

Paper 212-2013

Behind the Scenes with SAS®: Using Customer Graphics in SAS® Output

Louise S. Hadden, Abt Associates Inc., Cambridge, MA

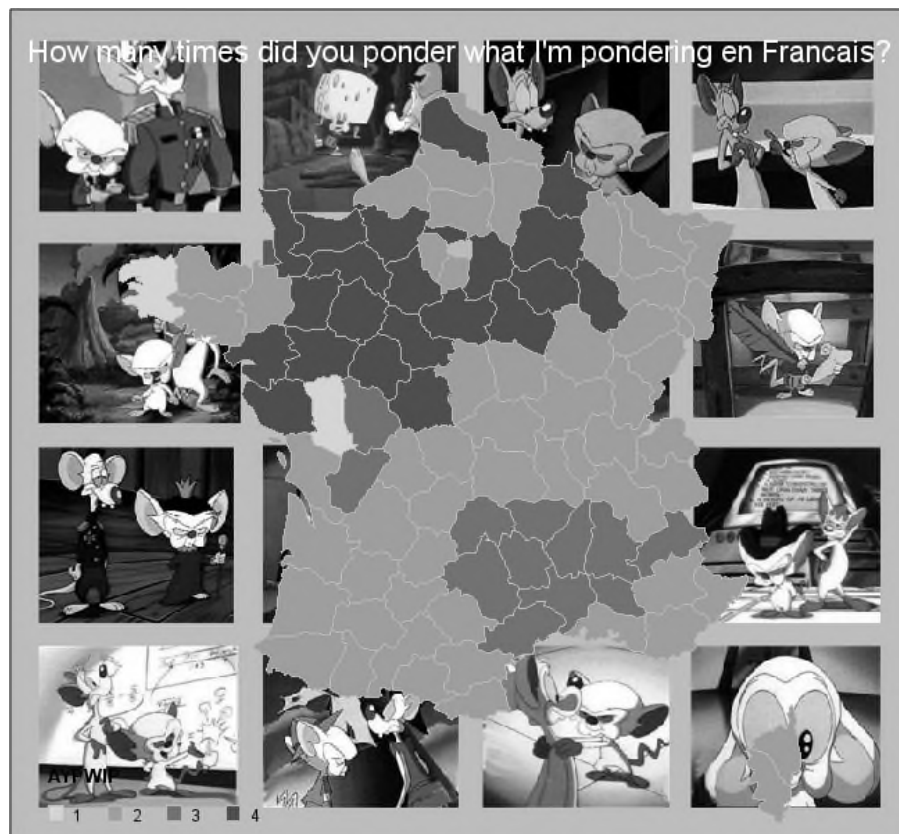
ABSTRACT

SAS® provides many opportunities to add customized images to SAS ODS output. This presentation will demonstrate various ways to add custom backgrounds to tabular and graphic output; add custom images to titles, footnotes, ODS text fields and tabular output; and add custom "fills" to SAS/GRAPH maps and graphs. Some possible uses of custom images include a company logo embedded in SAS output, graphic displays of positive or negative outcomes, and watermarks containing "draft" or "confidential". The SAS code to accomplish all these potential uses, and more, will be shown.

INTRODUCTION

The purpose of this paper is to demonstrate how easy it is to produce reports, images and web pages with custom backgrounds and custom images, using a variety of techniques. Examples are produced using SAS® 9.3 on the Windows 2008R Server platform.

The inspiration for this paper was one of the many samples provided by Robert Allison of SAS®. I cannot say enough about how wonderful, creative and useful these samples are. <http://robslink.com/SAS/Home.htm> While idly browsing his sample library one day, I came across a SAS/GRAPH map produced with a custom, tiled image, clearly a photograph, in the background behind the map. Adapting the code that Robert provides with each of his samples yielded this fun, SAS-created map of France with a custom background of Pinky and the Brain images.



That (eventual) success evolved into the beginning of a fruitful and instructive exploration into using custom images with SAS.

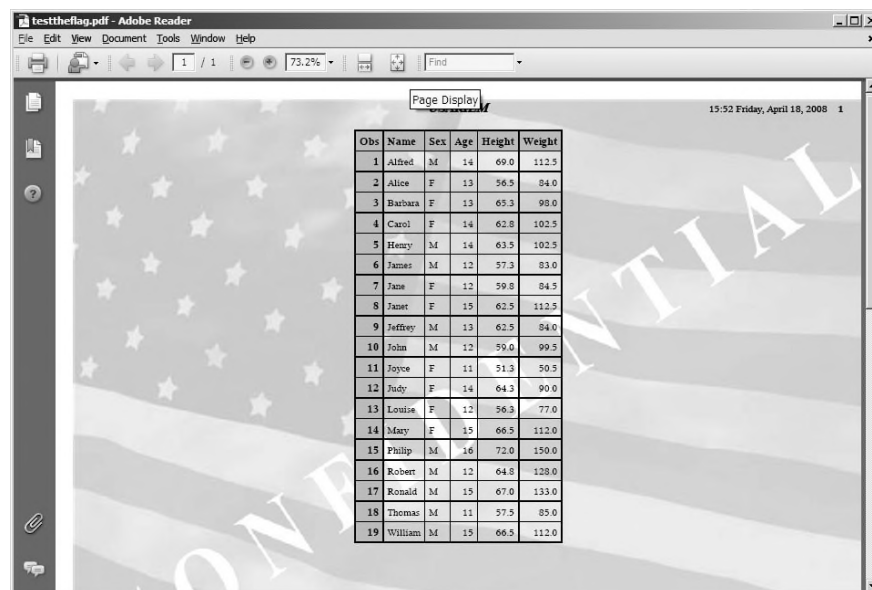
MODIFYING A STYLE TEMPLATE

Possibly the easiest and most efficient way to apply a background to SAS output is by modifying a style template to include a custom background image. Scott Huntley's paper "How to Add a Little Spice to your PDF Output" has an example of a commonly desired use of having "DRAFT" in the background. Since the background image is "stretched" by default, it is important to use an image that more or less "fits" the desired output size and shape. For example, for reports that are a standard landscape page you would want to use an image that is approximately 8 ½ inches high by 11 inches wide. It is also important to consider both the color and the intensity of the background image. A dark background might obscure the text of a report. In addition, some graphics procedures will overlay the background image so that it does not show. In the example presented below, a picture of the American flag was "faded" and the word "CONFIDENTIAL" was overlaid on the image.

There's no reason to reinvent the wheel when creating a style template. SAS provides a number of "canned" style templates which can be used as the "parent" for a new style with a custom background. It is best to use a relatively simple style like minimal or journal when you are adding your own background so that you don't have any conflicting color choices. As you'll see, adding a background image requires very few lines of code.

```
options textureloc='c:\images'; /* points SAS to where image store is located */

proc template;
  define style styles.usariem;
    parent=styles.minimal;
    replace graphbackground /
      background=colors('docbg')
      backgroundimage='rippleflag.jpg'
    ;
  end;
run;
```



GRAPHIC BACKGROUND

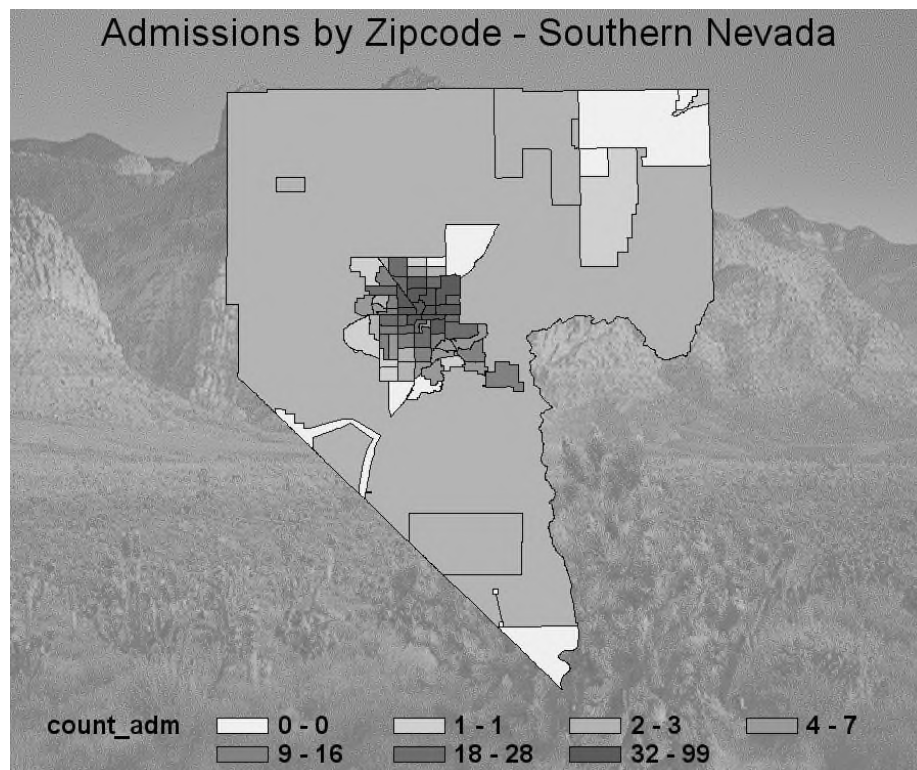
SAS/GRAPH procedures that produce an image, such as GCHART and GMAP, allow the placement of custom images in the graph's background area. The syntax is:

```
options iback='external image file' imagestyle=fit; /* or tile (tile is default)*/
Again, it is good to make sure your images "fit" your graph, so that you don't get poor quality images in your
background. It is possible to tile multiple images in the background by having multiple consecutive goptions
statements with different "iback" references.
```

```
goptions reset=all xpixels=900 ypixels=750 device=jpeg ftext="Arial/bo" cback=vpag;
goptions border iback='canyon1.jpg' imagestyle=fit htitle=2.75 htext=1.3;
ods listing close;
ods html path=odsout body='tryit.htm' ;
```

```
Title h=2 font=Arial bold italic 'Admissions by Zipcode - Southern Nevada';
proc gmap data=clark map=clark ;
  id zip;
  choro count_adm /
    coutline=black
    annotate=labels_clark name="tryit";
run;
quit;
```

```
ods html close;
ods listing;
```



BACKPLANE

Placing an image on a backplane, or “frame”, works with the GCHART, GPLOT, GRADAR, and GSLIDE procedures. The syntax is to place the “iframe” option on the procedure that generates the graph. “iframe” can refer to a fully qualified image name or a file reference (fileref.) The example below shows a fileref “aa” being used. The default is to tile the image to fit the backplane. You must specify “imagestyle=fit” to stretch the image to fit in the backplane.

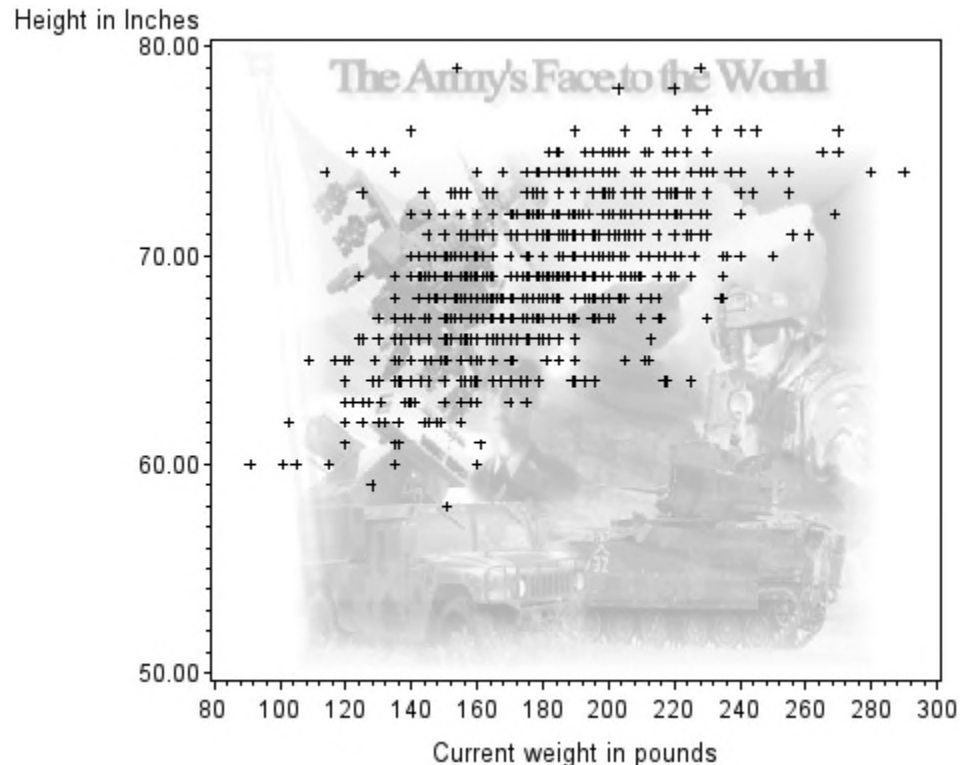
```
ods listing close;

options orientation=landscape;

goptions reset=all hsize=5 vsize=4 transparency border;
ods html file='trylayout.htm' device=png ;

proc gplot data=anal;
  plot v16_a*wt_lbs / iframe=aa imagestyle=fit;
run;

ods _all_ close;
```



BAR CHART / PATTERN STATEMENT

Custom images can be placed on two-dimensional bars of graphs generated by PROC GCHART's VBAR and HBAR statements. They can also be placed on three-dimensional (3D) bar graphs if the device used is ACTIVEEX or ACTXIMG. Images are specified in the PATTERN statement using the IMAGE option. The syntax is as follows:

```
Pattern image=fileref; OR
Pattern image='external-image-file';
```

VERTICAL BAR CHART 2D

```

%let name=tickbar;
filename odsout '.';

/* create response data and link */

data a2;
input year value link $ 10-80;
cards;
1999 7.9 http://www.abtassoc.com
2000 9.3 http://www.abtassoc.com
2001 10 http://www.abtassoc.com
;
run;

data a2; set a2;
length myhtml $ 254;
myhtml=
'title=|||
quote('year: ||trim(left(year))||' ' '|| 'Tick Infestations (thousands):
||trim(left(value))||' ' ')||' ' '||'href='||quote(trim(left(link))||' ' ');
run;

GOPTIONS DEVICE=png;

ODS LISTING CLOSE;
ODS HTML path=odsout body="&name..htm"
(title="Using images in pattern statement")
style=styles.ticks gtitle gfootnote ;
goptions border;

pattern1 v=s image='amerdogtickpair.jpg';
pattern2 v=s image='anibug.gif';
pattern3 v=s image='browndogpair.jpg';

axis2 label=('',) minor=none;
axis1 label=('',);

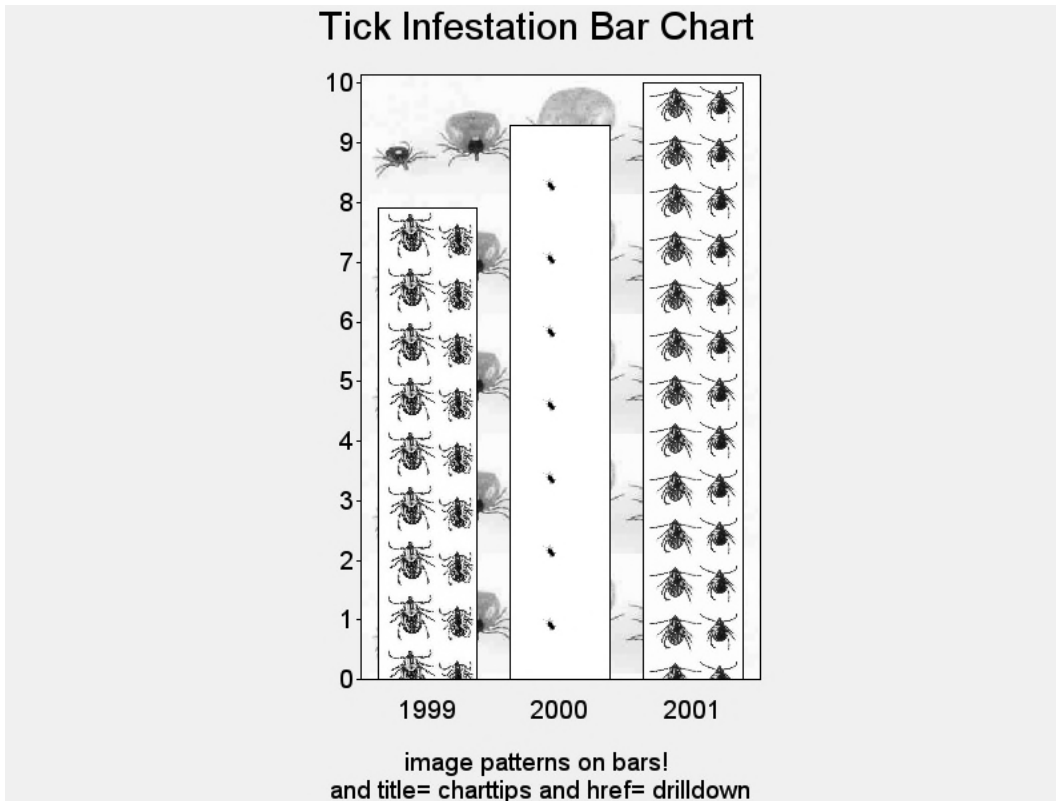
goptions htitle=.3in ftitle="arial" htext=.2in ftext="arial";

title 'Tick Infestation Bar Chart';
footnote 'image patterns on bars!';
footnote2 ' and title= charttips and href= drilldown';

goptions cback=cxf2f2df;
/* NOTE THIS OPTION WILL OVERRIDE THE CUSTOM BACKGROUND IN THE STYLE */
proc gchart data=a2;
vbar year /
maxis=axis1 raxis=axis2 discrete type=sum sumvar=value
subgroup=value coutline='black' width=15 nolegend
ascending space=5 frame iframe='lonestargroup4.jpg' imagestyle=tile
html=myhtml des="" name="&name" ;
run;

quit;
ODS HTML CLOSE;
ODS LISTING;

```



VERTICAL BAR CHART 3D (ONLY ELEMENTS THAT CHANGE SHOWN)

```

GOPTIONS DEVICE=actximg;

ODS LISTING CLOSE;
ODS HTML path=odsout body="&name..htm"
(title="Using images in pattern statement")
style=styles.ticks
gtitle gfootnote
;
goptions border;

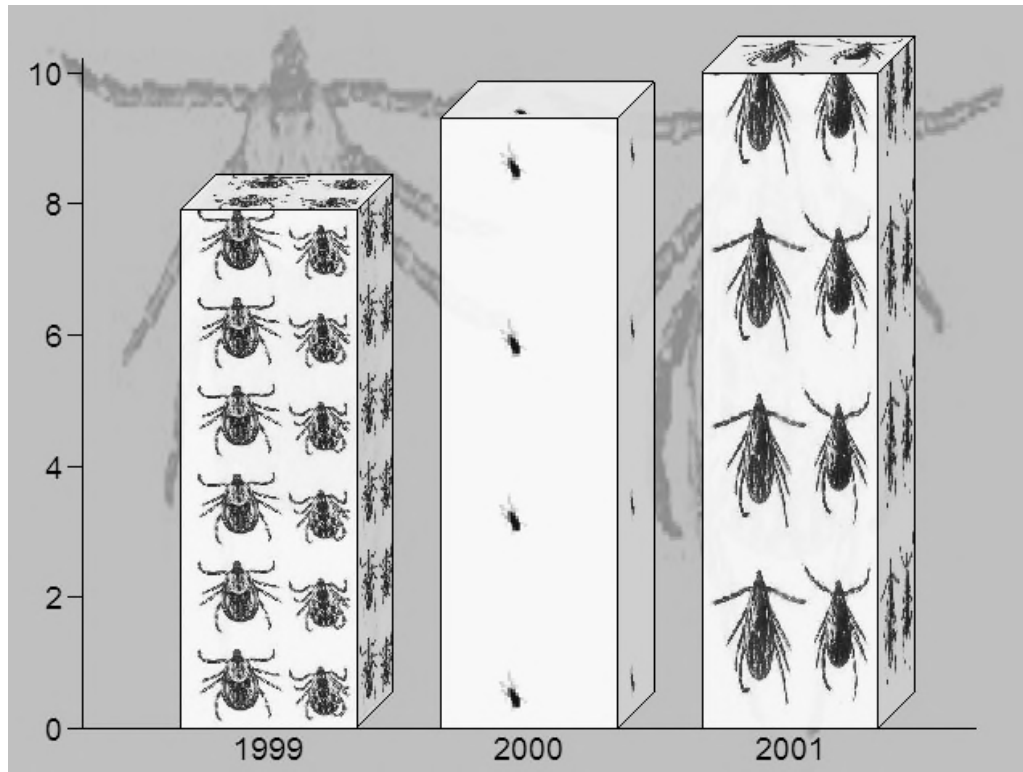
pattern1 v=s image='amerdogtickpair.jpg';
pattern2 v=s image='anibug.gif';
pattern3 v=s image='browndogpair.jpg';

axis2 label=('',) minor=none;
axis1 label=('',);

goptions htitle=.3in ftitle="arial" htext=.2in ftext="arial";

proc gchart data=a2;
vbar3d year /
maxis=axis1 raxis=axis2 discrete type=sum sumvar=value
subgroup=value outline='black' width=15 nolegend
ascending space=5 frame iframe='lonestargroup4.jpg' imagestyle=tile
html=myhtml des="" name="&name" ;
run;

```



IN-LINE STYLES

Pre and Post images may be added to titles, table cells, footnotes, header cells, and ODS TEXT fields via ODS ESCAPECHAR or using formats. An example of inserting a logo in an ODS TEXT statement follows.

```
ods escapechar="^";
ods text='^S={preimage="cmsnew.jpg"}';
```



Nursing Home Compare 5-Star Ratings of Nursing Homes

Nursing Home Rating Report - Incorporating data reported through 09/04/2012

ANNOTATE A LOGO IN SAS/GRAPH

This example is based on several samples from Robert Allison's excellent book "SAS/GRAPH®: Beyond the Basics."

```
/* create an annotate data set with the cmslogo */

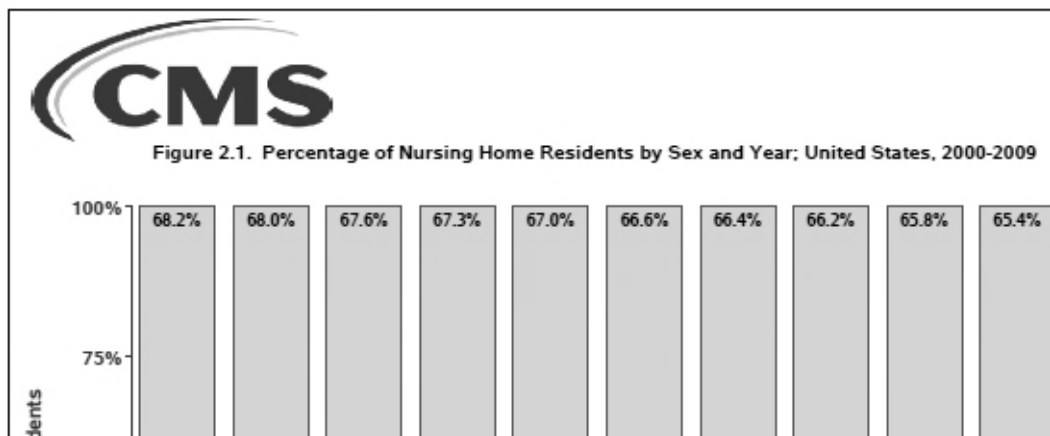
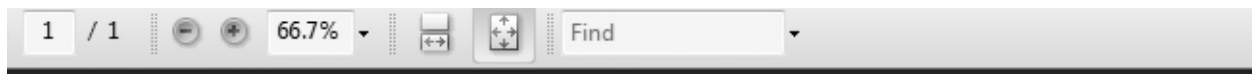
/* use style='fit' or 'tile' */

data logo_anno;
  length function $ 8;
  xsys='3'; ysys='3'; when='a';
  function='move'; x=2; y=85; output;
  function='image'; x=x+25.4; y=y+14; imgpath='smallcmsx.png';
  style='fit'; output;
run;

/* move the graph down with a empty title statement so there is room for the logo */

title1 h=15;
title2 /* h=2 */ "Figure 2.1. Percentage of Nursing Home Residents by Sex and Year;
United States, 2000-2009";

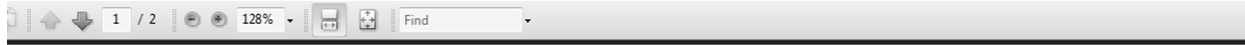
proc gchart data=my_data anno=logo_anno;
format stack_order barfmt.;
vbar year / discrete type=sum sumvar=percent
subgroup=stack_order coutline=gray55 space=2
width=8 noframe raxis=axis1 maxis=axis2
legend=legend1 annotate=barlabel
html=my_html des='' name="&name";
run;
quit;
```



Note that you can also annotate logos and other images into SG procedural output (new in SAS 9.3).

INSERT "SPARKLINES" INTO PROC REPORT COLUMNS

The sparkline images for this example were created using PROC SGPLOT and added to the report using PROC FORMAT and PROC REPORT. The code is fairly simple but lengthy so please contact the author for full code.



Embed Images in PROC REPORT Column

State Code	State	Proportion of G+ Deficiencies					Trend
		2007	2008	2009	2010	2011	
1	Alabama	0.1257862	0.1023622	0.0227273	0.0515021	0.0474138	
2	Alaska	0.1052632	0.1538462	0.0588235	0.2666667	0.125	
4	Arizona	0.1709402	0.2089552	0.2118644	0.1171171	0.1908397	
5	Arkansas	0.1356589	0.1166078	0.1333333	0.1322314	0.0995851	
6	California	0.1113924	0.1223629	0.0975177	0.0783451	0.0713012	
8	Colorado	0.4140969	0.3055556	0.281106	0.3195876	0.2765957	
9	Connecticut	0.3898305	0.2909836	0.2478261	0.258216	0.2760181	
10	Delaware	0.3191489	0.2244898	0.3255814	0.3076923	0.1025641	
11	District of Columbia	0.3809524	0.2352941	0.1904762	0.4736842	0.4210526	

```

/* THIS IS NOT FULL CODE!!! */
/* creating the macro to make the little sparklines */

%macro sparky2(statenm,state);
ods graphics on / width=.75in height=.15in border=off
  outputfmt=png imagemap=on imagename="&state.try2";
proc sgplot data=tempx (where=(fipsst=&statenm.)) ;
  series y=propgplus x=year / markers
    lineattrs=(color=navy thickness=3)
    markerattrs=(color=black symbol=circlefilled size=8)
    name="&state.try2";
  xaxis display=(nolabel noline noticks novalues);
  yaxis display=(nolabel noline noticks novalues);
run;

%mend sparky2;

proc report data=temp nowd;
  column fipsst ... image;
  define fipsst / group;
  define image / 'Trend' computed style(column)=[postimage=try2f.
    just=1 cellwidth=1.25in ] format=blank.;
  /* Set the computed variable equal to the variable */
  /* value that matches the value in the try2f. format. */
  compute image ;
    image = fipsst;
  endcomp;

```

GREPLAY

In this example, based on one of Robert Allison's samples, a map and bar chart are created separately, and then the bar chart is overlaid on the map using PROC GREPLAY.

```

GOPTIONS DEVICE=png;

ODS LISTING CLOSE;
ODS HTML path=odsout body="&name..htm" (title="DHS Data") style=grad1
gtitle gfootnote
;

proc greplay nofs;
igout=work.gseg;
delete _all_;

goptions ftitle="arial/bo" ftext=swissb htitle=6pct htext=2.5pct;
goptions nodisplay;

/* 1st map & bar chart */

title1 height=2 "Decline in Exclusive Breastfeeding from 0 to 5 Months";
title2 font="arial/bo" height=2 "(based on DHS data)";
proc gmap data=work.mydata map=work.clipped_world all;
  id country_name;
  choro gnc_usd / levels=4 html=myhtmlvar1 coutline=black
  nolegend des='' name='map';
run;

pattern1 v=s c=graydd;
axis1 label=none major=(number=5) minor=none;
axis2 label=none;
title h=10pct " "; /* just to add some space at top of bar chart */
footnote h=4pct " ";
proc gchart data=work.mydata;
  vbar country_name / sumvar=diff0_5 descending
  raxis=axis1 maxis=axis2 noframe caxis=graydd
  des='' name='bar';
run;

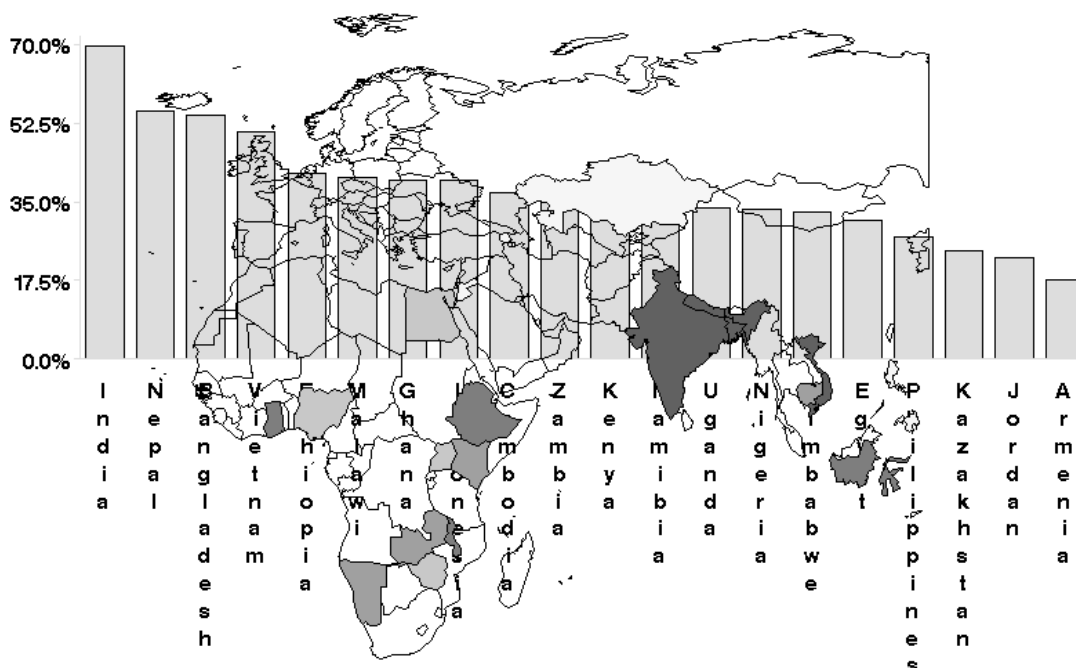
goptions display;
goptions border;

proc greplay tc=tempcat nofs igout=work.gseg;
  tdef one des='One'
  1/ llx = 0    lly = 0
     ulx = 0    uly = 100
     urx = 100  ury = 100
     lrx = 100  lry = 0
;
template = one;
treplay 1:bar 1: map des='' name="&name";
run;

quit;
ODS HTML CLOSE;
ODS LISTING;

```

Percent of Children Aged 0-5 Currently Breastfeeding (based on DHS data)



SPECIAL FONTS IN SAS/GRAPH

In this example, based on another one of Robert Allison's samples, a simple line graph is created using special characters specified in hexadecimal codes. The font is "Monotype Sorts" and the character chosen is a star. Readers are encouraged to consult the section on specifying fonts in SAS/GRAPH in the 9.3 documentation for more details. The hexadecimal code in Monotype Sorts for a star is 48. Similarly, you can use Unicode characters although I have found that some character codes seem to work while others don't.

```
GOPTIONS DEVICE=png xpixels=600 ypixels=400 cback=vpab;
goptions htitle=6pct htext=4pct ftitle="arial/bold" ftext="arial";

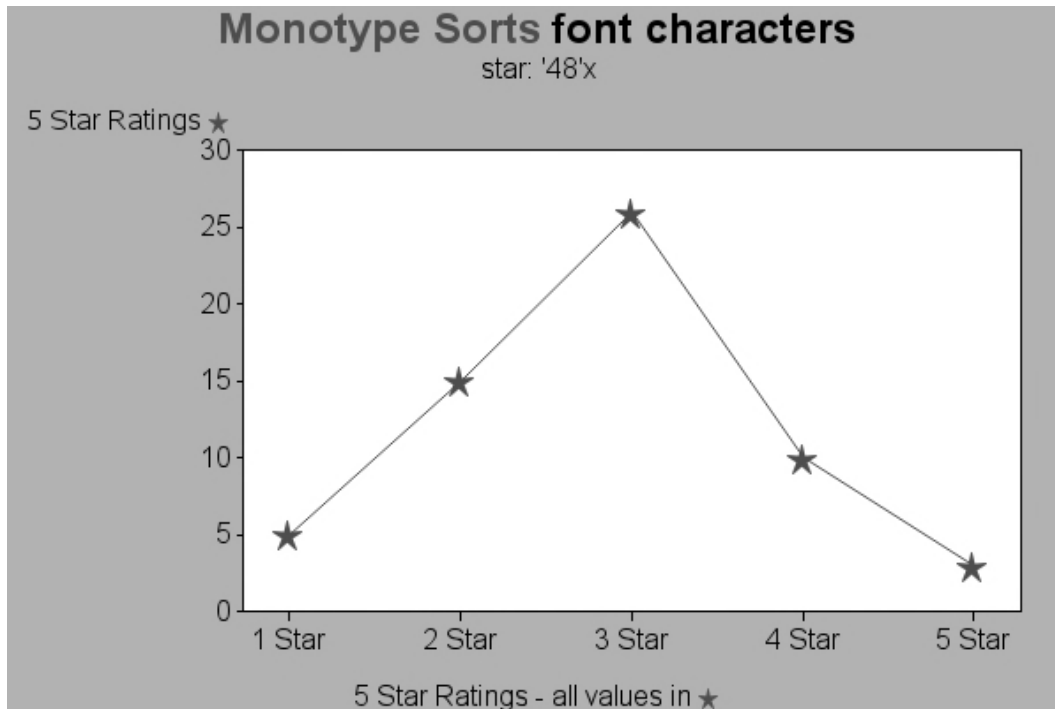
ODS LISTING CLOSE;
ODS HTML path=odsout body="&name..htm" style=minimal;

axis1 label=(font="Helvetica" '5 Star Ratings ' color=red font="Monotype Sorts" '48'x
) order=(0 to 30 by 5) minor=none offset=(0,0);
axis2 label=none offset=(5,5);

symbol1 i=join ci=red color=red font="Monotype Sorts" v='48'x height=3;

title1 c=red "Monotype Sorts " c=black "font characters";
title2 "star: '48'x";
title3 a=-90 " ";
footnote "5 Star Ratings - all values in " c=red font="Monotype Sorts" '48'x;
proc gplot data=a;
plot value*midpoint / vaxis=axis1 haxis=axis2 cframe=white des="" name="&name" ;
run;

quit;
ODS HTML CLOSE;
ODS LISTING;
```



OTHER POSSIBILITIES AND CONSIDERATIONS

Other possibilities for placing custom images in SAS procedural output include using ANNOTATE to display an image and using DSGI (the DATA Step Graphics Interface) to display an image.

Many of the options presented above are effectively SAS/GRAPH style overrides. Consideration must be given to the order of GOPTIONS and procedural statements. In the “tick” examples above, the code demonstrates that a custom style template can be overwritten with a GOPTIONS statement. In general, a statement on the procedure is given highest preference, then GOPTIONS, then custom style templates. In the case of conflicting statements of the same type, the last mentioned statement is used.

CONCLUSION

SAS and ODS provide SAS programmers with many options to include custom images and logos behind the scenes in procedural output. Among these options are custom style templates, custom graph backgrounds, custom graph backplanes, custom patterns on bar graphs, replaying graphic images, annotate and DSGI. ODS output can be greatly enhanced by the use of these options.

REFERENCES & RECOMMENDED READING

<http://robslink.com/SAS/Home.htm>

<http://support.sas.com/documentation/cdl/en/graphref/65389/HTML/default/viewer.htm#n19zerwwhegwjon1r1n1qhsut8tu.htm>

<http://support.sas.com/documentation/onlinedoc/index.html>

<http://support.sas.com/rnd/datavisualization/mapsonline/html/home.html>

<http://support.sas.com/rnd/papers>

<http://support.sas.com/rnd/datavisualization/papers/sugi30/GraphStyles.pdf>

<http://support.sas.com/samples>

<http://support.sas.com/documentation/cdl/en/graphref/65389/HTML/default/viewer.htm#p0t0koedeicq4zn1qj0mniq6wp9b.htm>

Allison, Robert. 2012. *SAS/GRAPH®: Beyond the Basics*. Cary, NC: SAS Institute Inc.

Cartier, Jeff. "Use of Styles in Graphics." *Proceedings of the Thirtieth Annual SAS® Users Group International Conference*. April 2005.

Heath, Dan. "Now You Can Annotate Your Statistical Graphics Procedure Graphs." *Proceedings of SAS Global Forum 2011 Conference*. April 2011.

Huntley, Scott. "How to Add a Little Spice to your PDF Output." *Proceedings of SAS Global Forum 2008 Conference*. March 2008.

Massengill, A. Darrell. "Tips and Tricks: Using SAS/GRAPH® Effectively." *Proceedings of the Thirtieth Annual SAS® Users Group International Conference*. April 2005.

Zdeb, Mike and Allison, Robert. "Stretching the Bounds of SAS/GRAPH Software." *Proceedings of the Thirtieth Annual SAS Users Group International Conference*. April 2005.

Zdeb, Mike and Hadden, Louise. "Zip Code 411: A Well Kept SAS Secret." *Proceedings of the Thirty-First Annual SAS Users Group International Conference*. March 2006.

Zdeb, Mike. 2002. *Maps Made Easy Using SAS*. Cary, NC: SAS Institute Inc.

Zender, Cynthia. "Funny ^Stuff~ in My Code: Using ODS ESCAPECHAR." *Proceedings of SAS Global Forum 2007 Conference*. April 2007.

Zender, Cynthia. "Creating Complex Reports." *Proceedings of SAS Global Forum 2008 Conference*. March 2008.

ACKNOWLEDGMENTS & DISCLAIMERS

Robert Allison of SAS®, SAS/GRAPH Wizard

Scott Huntley of SAS®, ODS Printer Guru

Mike Zdeb, SAS/GRAPH® Map Expert, who truly does make mapping easy with SAS®

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 registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION

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

Email: louise_hadden@abtassoc.com

Sample code is available from the author upon request.

