pp\draft.jpg';
         end;
               run;

ods pdf file="backgnd_large.pdf" style=bkgdimage;
proc print data=sashelp.class;run;
ods _all_ close;
```

Here is the output:

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84.0
10	John	M	12	59.0	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90.0
13	Louise	F	12	56.3	77.0
14	Mary	F	15	66.5	112.0
15	Philip	M	16	72.0	150.0
16	Robert	M	12	64.8	128.0
17	Ronald	M	15	67.0	133.0
18	Thomas	M	11	57.5	85.0
19	William	M	15	66.5	112.0

If you want to get really fancy, you can mix colors and set margins and get some incredible output. This example uses PROC TEMPLATE again. Using several options like background colors, images, and margins, the output looks incredibly fancy. The code for this output is at the end of the paper. Here is a small snapshot of the output:



Data Set Name	SASHELP.CLASS	Observations	19
Member Type	DATA	Variables	5
Engine	V9	Indexes	0
Created	Thursday, July 26, 2007 04:28:09 PM	Observation Length	40
Last Modified	Thursday, July 26, 2007 04:28:09 PM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label	Student Data		
Data Representation	WINDOWS_32		
Encoding	us-ascii ASCII (ANSI)		

SPICE UP YOUR OUTPUT WITH PDF OPTIONS

There are several new global options to use with your PDF output. Global options apply to an entire SAS invocation, not just a specific ODS statement. Using global options can save time when you want all of your files in a SAS job to have the same set of options applied to them. These global options are valid in configuration files, on SAS invocation, in OPTIONS statements, or in the SAS System Options window.

PDF SECURITY

The new PDFSECURITY option helps you control the level of PDF encryption on your document. It has three possible values: NONE (default), LOW, and HIGH. When you set it to HIGH, it uses a 128-bit encryption on the document. When you set it to LOW, it uses 40-bit encryption. Encryption requires Acrobat 5.0 or later.

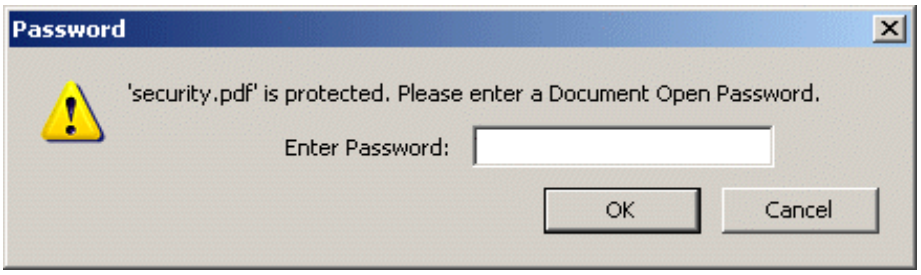
The new PDFPASSWORD (or its alias PDFPW) option sets one of two new passwords on your PDF document. You can set an Owner password, a User password, or both. An Owner password allows you to set or change permissions for a document. You can restrict what a user can perform on your document such as printing or editing. A User password allows you to decide who can open your document. Any attempt to open your document is prompted by a dialog box asking for the User password.

You do not need to set both passwords, but if you do, you need to remember that the Owner and User passwords cannot be the same. If you use the same password, you get a warning message in your log file. Also, if you set the option PDFSECURITY to HIGH or LOW, you must also set PDFPW with at least one password to create a file. You get a warning message in your log file if you forget to set a password. If you set PDFPW, and then set PDFSECURITY=NONE, you get an output file, but the password settings are ignored.

The code uses security and passwords:

```
options pdfsecurity=high pdfpw=(owner="owner" open="open");
ods pdf file="security.pdf";
proc print data=sashelp.class;run;
ods pdf close;
```

Here is what displays:



Options such as PDFCOPY, PDFPRINT, and PDFCOMMENT enable a user with the Owner password to control features like copying, printing, and seeing comments when viewing the document. These options are Boolean options. To set the PDFCOPY option, you type it on the OPTIONS statement as is. To disable it, you type NO before the option.

PDF VIEWS

Two new options enable you to control the way you view your PDF document. The PDFPAGELAYOUT option controls the page layout. This setting is the equivalent of the **View→Page Display** option in Adobe Acrobat Reader when a document is open. The PDFPAGEVIEW option controls the page viewing mode. This setting is the equivalent of the **View→Zoom** option in Adobe Acrobat Reader.

PDF COMPRESSION

A new global option controls the level of deflate compression of a PDF document. Currently, in the ODS PRINTER statement, you can set the compression level for a PDF document using COMPRESSION=*n* (where *n* is a whole number between 0 and 9). All files that are created in a SAS invocation are deflated with this value, unless the local statement setting is different. Setting this option is exactly the same as setting the compression on an ODS PDF statement. In the ODS PDF statement, set DEFLATION=*n* (or use the alias DEFLATE). The variable *n* is any whole number between 0 and 9. Zero means no deflation, while 9 is the highest deflation possible. The SVG and SAS/GRAPH PNG drivers comply with this option.

The DEFLATION setting interacts with the global option UPRINTCOMPRESSION. UPRINTCOMPRESSION is a master switch that enables or disables compression. If NOUPRINTCOMPRESSION is set, then DEFLATION is ignored.

This is a brief view of some of the new options used by PDF documents. See the SAS 9.2 documentation for other new options and how they can be used.

SPICE UP YOUR OUTPUT WITH NEW URLS

Have you ever wanted to have a link in your PDF send an e-mail or connect to a newsgroup? In SAS 9.2, new URL support does things just like that. In previous releases of SAS, URLs worked only with HTTP, but now support for HTTPS, FTP, NEWS, and NNTP is available.

The following code uses URLs. The default link color is blue, but the link color in the title is changed in the TITLE statement. An inline style changes the link color in the last TEXT statement. You can see in the code how the URL functions FTP, NEWS, and MAILTO are used to create these statements:

```
title "^S={url='ftp://ftp.sas.com/techsup/download/blind/'} Click here to use
FTP";
proc print data=sashelp.class(obs=5);
run;
ods pdf text="^S={url='news:comp.soft-sys.sas'} Search the SAS-L Newsgroup ";
ods pdf text="^S={linkcolor=yellow url='mailto:ods@sas.com'} Questions or
comments??? Click here ";
```

Here is the output:

Click here to use FTP

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5

Search the SAS-L Newsgroup

Questions or comments??? Click here

Clicking on each URL opens a different thing. The FTP link opens a browser in Microsoft Windows pointing to the site designated in the code. The newsgroup link opens a utility like Microsoft Outlook Express that you can use to read newsgroups. The question link opens a new e-mail with the To: address already filled in. Using URLs brings a different ability to your output. Now you can create output that performs actions based on user input.

SPICE UP YOUR OUTPUT WITH NEW GRAPH INTERACTION

SAS/GRAPH has always been accessible within ODS PRINTER, but the integration has not always been the best. In SAS 9.2, there is tighter integration. In the past, when ODS PRINTER encountered a graph in the output, it would break to a new page and put the graph on a page by itself. In SAS 9.2, options have been provided to help with graph interaction. ODS PRINTER accepts the options HSIZE and VSIZE from the GOPTIONS statement. HSIZE and VSIZE control the horizontal and vertical size of the graphics output area. Using these options can shrink your graph to a size that fits within half a page. The example uses these attributes:

```
goptions reset=all hsize=4 vsize=4 border;
  ods pdf file='test.pdf';
title "hsize=vsize=4";
    proc gplot data=sashelp.class;
      plot height*weight;
       run;
    quit;
ods _all_ close;
```

Running the code produces a plot with the exact dimensions of 4 inches by 4 inches on the page. By setting the options HSIZE and VSIZE on the GOPTIONS statement, the plot is downsized while keeping the proper aspect ratio.

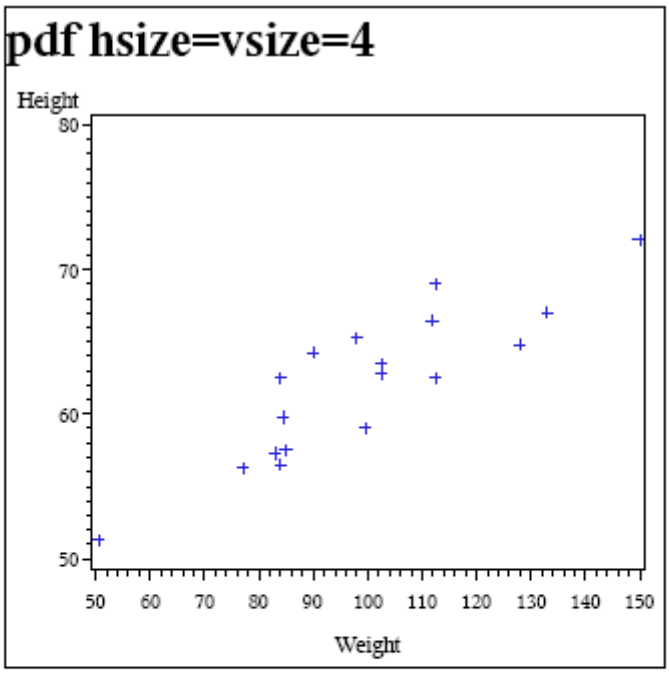
In SAS 9.2, you can have several objects on the same page. Using the ODS attribute STARTPAGE, page breaks before and after graphs can be prevented. But first, a little history. Before SAS 9.2, whenever ODS PRINTER encountered a graph or plot in the output, it would break to a new page. SAS/GRAPH is a completely different code path from ODS PRINTER. In the past, there was no way of determining the size of the graph that was requested. To make sure there was room, a page break was inserted before and after the graph. As time progressed, several SAS programmers wanted to have graphs and tables on the same page. Interaction between ODS PRINTER and SAS/GRAPH was enhanced to pass sizing information back and forth. By setting the graph size and using STARTPAGE=NO, the output is changed dramatically:

```
goptions reset=all hsize=4 vsize=4 border;
ods pdf file='test.pdf' startpage=no;
title1  j=l "pdf hsize=vsize=4";
  proc gplot data=sashelp.class;
    plot height*weight;
     run;
  quit;

proc print data=sashelp.class(obs=5);run;
ods _all_ close;
```

The GOPTIONS statement sets the options needed by PROC GPLOT. The BORDER option sets exactly how big the plot is. In the ODS PDF statement, STARTPAGE=NO tells ODS PRINTER not to break to a new page when processing a graph. The code performs a PROC GPLOT, and then a PROC PRINT. Internally, ODS PRINTER receives the plot first and places it in the output. Because STARTPAGE is set to NO, no page break occurs before the next output object. Then, the print output arrives. ODS PRINTER determines whether any space is left on the page. It determines the space that is needed for the output

table. And, if it fits, it puts it in the output. Otherwise, a page break is inserted and the table appears on the next page. Here is the output:



Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5

SPICE UP YOUR OUTPUT WITH LAYOUT?

Now that I have your attention, ODS LAYOUT will still be preproduction in SAS 9.2. But, its foundation is established. It is more stable and more robust.

So, what do we have in SAS 9.2? Gridded layout tells ODS LAYOUT how many columns you want and layout tries to allocate the space for the objects to fit. In the example, three objects are output. Gridded layout places the first object in the left column. Working left to right, top to bottom, it places the second object in the right column. Then, it places the third object under the first object in the left column. This all depends on the space that is available. If there was a fourth object, it would go under the second object in the right column.

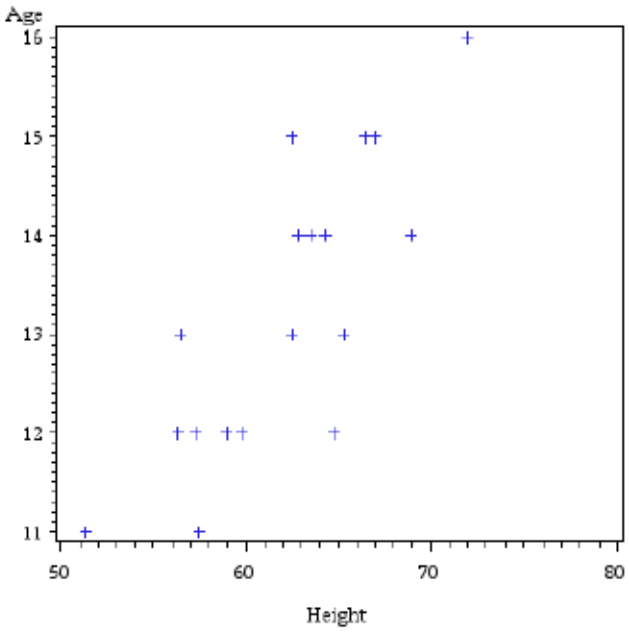
The code uses layout:

```
goptions reset=all hsize=4 vsize=4;
ods pdf file="layout.pdf";
ods layout start columns=2;
proc print data=sashelp.class(obs=5); run;
proc gplot data=sashelp.class;
plot age*height; run;
proc print data=sashelp.air(obs=5); run;
```

```
ods layout end;
ods pdf close;
```

Here is the output:

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5



Obs	DATE	AIR
1	JAN49	112
2	FEB49	118
3	MAR49	132
4	APR49	129
5	MAY49	121

When you work with ODS LAYOUT, it is trial and error. You have to make sure your output fits into the space you think it does. When your data increases, your output objects increase. If my first output object had been more than 5 observations, it could have pushed my third output object onto the next page. ODS LAYOUT is a powerful tool. You just need to make sure your output matches your expectations.

CONCLUSION

So much has been added to ODS PRINTER in SAS 9.2, that it's hard to list it all. Most of the features mentioned in this paper were driven by customers, either through the SASware Ballot or Technical Support requests. Using the spice mentioned in this paper will give your output that special flare. These features will give you more power over the appearance of your output. ODS PRINTER is always looking for new ways to make your output the best it can be and have the spice that makes it stand out.

ACKNOWLEDGMENTS

The author would like to thank Bari Lawhorn, Chevell Parker, Chuck Bass, Scott Singer, Kevin Smith, and Tim Hunter for their contributions to this paper.

CONTACT INFORMATION

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

Scott Huntley
SAS Institute Inc.
R1444 SAS Campus Drive
Cary, NC 27513
scott.huntley@sas.com

CODE

This is the code for the output with the background image and PROC TEMPLATE combined:

```
options nonumber;
ods escapechar='^';
proc template;
define style myprinter; parent=styles.printer;
    class body /
        margintop = 1in
        marginbottom = 1in
        marginleft = 1in
        marginright = 1in
        backgroundcolor = #303B25
        backgroundimage = '\\dntsrc\u\kesmit\pp\test\rabbitmtn.png'
        textalign = right
    ;
    class table /
        cellpadding = 4
        bordercolor = _undef_
        borderwidth = _undef_
        borderstyle = _undef_
        frame = _undef_
        rules = _undef_
             borderspacing=0
    ;
    class header, rowheader /
        fontfamily = arial
        fontsize = 8pt
        backgroundcolor = #3F4D2C
        color = white
        fontweight = medium
    ;
    class data /
        fontfamily = arial
        fontsize = 8pt
        backgroundcolor = #CBAC8F
    ;
    class systemtitle, proctitle /
        fontfamily = arial
        fontweight = bold;
    ;
    class pageno /
        fontfamily = arial
        color = white
        fontweight = medium
    ;
end;
```

```
define table base.template.table;
    cellstyle _style_ like '%Header' and mod(_row_,2) as
{backgroundcolor=cx616D46},
              _style_ like '%Header' as {backgroundcolor=cx3F4D2C},
              mod(_row_,2) as {backgroundcolor=cxF5D8AD},
              1            as {backgroundcolor=cxD5B58D};
end;
run;

options nodate;

ods noproctitle;
ods pdf file="bg.pdf" style=myprinter;

title '^{newline 5}';
proc contents data=sashelp.class;
run;

ods pdf close;
```