Paper 65-26

Simple Ways To Publish Reports And Graphs On The Web

Prepared by

destiny
CORPORATION

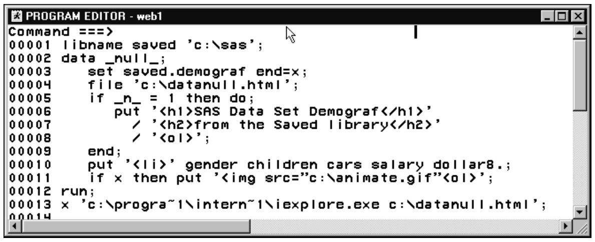
International SAS® Training and Consulting
A SAS Institute Quality Partner™

100 Great Meadow Rd, Suite 601
Wethersfield, CT 06109-2379
Phone: (860) 721-1684 1-800-7TRAINING
Fax: (860) 721-9784
Web: www.destinycorp.com Email: info@destinycorp.com

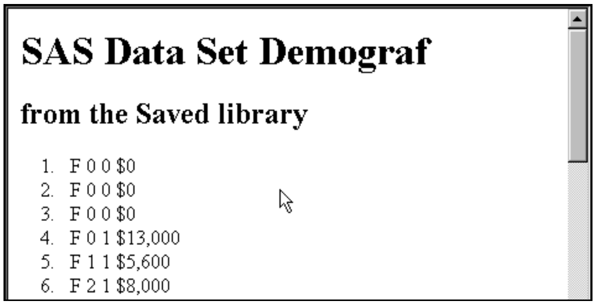
This presentation is designed to show SAS users some simple and quick ways to publish reports and graphs from various SAS software modules, including hyper linking and some basic uses of the Output Delivery System. This presentation is designed for new users in SAS software.

Basic HTML Publishing

The most basic form is brute force HTML generation using the traditional data step. This tends to be the most flexible, but the most time consuming and requires complete knowledge of how HTML works. FILE and PUT statements are used extensively.



```
Command ===>
libname saved 'c:\sas';
data _null_;
   set saved.demograf end=x;
   file 'c:\datanull.html';
   if _n_ = 1 then do;
      put '<h1>SAS Data Set Demograf</h1>'
        / '<h2>from the Saved library</h2>'
        / '<ol>';
   end;
   put '<li>' gender children cars salary dollar8.;
   if x then put '<img src="c:\animate.gif"<ol>';
run;
x 'c:\progra~1\intern~1\iexplore.exe c:\datanull.html';
```



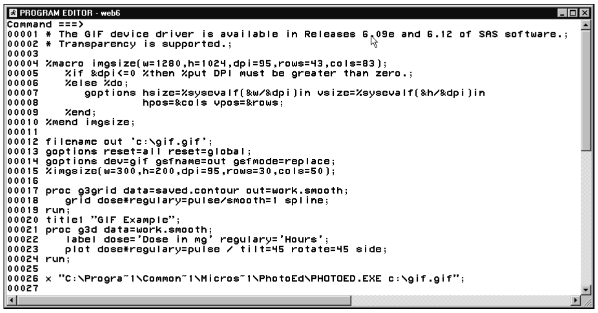
SAS Data Set Demograf

from the Saved library

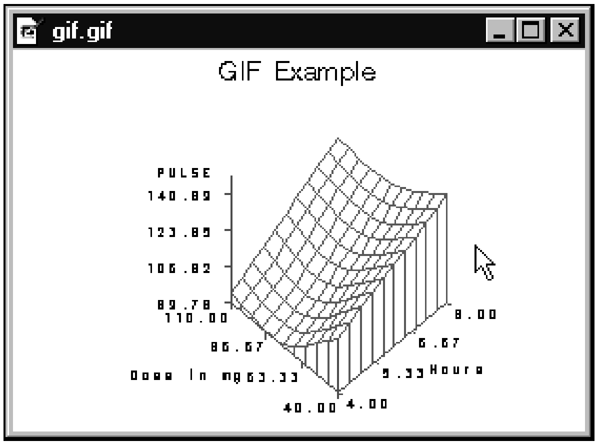
1. F 0 0 $0
2. F 0 0 $0
3. F 0 0 $0
4. F 0 1 $13,000
5. F 1 1 $5,600
6. F 2 1 $8,000

The following drivers are designed to create graphical output used on the web with SAS/Graph software.

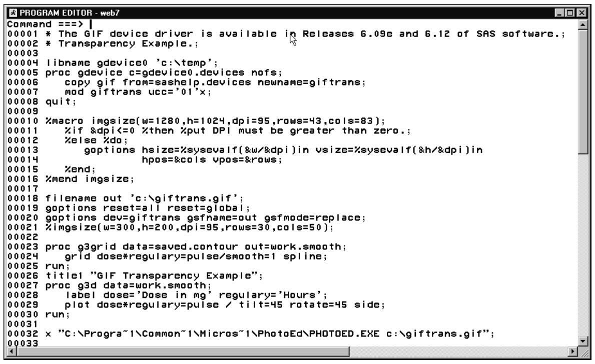
Graphical Publishing with GIFs



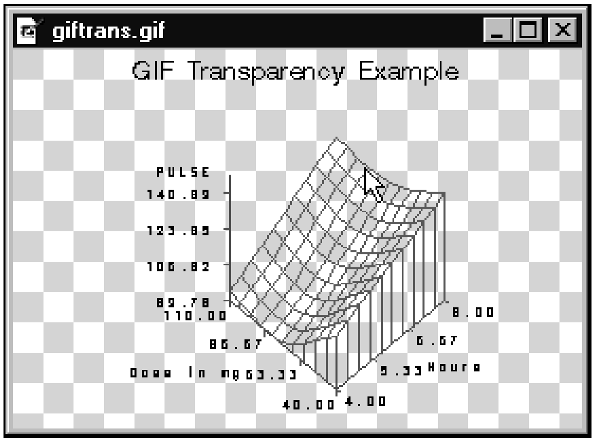
```
Command ===>
* The GIF device driver is available in Releases 6.09e and 6.12 of SAS software.;
* Transparency is supported.;

%macro imgsize(w=1280,h=1024,dpi=95,rows=43,cols=83);
   %if &dpi<=0 %then %put DPI must be greater than zero.;
   %else %do;
      goptions hsize=%sysevalf(&w/&dpi)in vsize=%sysevalf(&h/&dpi)in
               hpos=&cols vpos=&rows;
   %end;
%mend imgsize;

filename out 'c:\gif.gif';
goptions reset=all reset=global;
goptions dev=gif gsfname=out gsfmode=replace;
%imgsize(w=300,h=200,dpi=95,rows=30,cols=50);

proc g3grid data=saved.contour out=work.smooth;
   grid dose*regulary=pulse/smooth=1 spline;
run;
title1 "GIF Example";
proc g3d data=work.smooth;
   label dose='Dose in mg' regulary='Hours';
   plot dose*regulary=pulse / tilt=45 rotate=45 side;
run;

x "C:\Progra~1\Common~1\Micros~1\PhotoEd\PHOTOED.EXE c:\gif.gif";
```



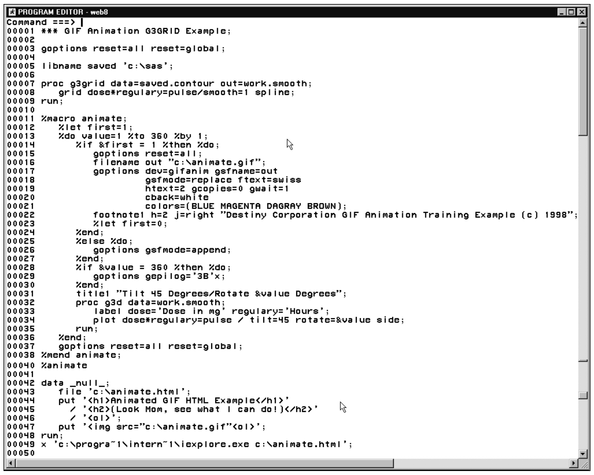
Graphical Publishing with GIFs Transparency



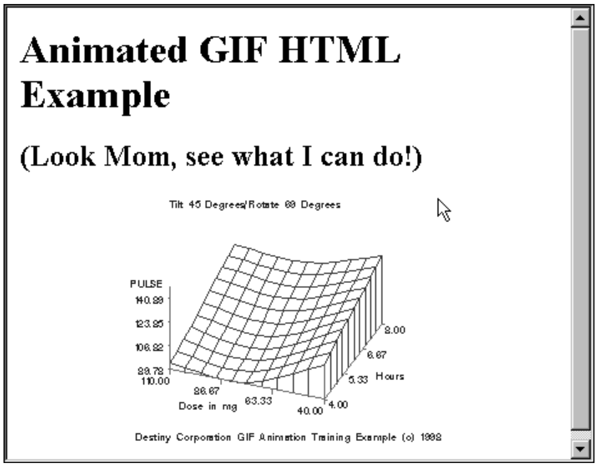
```
Command ===>
* The GIF device driver is available in Releases 6.09e and 6.12 of SAS software.;
* Transparency Example.;

libname gdevice0 'c:\temp';
proc gdevice c=gdevice0.devices nofs;
   copy gif from=sashelp.devices newname=giftrans;
   mod giftrans ucc='01'x;
quit;

%macro imgsize(w=1280,h=1024,dpi=95,rows=43,cols=83);
   %if &dpi<=0 %then %put DPI must be greater than zero.;
   %else %do;
      goptions hsize=%sysevalf(&w/&dpi)in vsize=%sysevalf(&h/&dpi)in
               hpos=&cols vpos=&rows;
   %end;
%mend imgsize;

filename out 'c:\giftrans.gif';
goptions reset=all reset=global;
goptions dev=giftrans gsfname=out gsfmode=replace;
%imgsize(w=300,h=200,dpi=95,rows=30,cols=50);

proc g3grid data=saved.contour out=work.smooth;
   grid dose*regulary=pulse/smooth=1 spline;
run;
title1 "GIF Transparency Example";
proc g3d data=work.smooth;
   label dose='Dose in mg' regulary='Hours';
   plot dose*regulary=pulse / tilt=45 rotate=45 side;
run;

x "C:\Progra~1\Common~1\Micros~1\PhotoEd\PHOTOED.EXE c:\giftrans.gif";
```



Animation with GIFs



```
*** GIF Animation G3GRID Example;

goptions reset=all reset=global;

libname saved 'c:\sas';

proc g3grid data=saved.contour out=work.smooth;
   grid dose*regulary=pulse/smooth=1 spline;
run;

%macro animate;
   %let first=1;
   %do value=1 %to 360 %by 1;
      %if &first = 1 %then %do;
         goptions reset=all;
         filename out "c:\animate.gif";
         goptions dev=gifanim gsfname=out
                  gsfmode=replace ftext=swiss
                  htext=2 gcopies=0 gwait=1
                  cback=white
                  colors=(BLUE MAGENTA DAGRAY BROWN);
         footnote1 h=2 j=right "Destiny Corporation GIF Animation Training Example (c) 1998";
         %let first=0;
      %end;
      %else %do;
         goptions gsfmode=append;
      %end;
      %if &value = 360 %then %do;
         goptions gepilog='3B'x;
      %end;
      title1 "Tilt 45 Degrees/Rotate &value Degrees";
      proc g3d data=work.smooth;
         label dose='Dose in mg' regulary='Hours';
         plot dose*regulary=pulse / tilt=45 rotate=&value side;
      run;
   %end;
   goptions reset=all reset=global;
%mend animate;
%animate

data _null_;
   file 'c:\animate.html';
   put '<h1>Animated GIF HTML Example</h1>'
     / '<h2>(Look Mom, see what I can do!)</h2>'
     / '<ol>';
   put '<img src="c:\animate.gif"<ol>';
run;
x 'c:\progra~1\intern~1\iexplore.exe c:\animate.html';
```



Animated GIF HTML Example

(Look Mom, see what I can do!)

Tilt 45 Degrees/Rotate 68 Degrees
PULSE
140.89
123.85
106.82
89.78
110.00
86.67
Dose in mg
63.33
40.00
4.00
5.33
6.67
8.00
Hours
Destiny Corporation GIF Animation Training Example (c) 1998

Version 8 and the Output Delivery System

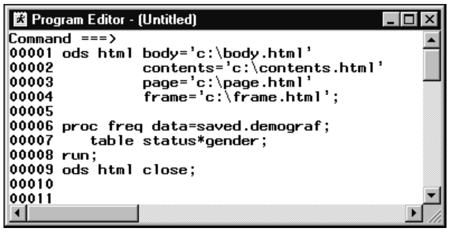
When ODS was introduced in Version 8 of SAS software, a whole new way of publishing was created.

The simplest form of output can be seen by the following example.

First, notice the typical style of using ODS.

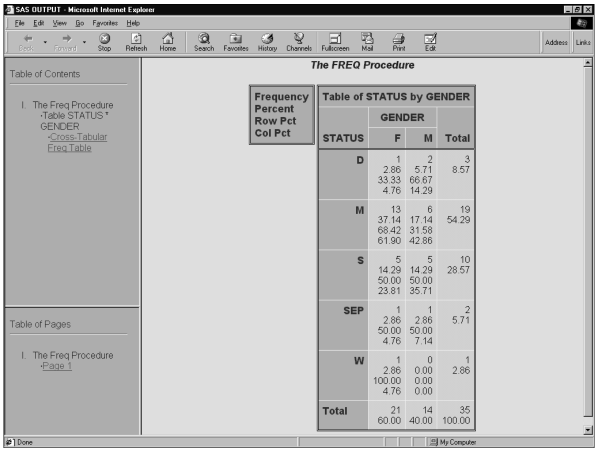
1. The ODS 'capture style' utility is turned on.
2. The process executes.
3. The ODS utility is turned off.

The general ODS syntax can look like the following.



```
ods html body='c:\body.html'
         contents='c:\contents.html'
         page='c:\page.html'
         frame='c:\frame.html';

proc freq data=saved.demograf;
   table status*gender;
run;
ods html close;
```

This syntax routes HTML output anywhere you desire. It creates four files. The FRAME.HTML file pulls them all together.

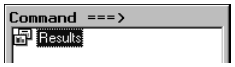


The FREQ Procedure

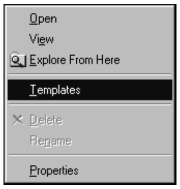
Frequency Percent Row Pct Col Pct	Table of STATUS by GENDER		
	GENDER		
STATUS	F	M	Total
D	1 2.86 33.33 4.76	2 5.71 66.67 14.29	3 8.57
M	13 37.14 68.42 61.90	6 17.14 31.58 42.86	19 54.29
S	5 14.29 50.00 23.81	5 14.29 50.00 35.71	10 28.57
SEP	1 2.86 50.00 4.76	1 2.86 50.00 7.14	2 5.71
W	1 2.86 100.00 4.76	0 0.00 0.00 0.00	1 2.86
Total	21 60.00	14 40.00	35 100.00

Styles in HTML

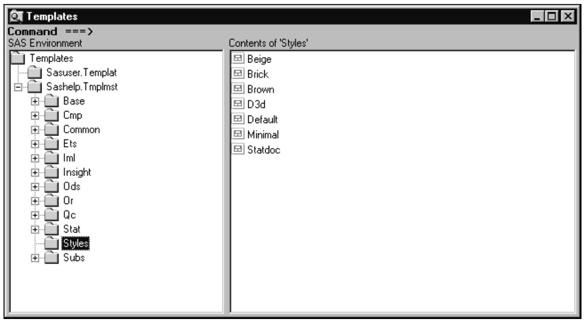
There are several HTML styles that come with SAS. They are designed to allow the user to create an HTML standard of colors, fonts and more across all HTML output. Any of the default ones can be used, or they can be modified to suit your needs. To see the existing styles, go to the Results window and select.



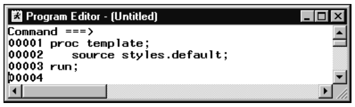
Right click and select Templates.



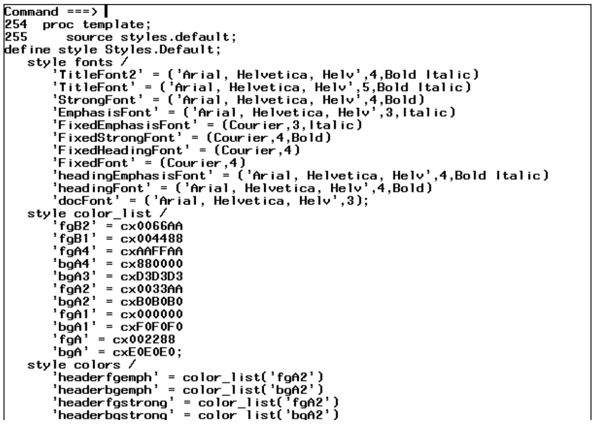
Select SASHELP.TMPLMST, Styles to see the available styles.



To examine a style, execute Proc Template with the Style name. Default is the default style used.



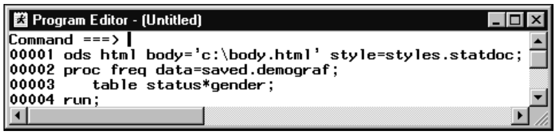
```
proc template;
   source styles.default;
run;
```



```
Command ===>
proc template;
    source styles.default;
define style Styles.Default;
   style fonts /
      'TitleFont2' = ('Arial, Helvetica, Helv',4,Bold Italic)
      'TitleFont' = ('Arial, Helvetica, Helv',5,Bold Italic)
      'StrongFont' = ('Arial, Helvetica, Helv',4,Bold)
      'EmphasisFont' = ('Arial, Helvetica, Helv',3,Italic)
      'FixedEmphasisFont' = (Courier,3,Italic)
      'FixedStrongFont' = (Courier,4,Bold)
      'FixedHeadingFont' = (Courier,4)
      'FixedFont' = (Courier,4)
      'headingEmphasisFont' = ('Arial, Helvetica, Helv',4,Bold Italic)
      'headingFont' = ('Arial, Helvetica, Helv',4,Bold)
      'docFont' = ('Arial, Helvetica, Helv',3);
   style color_list /
      'fgB2' = cx0066AA
      'fgB1' = cx004488
      'fgA4' = cxAAFFAA
      'bgA4' = cx880000
      'bgA3' = cxD3D3D3
      'fgA2' = cx0033AA
      'bgA2' = cxB0B0B0
      'fgA1' = cx000000
      'bgA1' = cxF0F0F0
      'fgA' = cx002288
      'bgA' = cxE0E0E0;
   style colors /
      'headerfgemph' = color_list('fgA2')
      'headerbgemph' = color_list('bgA2')
      'headerfgstrong' = color_list('fgA2')
      'headerbgstrong' = color_list('bgA2')
```

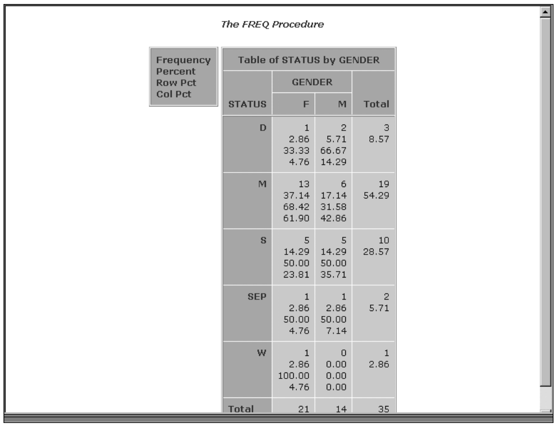
Cut and paste the code for modification. Rerun it with a new name and your own style can be created.

To select a style, use the ODS statement



```
Command ===>
ods html body='c:\body.html' style=styles.statdoc;
proc freq data=saved.demograf;
   table status*gender;
run;
```

It produces the following HTML. (Looks different than styles.default)



The FREQ Procedure

Frequency Percent Row Pct Col Pct

Table of STATUS by GENDER

STATUS	GENDER F	GENDER M	Total
D	1 2.86 33.33 4.76	2 5.71 66.67 14.29	3 8.57
M	13 37.14 68.42 61.90	6 17.14 31.58 42.86	19 54.29
S	5 14.29 50.00 23.81	5 14.29 50.00 35.71	10 28.57
SEP	1 2.86 50.00 4.76	1 2.86 50.00 7.14	2 5.71
W	1 2.86 100.00 4.76	0 0.00 0.00 0.00	1 2.86
Total	21	14	35

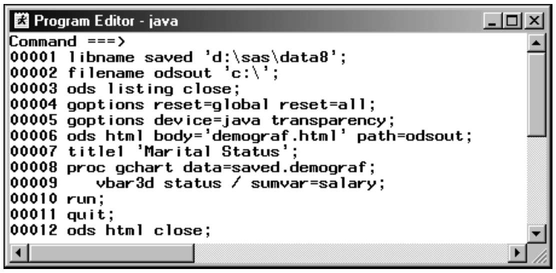
See the online help to learn more about changing styles and what the values mean.

Graphical Support

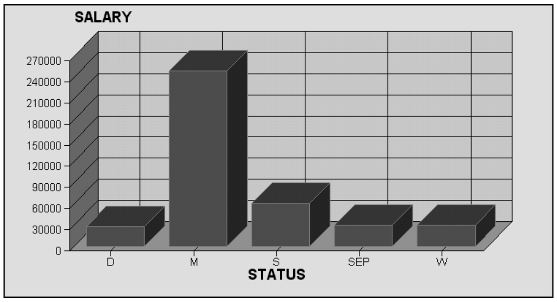
There are several other drivers available for outputting graphical objects. Consider the following.

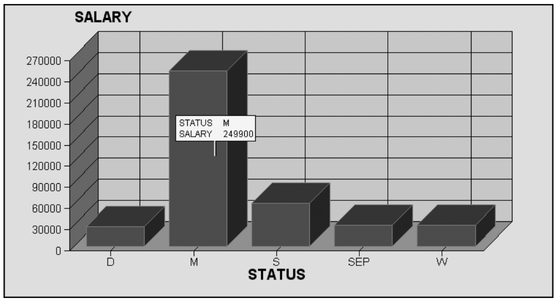
Java Support

There is Java support. Consider the following driver.

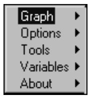


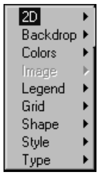
```
Command ===>
libname saved 'd:\sas\data8';
filename odsout 'c:\';
ods listing close;
goptions reset=global reset=all;
goptions device=java transparency;
ods html body='demograf.html' path=odsout;
title1 'Marital Status';
proc gchart data=saved.demograf;
   vbar3d status / sumvar=salary;
run;
quit;
ods html close;
```

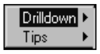


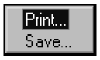


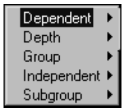
There are several menus available from the right click of a mouse.

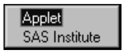








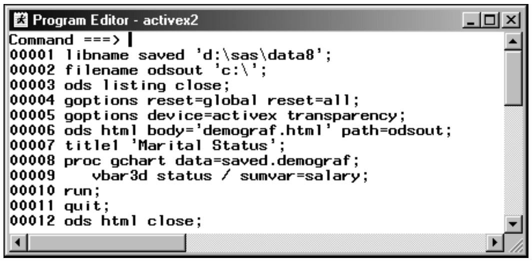




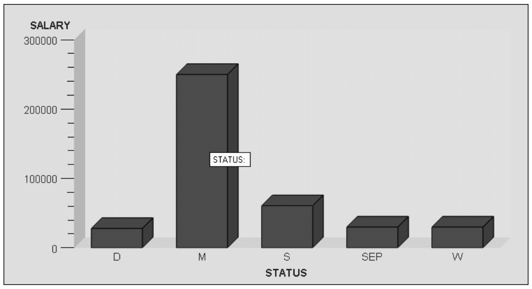
Experiment with each to understand the capabilities.

Active X Support

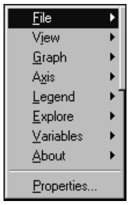
SAS now supports Active X controls. Consider the following code with the Active X Driver.

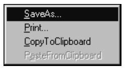


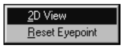
```
Command ===>
libname saved 'd:\sas\data8';
filename odsout 'c:\';
ods listing close;
goptions reset=global reset=all;
goptions device=activex transparency;
ods html body='demograf.html' path=odsout;
title1 'Marital Status';
proc gchart data=saved.demograf;
   vbar3d status / sumvar=salary;
run;
quit;
ods html close;
```

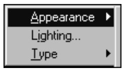


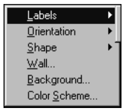
A right click yields many possibilities.

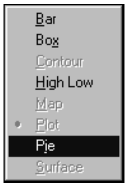


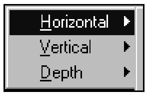


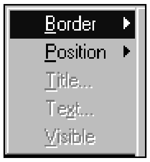


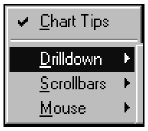












Consider testing out the different options.

Using Styles in Procedures

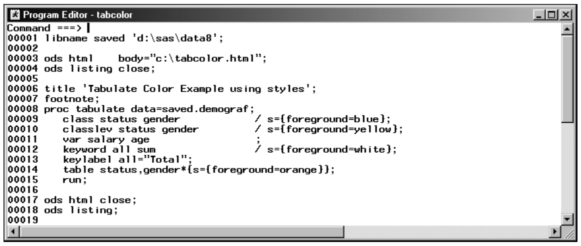
Style options are available in SAS. Their syntax is added to procedures when formatting them for output. This is available on the procedure level in the following way. Use the syntax

S={foreground=blue}

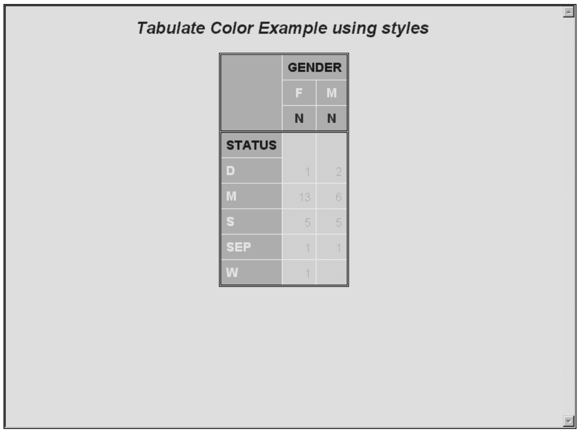
At various locations to specify colors.

Current supported Procedures are Tabulate and Report.

Styles in Proc Tabulate



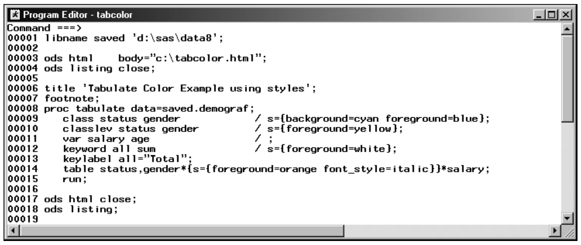
```
Command ===>
libname saved 'd:\sas\data8';

ods html    body="c:\tabcolor.html";
ods listing close;

title 'Tabulate Color Example using styles';
footnote;
proc tabulate data=saved.demograf;
   class status gender            / s={foreground=blue};
   classlev status gender         / s={foreground=yellow};
   var salary age                 ;
   keyword all sum                / s={foreground=white};
   keylabel all="Total";
   table status,gender*{s={foreground=orange}};
   run;

ods html close;
ods listing;
```



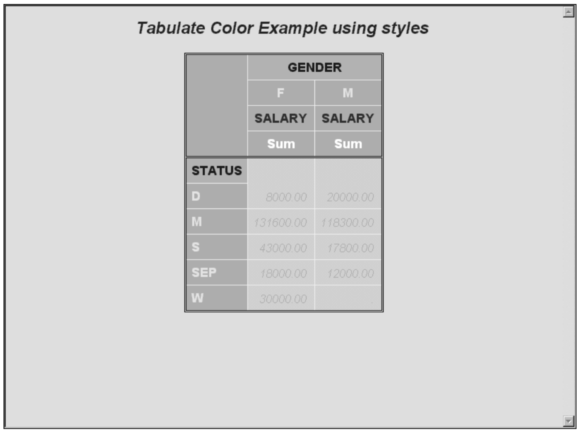
Tabulate Color Example using styles

	GENDER	
	F	M
	N	N
STATUS		
D	1	2
M	13	6
S	5	5
SEP	1	1
W	1	.

Additional examples use background and font_style.



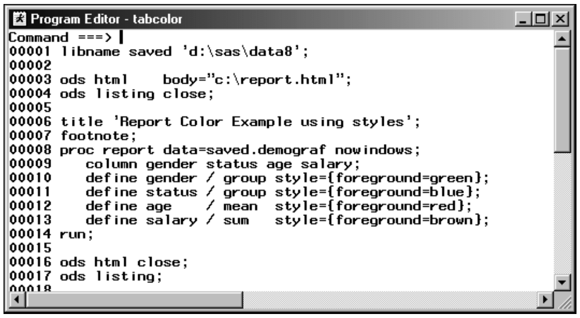
```
Command ===>
libname saved 'd:\sas\data8';

ods html    body="c:\tabcolor.html";
ods listing close;

title 'Tabulate Color Example using styles';
footnote;
proc tabulate data=saved.demograf;
   class status gender            / s={background=cyan foreground=blue};
   classlev status gender         / s={foreground=yellow};
   var salary age                 / ;
   keyword all sum                / s={foreground=white};
   keylabel all="Total";
   table status,gender*{s={foreground=orange font_style=italic}}*salary;
   run;

ods html close;
ods listing;
```



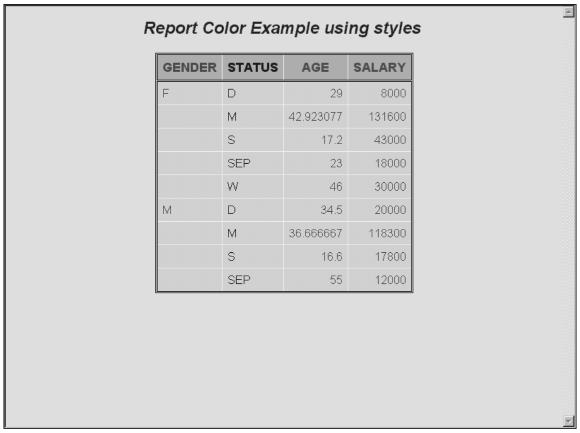
Tabulate Color Example using styles

	GENDER	
	F	M
	SALARY	SALARY
	Sum	Sum
STATUS		
D	8000.00	20000.00
M	131600.00	118300.00
S	43000.00	17800.00
SEP	18000.00	12000.00
W	30000.00	.

Styles in Proc Report



```
libname saved 'd:\sas\data8';

ods html    body="c:\report.html";
ods listing close;

title 'Report Color Example using styles';
footnote;
proc report data=saved.demograf nowindows;
   column gender status age salary;
   define gender / group style={foreground=green};
   define status / group style={foreground=blue};
   define age    / mean  style={foreground=red};
   define salary / sum   style={foreground=brown};
run;

ods html close;
ods listing;
```

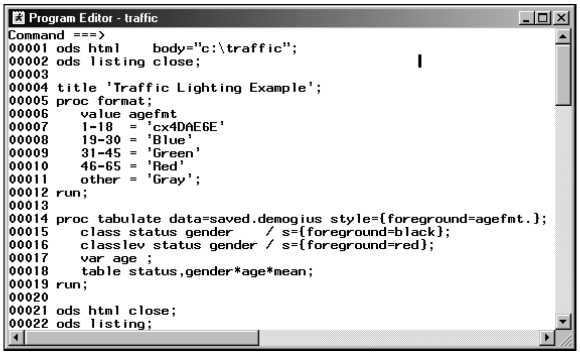


Report Color Example using styles

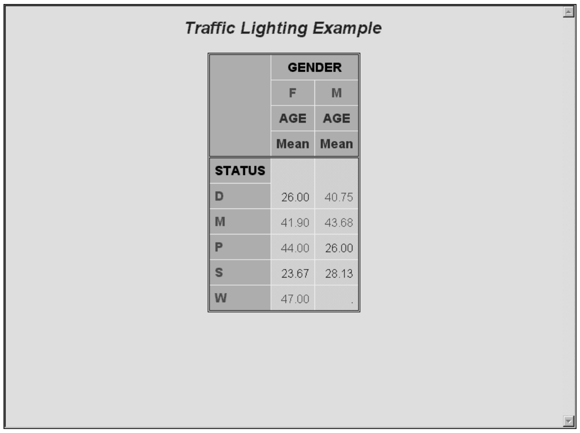
GENDER	STATUS	AGE	SALARY
F	D	29	8000
	M	42.923077	131600
	S	17.2	43000
	SEP	23	18000
	W	46	30000
M	D	34.5	20000
	M	36.666667	118300
	S	16.6	17800
	SEP	55	12000

Traffic Lighting

This is a technique that employs styles with formatted values to allow changing the color and presentation of cells as the value changes. Examine the following code.



```
ods html    body="c:\traffic";
ods listing close;

title 'Traffic Lighting Example';
proc format;
   value agefmt
   1-18  = 'cx4DAE6E'
   19-30 = 'Blue'
   31-45 = 'Green'
   46-65 = 'Red'
   other = 'Gray';
run;

proc tabulate data=saved.demogius style={foreground=agefmt.};
   class status gender    / s={foreground=black};
   classlev status gender / s={foreground=red};
   var age ;
   table status,gender*age*mean;
run;

ods html close;
ods listing;
```



Traffic Lighting Example

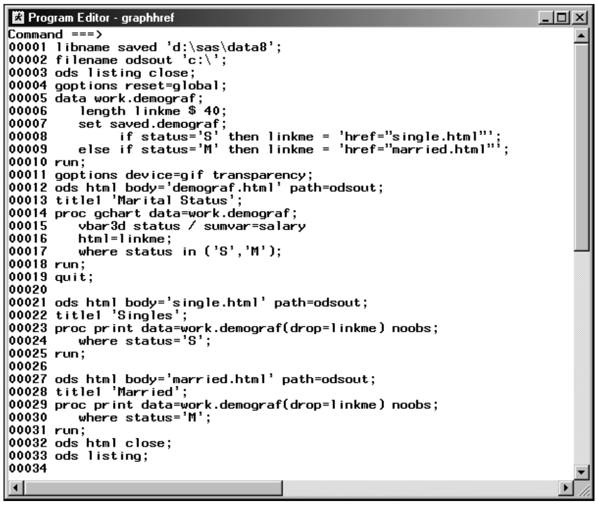
	GENDER	
	F	M
	AGE	AGE
	Mean	Mean
STATUS		
D	26.00	40.75
M	41.90	43.68
P	44.00	26.00
S	23.67	28.13
W	47.00	.

Hyperlinks in Graphs and Reports

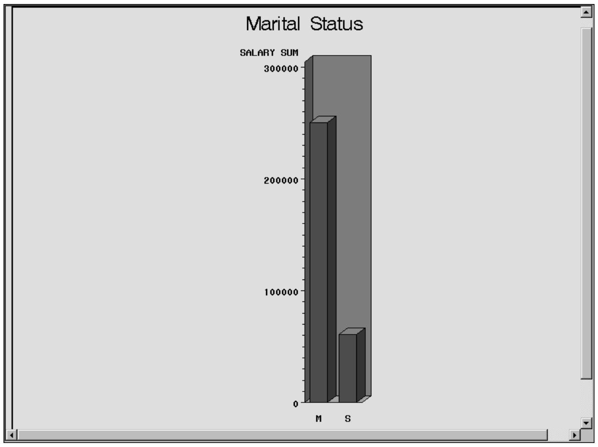
Hyperlinks in Graphs

In Version 8 of SAS Software, HTML examples that link to each other are possible. Wouldn't it be great to create a graph and then be able to click on the items in the graph to branch to appropriate detail about those items?

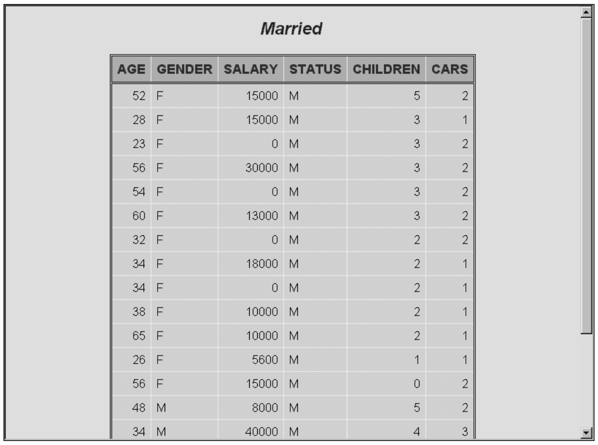
The following code could create that result.



```
libname saved 'd:\sas\data8';
filename odsout 'c:\';
ods listing close;
goptions reset=global;
data work.demograf;
   length linkme $ 40;
   set saved.demograf;
        if status='S' then linkme = 'href="single.html"';
   else if status='M' then linkme = 'href="married.html"';
run;
goptions device=gif transparency;
ods html body='demograf.html' path=odsout;
title1 'Marital Status';
proc gchart data=work.demograf;
   vbar3d status / sumvar=salary
   html=linkme;
   where status in ('S','M');
run;
quit;

ods html body='single.html' path=odsout;
title1 'Singles';
proc print data=work.demograf(drop=linkme) noobs;
   where status='S';
run;

ods html body='married.html' path=odsout;
title1 'Married';
proc print data=work.demograf(drop=linkme) noobs;
   where status='M';
run;
ods html close;
ods listing;
```

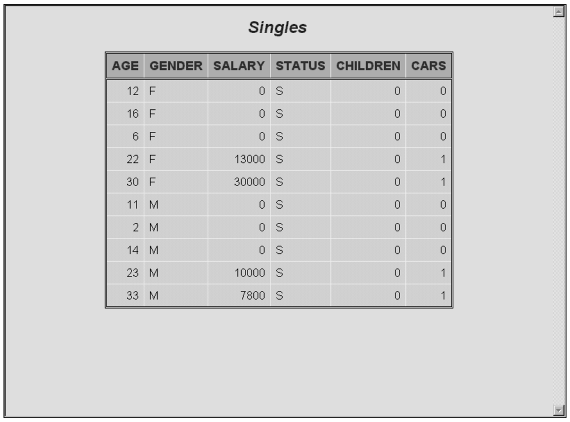


Click on the appropriate bars to see the detail.



Married

AGE	GENDER	SALARY	STATUS	CHILDREN	CARS
52	F	15000	M	5	2
28	F	15000	M	3	1
23	F	0	M	3	2
56	F	30000	M	3	2
54	F	0	M	3	2
60	F	13000	M	3	2
32	F	0	M	2	2
34	F	18000	M	2	1
34	F	0	M	2	1
38	F	10000	M	2	1
65	F	10000	M	2	1
26	F	5600	M	1	1
56	F	15000	M	0	2
48	M	8000	M	5	2
34	M	40000	M	4	3



Singles

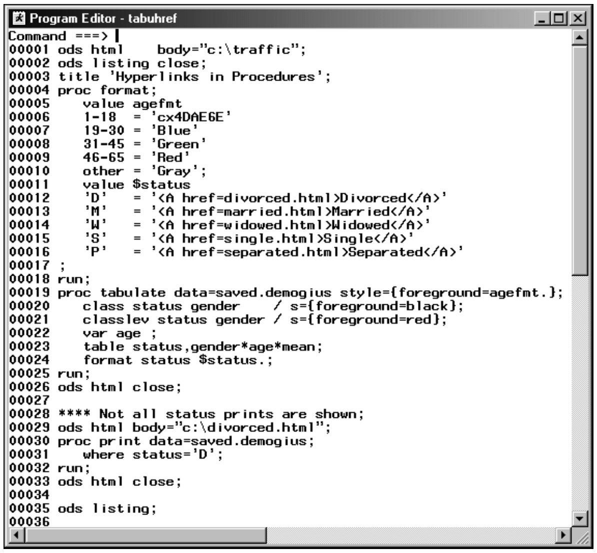
AGE	GENDER	SALARY	STATUS	CHILDREN	CARS
12	F	0	S	0	0
16	F	0	S	0	0
6	F	0	S	0	0
22	F	13000	S	0	1
30	F	30000	S	0	1
11	M	0	S	0	0
2	M	0	S	0	0
14	M	0	S	0	0
23	M	10000	S	0	1
33	M	7800	S	0	1

The key items to get this to work are the following:

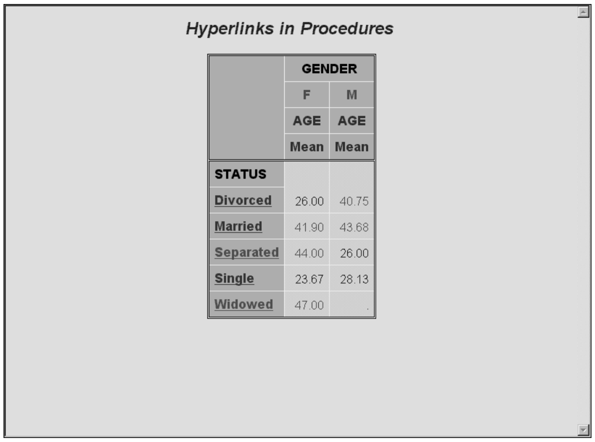
1. Use the GIF driver for graph output.
2. Create a variable attached to the grouped data item that includes a valid hyperlink. Here we use LINKME.
3. Use the HTML= option on the charting statement.

Hyperlinks in Reports

This is an easy task by formatting the values with predefined hyper links. Examine the following code:



```
Command ===>
ods html    body="c:\traffic";
ods listing close;
title 'Hyperlinks in Procedures';
proc format;
   value agefmt
   1-18  = 'cx4DAE6E'
   19-30 = 'Blue'
   31-45 = 'Green'
   46-65 = 'Red'
   other = 'Gray';
   value $status
   'D'   = '<A href=divorced.html>Divorced</A>'
   'M'   = '<A href=married.html>Married</A>'
   'W'   = '<A href=widowed.html>Widowed</A>'
   'S'   = '<A href=single.html>Single</A>'
   'P'   = '<A href=separated.html>Separated</A>'
;
run;
proc tabulate data=saved.demogius style={foreground=agefmt.};
   class status gender    / s={foreground=black};
   classlev status gender / s={foreground=red};
   var age ;
   table status,gender*age*mean;
   format status $status.;
run;
ods html close;

**** Not all status prints are shown;
ods html body="c:\divorced.html";
proc print data=saved.demogius;
   where status='D';
run;
ods html close;

ods listing;

```



Hyperlinks in Procedures

	GENDER	
	F	M
	AGE	AGE
	Mean	Mean
STATUS		
Divorced	26.00	40.75
Married	41.90	43.68
Separated	44.00	26.00
Single	23.67	28.13
Widowed	47.00	.

This paper consists of excerpts from Destiny Corporation's course materials. Please contact Destiny Corporation for more information.