

Prepared by



International SAS® Training and Consulting

Destiny Corporation

100 Great Meadow Rd, Suite 601

Wethersfield, CT 06109-2379

Phone: (860) 721-1684 1-800-7TRAINING

Fax: (860) 721-9784

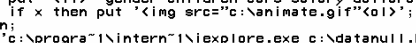
Web: www.destinycorp.com Email: info@destinycorp.com

Copyright 2002

This presentation is designed to demonstrate the different possibilities we use when we build web based applications for clients and teach them the web publishing techniques we use with SAS Software. Actual examples are available on our web site.

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.



```

PROGRAM EDITOR - web1
Command Prompt
00001 libname saved 'c:\sas';
00002 data _null_;
00003   set saved.demograg end=x;
00004   file 'c:\nnull.html';
00005   if _n_ = 1 then do;
00006     put ' <h1>SRS Data Set Demograg</h1>'
00007         ' <h2>from the Saved libname</h2>'
00008         ' <ol>';
00009   end;
00010   put ' <li> gender children care salary dollars';
00011   if x then put '  <ol>';
00012 run;
00013 x 'c:\program\intern\iexplore.exe c:\nnull.html';
00014

```

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

Data Set Publishing with DS2HTML

SAS offers a macro that allows one to take a SAS data file and present it in an HTML form.

```

PROGRAM EDITOR - web2
Command ==>
00001 libname saved 'c:\eas';
00002 data work.demograf;
00003 set saved.demograf;
00004 format salary dollar10.;
00005 run;
00006
00007 title1 'Data Set Table';
00008
00009 ods2htm(data =work.demograf,
00010 where =status ^= 'SEP',
00011 var =gender status children salary,
00012 htmlfile=c:\test1.html,
00013 runmode =b);
00014
00015 x 'c:\program\intern\i\explore.exe c:\test1.html';
00016

```

Data Set Table

GENDER	STATUS	CHILDREN	SALARY
F	S	0	\$0
F	S	0	\$0
F	S	0	\$0
F	S	0	\$13,000
F	M	1	\$5,600

SAS Output Publishing with OUT2HTM

Any procedure's output can be captures and published with the OUT2HTM macro.

```
PROGRAM EDITOR - web3
Command >>>
00001 libname saved 'c:\sas';
00002 xoutzhtn(capture=on);
00003
00004 title "Proc Report Output";
00005 options nodate formchar='|---+|---+---|\<'>
00006 PROC REPORT DATA=SASUSER.DENOGRAF PS=54 SPLIT=/' CENTER NOWINDOWS;
00007 COLUMN STATUS RAGE SALARY CHILDREN CARS;
00008 DEFINE GENDER / GROUP FORMAT= $8. WIDTH=8 SPACING=2 LEFT "GENDER";
00009 DEFINE STATUS / GROUP FORMAT= $8. WIDTH=8 SPACING=2 LEFT "STATUS";
00010 DEFINE AGE / MEAN FORMAT= B8. WIDTH=8 SPACING=2 RIGHT "AGE";
00011 DEFINE SALARY / SUM FORMAT= BEST8. WIDTH=8 SPACING=2 RIGHT "SALARY";
00012 DEFINE CHILDREN / SUM FORMAT= BEST8. WIDTH=5 SPACING=2 RIGHT "CHILDREN";
00013 CARS / SUM FORMAT= BEST8. WIDTH=8 SPACING=2 RIGHT "CARS";
00014 BREAK AFTER GENDER / DL SKIP SUMMARIZE ;
00015 RUN;
00016
00017 title "Proc Freq Output";
00018 proc freq data=saved.denograf;
00019 table status*gender;
00020 run;
00021
00022 title "Proc Tabulate Output";
00023 proc tabulate data=saved.denograf;
00024 class status gender;
00025 var salary;
00026 table status,gender*salary;
00027 run;
00028
00029 xoutzhtn(capture=off,
00030 rmmode=b
00031 htmfile=c:\test1.html,
00032 ctext=blue,
00033 ccolor=green);
00034 %x prog% \In\Internet\IExplore.exe c:\test1.html;
00035
```

Proc Report Output					
GENDER	STATUS	AGE	SALARY	CHILDREN	CARS
F	D	29	8000	2	1
	M	43	131600	31	20
	S	17	43000	0	2
	SEP	23	18000	1	1
	W	45	30000	3	1

F		35	230600	37	25
M	D	35	20000	5	2
	M	37	118300	10	9
	S	17	17800	0	2
	SEP	55	12000	2	1

M		31	168100	25	14

Proc Freq Output

TABLE OF STATUS BY GENDER

STATUS		GENDER		Total
Frequency	Percent			
Row Pct	Col Pct	F	M	
D		1	2	3
		2.86	5.71	8.57
		33.33	66.67	
		4.76	14.29	
M		13	6	19
		37.14	17.14	54.29
		68.42	31.58	
		61.90	42.86	
S		14.29	14.29	28.57
		50.00	50.00	
		23.81	35.71	
SEP		1	1	2
		2.86	2.86	5.71
		50.00	50.00	

Proc Tabulate Output

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

Tabulate Publishing with TAB2HTM

A special macro has been designed to capture the output of a PROC TABULATE and create an HTML table. This is a better presentation for a PROC TABULATE than using OUT2HTM.

```

Command ===>
00001 libname saved 'c:\sae';
00002 %tab2htm(capture=on);
00003
00004 title1 'Proc Tabulate Output';
00005 options nodate;
00006
00007 proc tabulate data=saved.demograp formchar='8283048586878889a8b8c'x format=t.;
00008 class status gender;
00009 table status*gender;
00010 run;
00011
00012 %tab2htm(capture=off,
00013           runmode=b,
00014           htmlfile=c:\test1.html,
00015           ctext=blue,
00016           fcolor=green);
00017 x 'c:\progra\intern\explor.exe c:\test1.html';
00018

```

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

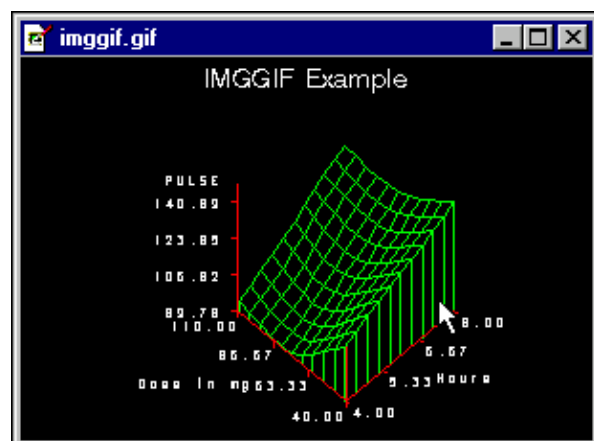
The following drivers are designed to create graphical output used on the web with SAS/Graph software.

Graphical Publishing with GIFs 6.09

```

PROGRAM EDITOR - web5
Command ===>
00001 * The IMGGIF driver is available at Releases 6.09 and above of SAS software.;
00002
00003 %macro imgsize(w=1280,h=1024,dp=95,rows=43,cols=83);
00004 %if &dp<0 %then %put DPI must be greater than zero.;
00005 %else %do;
00006 %options hsize=&wsevalif(&w/&dp)in vsize=&hsevalif(&h/&dp)in
00007 %pse=&cols vpose=&rows;
00008 %end;
00009 %mend imgsize;
00010
00011 filename out 'c:\imggif.gif';
00012 %options resetall reset=global;
00013 %options dev=imggif gfname=out gfname=replace;
00014 %imgsize(w=300,h=200,dp=95,rows=30,cols=50);
00015
00016 proc g3grid data=saved.contour out=work.smooth;
00017 grid dose*regulary=pulse/smooth=1 spline;
00018 run;
00019 title1 "IMGGIF Example";
00020 proc g3d data=work.smooth;
00021 label dose="Dose in mg" regulary="Hours";
00022 plot dose*regulary=pulse / tilt=45 rotate=45 side;
00023 run;
00024
00025 x "C:\Progra\NCommon\NMicros\NPhotoEd\PHOTOED.EXE c:\imggif.gif";

```

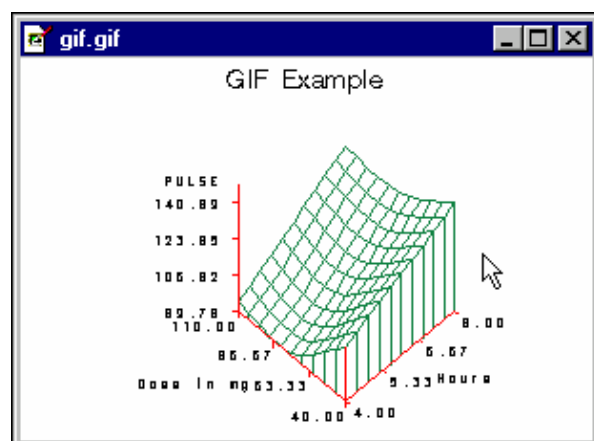


Graphical Publishing with GIFs 6.09e and 6.12

```

PROGRAM EDITOR - web6
Command ===>
00001 * The GIF device driver is available in Releases 6.09e and 6.12 of SAS software.;
00002 * Transparency is supported.;
00003
00004 %macro imgsize(w=1280,h=1024,dp=95,rows=43,cols=83);
00005 %if &dp<0 %then %put DPI must be greater than zero.;
00006 %else %do;
00007 %options hsize=&wsevalif(&w/&dp)in vsize=&hsevalif(&h/&dp)in
00008 %pse=&cols vpose=&rows;
00009 %end;
00010 %mend imgsize;
00011
00012 filename out 'c:\gif.gif';
00013 %options resetall reset=global;
00014 %options dev=gif gfname=out gfname=replace;
00015 %imgsize(w=300,h=200,dp=95,rows=30,cols=50);
00016
00017 proc g3grid data=saved.contour out=work.smooth;
00018 grid dose*regulary=pulse/smooth=1 spline;
00019 run;
00020 title1 "GIF Example";
00021 proc g3d data=work.smooth;
00022 label dose="Dose in mg" regulary="Hours";
00023 plot dose*regulary=pulse / tilt=45 rotate=45 side;
00024 run;
00025
00026 x "C:\Progra\NCommon\NMicros\NPhotoEd\PHOTOED.EXE c:\gif.gif";
00027

```

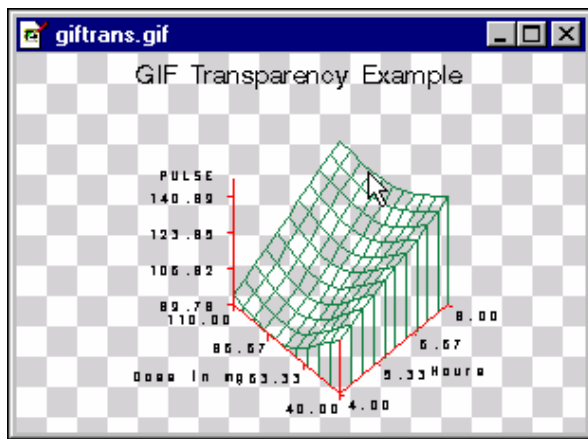


Graphical Publishing with GIFs 6.09e and 6.12 Transparency

```

PROGRAM EDITOR - web7
Command ==>
00001 * The GIF device driver is available in Releases 6.09e and 6.12 of SAS software.;
00002 * Transparency Example.;
00003
00004 libname gdevice0 'c:\temp';
00005 proc gdevice c=gdevice0 device=none;
00006 copy gif from=saselp.devices newname=giftrans;
00007 mod giftrans ucc='01';
00008 quit;
00009
00010 %macro imgsize(w=128,h=128,dpi=95,row=43,col=83);
00011 %if &dpi<=0 %then %put DPI must be greater than zero.;
00012 %else %do;
00013 %options hsize=&w&h&dpi in vsize=&h&dpi in;
00014 %hsize=&w&h&dpi;
00015 %end;
00016 %mend imgsize;
00017
00018 filename out 'c:\giftrans.gif';
00019 options reset=all reset=global;
00020 %options dev=giftrans gdfname=out gdfmode=replace;
00021 %imgsize(w=385,h=200,dpi=95,row=30,col=50);
00022
00023 proc g3grid data=saved.contour out=work.smooth;
00024 grid dose=regularly=pulse/smooth=1 spline;
00025 run;
00026 title1 "GIF Transparency Example";
00027 proc g3d data=work.smooth;
00028 label dose="Dose in mg" regularly="Hours";
00029 plot dose=regularly=pulse / tilt=45 rotate=45 side;
00030 run;
00031
00032 x "C:\Program\Internet\Nt\PhotoEd\PHOTOED.EXE c:\giftrans.gif";
00033

```



Animation with GIFs

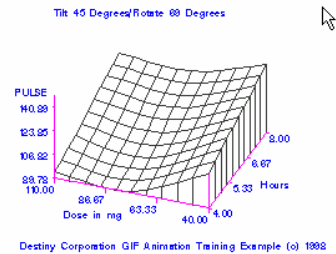
```

PROGRAM EDITOR - web7
Command ==>
00001 *** GIF Animation G3GRID Example;
00002
00003 options reset=all reset=global;
00004
00005 libname saved 'c:\save';
00006
00007 proc g3grid data=saved.contour out=work.smooth;
00008 grid dose=regularly=pulse/smooth=1 spline;
00009 run;
00010
00011 %macro animate;
00012 %let first=1;
00013 %do values=1 to 360 %by 1;
00014 %if &first=1 %then %do;
00015 %options reset=all;
00016 %filename out 'c:\animate.gif';
00017 %options dev=giftrans gdfname=out
00018 gdfmode=replace gdfwidth=
00019 htext=2 gdfrows=0 gdw=1;
00020 %backwidth;
00021 %color=(BLUE MAGENTA DORRBY BROWN);
00022 %footnote1 h=2 j=right "Destiny Corporation GIF Animation Training Example (c) 1998";
00023 %let first=0;
00024 %end;
00025 %else %do;
00026 %options gdfmode=append;
00027 %end;
00028 %if &value=360 %then %do;
00029 %options gdfrows=36;
00030 %end;
00031 title1 "Tilt 45 Degrees/rotate 45 Degrees";
00032 proc g3d data=work.smooth;
00033 label dose="Dose in mg" regularly="Hours";
00034 plot dose=regularly=pulse / tilt=45 rotate=45 side;
00035 run;
00036 %end;
00037 %options reset=all reset=global;
00038 %mend animate;
00039
00040 %animate;
00041
00042 data _null_;
00043 file 'c:\animate.html';
00044 put "<html><title>Animated GIF HTML Example</title>";
00045 put "<body><img src='c:\animate.gif' alt='Look Mom, see what I can do!'></body>";
00046 put "</html>";
00047 run;
00048 x "C:\Program\Internet\Nt\PhotoEd\PHOTOED.EXE c:\animate.html";
00049

```

Animated GIF HTML Example

(Look Mom, see what I can do!)



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.

```

Program Editor - (Untitled)
Command ==>
00001 ods html body='c:\body.html'
00002 contents='c:\contents.html'
00003 page='c:\page.html'
00004 frame='c:\frame.html';
00005
00006 proc freq data=saved.denograp;
00007 table status*gender;
00008 run;
00009 ods html close;
00010
00011

```

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

SAS OUTPUT - Microsoft Internet Explorer

Table of Contents

The FREQ Procedure

Frequency Percent Row Pct Col Pct

Table of STATUS by GENDER

	GENDER			
STATUS	F	M	Total	
D	1	2	3	
	2.86	5.71	8.57	
	33.33	66.67	100.00	
	4.76	14.29		
M	13	6	19	
	37.14	17.14	54.29	
	68.42	31.58	100.00	
	61.90	42.86		
S	5	5	10	
	14.29	14.29	28.57	
	50.00	50.00	100.00	
	23.81	35.71		
SEP	1	1	2	
	2.86	2.86	5.71	
	50.00	50.00	100.00	
	4.76	7.14		
W	4	0	4	
	11.43	0.00	11.43	
	100.00	0.00	100.00	
	4.76	0.00		
Total	21	14	35	
	60.00	40.00	100.00	

Table of Pages

1. The Freq Procedure

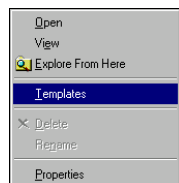
Page 1

Styles in HTML

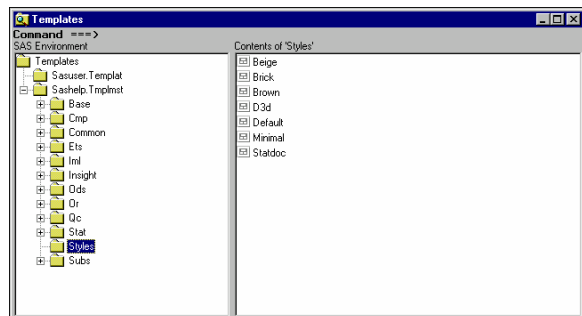
There are several HTML styles that come with SAS. They are designed to allow the user to create an HTML standard of colors, fonts and more across all HTML output. Any of the default ones can be used, or they can be modified to suit your needs. To see the existing styles, go to the Results window and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



Right click and select Templates.



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



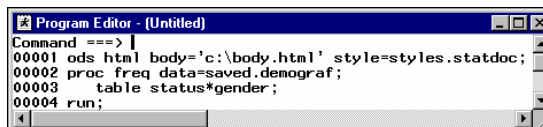
To examine a style, execute Proc Template with the Style name. Default is the default style used.



```
Command ==> |
254 proc template;
255   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',3,Italic)
    'docFont' = ('Arial, Helvetica, Helv',3);
  style color_list /
    'fgB2' = cx0055a9
    'fgB1' = cx004488
    'fgA4' = cx00aaff
    'bgA4' = cx880000
    'bgA3' = cx003333
    'fgA2' = cx0033a9
    'bgA2' = cx000000
    'fgA1' = cx000000
    'bgA1' = cx000000
    'fgA' = cx002288
    'bgA' = cx000000;
  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



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

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of STATUS by GENDER			
	GENDER		Total	
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 66.42 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	

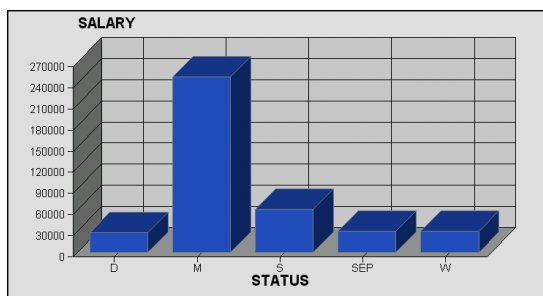
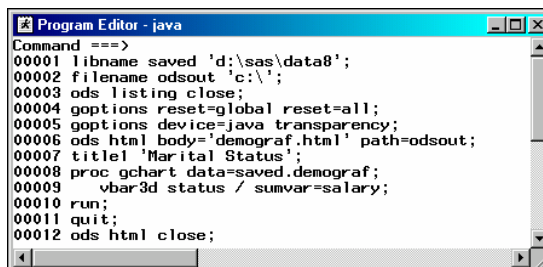
See the online help to learn more about changing styles and what the values mean.

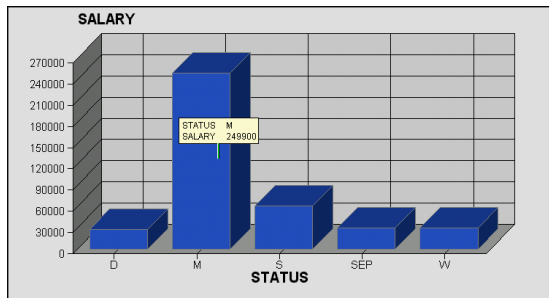
Graphical Support

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

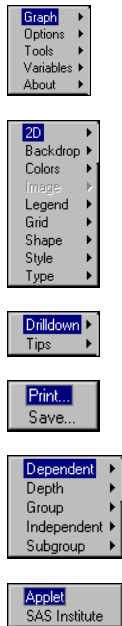
Java Support

There is Java support. Consider the following driver.





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



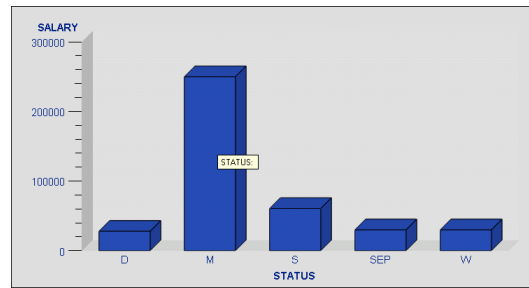
Experiment with each to understand the capabilities.

Active X Support

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

```

Program Editor - activex2
Command ==> |
00001 libname saved 'd:\sas\data8';
00002 filename odsout 'c:\';
00003 ods listing close;
00004 goptions reset=global reset=all;
00005 goptions device=activex transparency;
00006 ods html body='demograf.html' path=odsout;
00007 title1 'Marital Status';
00008 proc gchart data=saved.demograf;
00009     vbar3d status / sumvar=salary;
00010 run;
00011 quit;
00012 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

```
Program Editor - tabcolor
Command ==> |
00001 libname saved 'd:\sas\data8';
00002
00003 ods html body="c:\tabcolor.html";
00004 ods listing close;
00005
00006 title 'Tabulate Color Example using styles';
00007 footnote;
00008 proc tabulate data=saved.demograf;
00009   class status gender / s={foreground=blue};
00010   classlev status gender / s={foreground=yellow};
00011   var salary age / s={foreground=white};
00012   keyword all sum / s={foreground=white};
00013   keylabel all="Total";
00014   table status,gender*(s={foreground=orange});
00015 run;
00016
00017 ods html close;
00018 ods listing;
00019
```

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.

```
Program Editor - tabcolor
Command ==> |
00001 libname saved 'd:\sas\data8';
00002
00003 ods html body="c:\tabcolor.html";
00004 ods listing close;
00005
00006 title 'Tabulate Color Example using styles';
00007 footnote;
00008 proc tabulate data=saved.demograf;
00009   class status gender / s={background=cyan foreground=blue};
00010   classlev status gender / s={foreground=yellow};
00011   var salary age / s={foreground=white};
00012   keyword all sum / s={foreground=white};
00013   keylabel all="Total";
00014   table status,gender*(s={foreground=orange font_style=italic})*salary;
00015 run;
00016
00017 ods html close;
00018 ods listing;
00019
```

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

```
Program Editor - tabcolor
Command ==> |
00001 libname saved 'd:\sas\data8';
00002
00003 ods html body="c:\report.html";
00004 ods listing close;
00005
00006 title 'Report Color Example using styles';
00007 footnote;
00008 proc report data=saved.demograf nowindows;
00009   column gender status age salary;
00010   define gender / group style={foreground=green};
00011   define status / group style={foreground=blue};
00012   define age / mean style={foreground=red};
00013   define salary / sum style={foreground=brown};
00014 run;
00015
00016 ods html close;
00017 ods listing;
00018
```

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

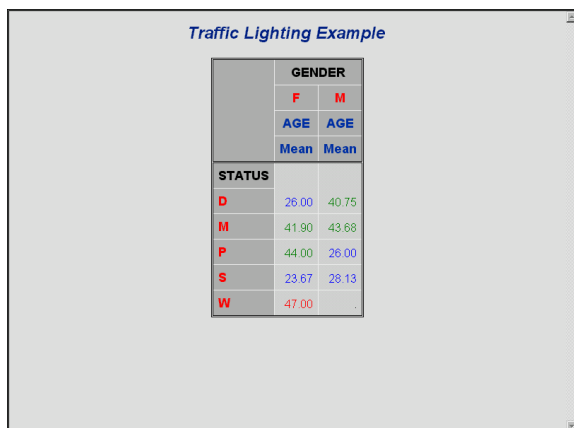
Traffic Lighting

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.

```

Program Editor - traffic
Command ==>
00001 ods html body="c:\traffic";
00002 ods listing close;
00003
00004 title 'Traffic Lighting Example';
00005 proc format;
00006   value agefmt
00007     1-18 = 'cx4DAEGE'
00008     19-30 = 'Blue'
00009     31-45 = 'Green'
00010     46-65 = 'Red'
00011     other = 'Gray';
00012 run;
00013
00014 proc tabulate data=saved.demogius style={foreground=agefmt.};
00015   class status gender / s={foreground=black};
00016   classlev status gender / s={foreground=red};
00017   var age;
00018   table status,gender*age*mean;
00019 run;
00020
00021 ods html close;
00022 ods listing;

```



Hyperlinks in Graphs and Reports

Hyperlinks in Graphs

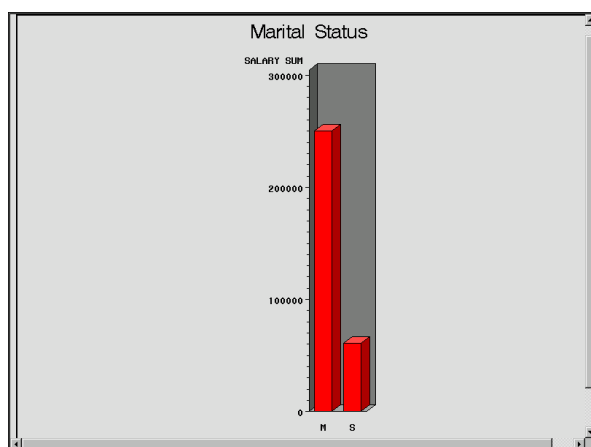
In Version 8 of SAS Software, HTML examples that link to each other are possible. Wouldn't it be great to create a graph and then be able to click on the items in the graph to branch to appropriate detail about those items?

The following code could create that result.

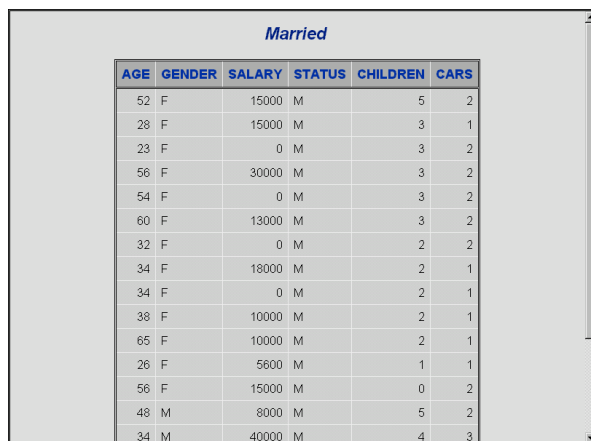
```

Program Editor - graphhref
Command ==>
00001 libname saved 'd:\sas\data8';
00002 filename odsout 'c:\';
00003 ods listing close;
00004 options reset-global;
00005 data work.demograf;
00006   length linkme $ 40;
00007   set saved.demograf;
00008   if status='S' then linkme = 'href="single.html"';
00009   else if status='M' then linkme = 'href="married.html"';
00010 run;
00011 options device=gif transparency;
00012 ods html body='demograf.html' path=odsout;
00013 title 'Marital Status';
00014 proc gchart data=work.demograf;
00015   vbar3d status / sumvar=salary
00016   htab=linkme;
00017   where status in ('S','M');
00018 run;
00019 quit;
00020
00021 ods html body='single.html' path=odsout;
00022 title 'Singles';
00023 proc print data=work.demograf(drop=linkme) noobs;
00024   where status='S';
00025 run;
00026
00027 ods html body='married.html' path=odsout;
00028 title 'Married';
00029 proc print data=work.demograf(drop=linkme) noobs;
00030   where status='M';
00031 run;
00032 ods html close;
00033 ods listing;
00034

```



Click on the appropriate bars to see the detail.



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:

```

Program Editor - tabuhref
Command ==> |
00001 ods html    body="c:\traffic";
00002 ods listing close;
00003 title 'Hyperlinks in Procedures';
00004 proc format;
00005   value agefmt
00006     1-18  = 'cx4DAEGE'
00007     19-30 = 'Blue'
00008     31-45 = 'Green'
00009     46-65 = 'Red'
00010     other = 'Gray';
00011   value $status
00012     'D' = '<A href=divorced.html>Divorced</A>'
00013     'M' = '<A href=married.html>Married</A>'
00014     'W' = '<A href=widowed.html>Widowed</A>'
00015     'S' = '<A href=single.html>Single</A>'
00016     'P' = '<A href=separated.html>Separated</A>'
00017 ;
00018 run;
00019 proc tabulate data=saved.demogius style={foreground=agefmt.};
00020   class status gender / s={foreground=black};
00021   classlev status gender / s={foreground=red};
00022   var age ;
00023   table status,gender*age*mean;
00024   format status $status.;
00025 run;
00026 ods html close;
00027
00028 **** Not all status prints are shown;
00029 ods html body="c:\divorced.html";
00030 proc print data=saved.demogius;
00031   where status='D';
00032 run;
00033 ods html close;
00034
00035 ods listing;
00036

```

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. Copyright © 2000. This material may not be duplicated in any way. Please contact Destiny Corporation for more information.