

# Simple Ways To Publish Reports And Graphs On The Web

Prepared by



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

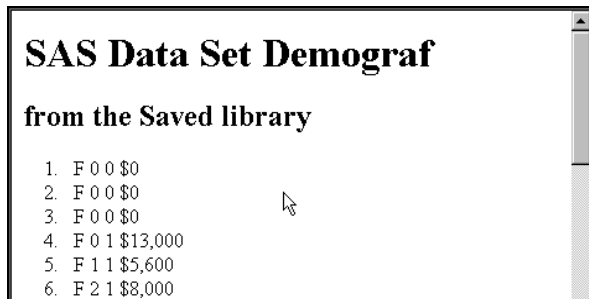
Web: [www.destinycorp.com](http://www.destinycorp.com) Email: [info@destinycorp.com](mailto:info@destinycorp.com)

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.

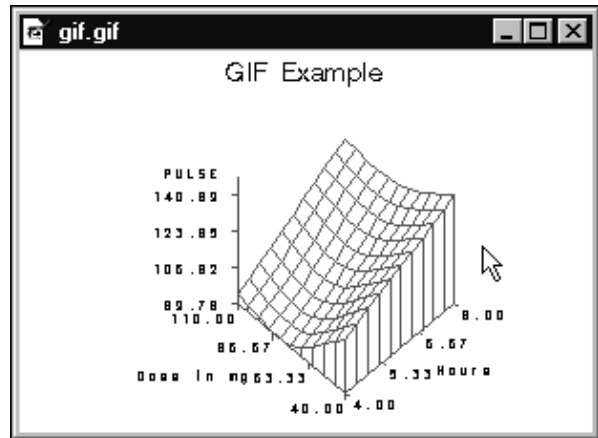
```
PROGRAM EDITOR - web1
Command ==>
00001 libname saved 'c:\sas';
00002 data _null_;
00003 set saved.demograf end=x;
00004 file 'c:\datanull.html';
00005 if _n_ = 1 then do;
00006 put '<h1>SAS Data Set Demograf</h1>';
00007 / '<h2>from the Saved library</h2>';
00008 / '<ol>';
00009 end;
00010 put '<li>' gender children cars salary dollar8.;
00011 if x then put '';
00012 run;
00013 x 'c:\progra\intern\explore.exe c:\datanull.html';
00014
```



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

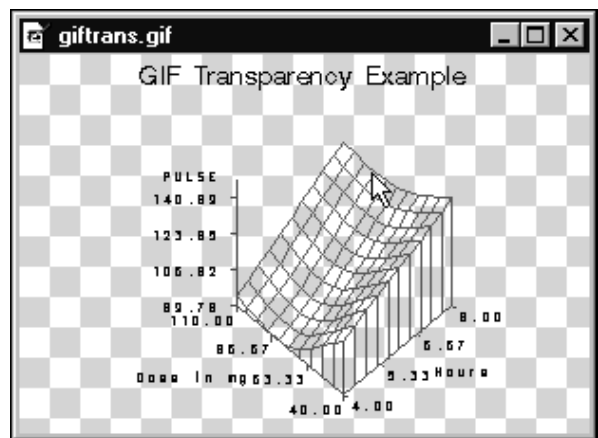
## Graphical Publishing with GIFs

```
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=128,h=1024,dpi=95,rows=43,cols=83);
00005 %if &dpi<=0 %then %put DPI must be greater than zero.;
00006 %else %do;
00007 %options hsize=%sysvalif(&w/&dpi)in vsize=%sysvalif(&h/&dpi)in
00008 %pss=%cols vpost=%rows;
00009 %end;
00010 %mend imgsize;
00011
00012 filename out 'c:\gif.gif';
00013 %options reset=all reset=global;
00014 %options device=ggif gfname=out gfnameout gfnmode=replace;
00015 %imgsize(w=300,h=200,dpi=95,rows=30,cols=50);
00016
00017 proc g3grid data=saved.contour out=work.smooth;
00018 grid dose=regularly=pulse/smooth=1 spline;
00019 run;
00020 title "GIF Example";
00021 proc g3d data=work.smooth;
00022 label dose=base in ng regularly=Hours;
00023 plot dose=regularly=pulse / tilt=45 rotate=45 side;
00024 run;
00025
00026 x "c:\progra\common\microsoft\photoed\PHOTOED.EXE c:\gif.gif";
00027
```



## Graphical Publishing with GIFs 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 gdevice 'c:\temp';
00005 proc gdevice=gdevice device=giftrans;
00006 copy gif from=sashelp.devices newname=giftrans;
00007 mod giftrans ucc='01x';
00008 quit;
00009
00010 %macro imgsize(w=128,h=1024,dpi=95,rows=43,cols=83);
00011 %if &dpi<=0 %then %put DPI must be greater than zero.;
00012 %else %do;
00013 %options hsize=%sysvalif(&w/&dpi)in vsize=%sysvalif(&h/&dpi)in
00014 %pss=%cols vpost=%rows;
00015 %end;
00016 %mend imgsize;
00017
00018 filename out 'c:\giftrans.gif';
00019 %options reset=all reset=global;
00020 %options device=ggiftrans gfname=out gfnameout gfnmode=replace;
00021 %imgsize(w=300,h=200,dpi=95,rows=30,cols=50);
00022
00023 proc g3grid data=saved.contour out=work.smooth;
00024 grid dose=regularly=pulse/smooth=1 spline;
00025 run;
00026 title "GIF Transparency Example";
00027 proc g3d data=work.smooth;
00028 label dose=base in ng regularly=Hours;
00029 plot dose=regularly=pulse / tilt=45 rotate=45 side;
00030 run;
00031
00032 x "c:\progra\common\microsoft\photoed\PHOTOED.EXE c:\giftrans.gif";
00033
```

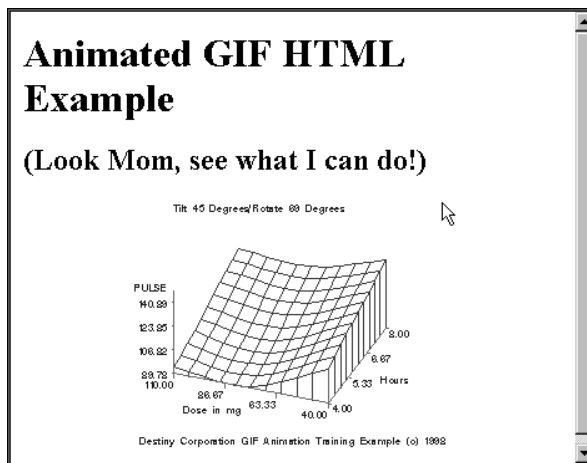


### Animation with GIFs

```

PROGRAM EDITOR - web8
Command ==>
00001 *** GIF Animation G3ORID Example;
00002
00003 goptions reset=all reset=global;
00004
00005 libname saved 'c:\save';
00006
00007 proc g3orid data=saved.contour out=work.smooth;
00008   grid dose=regular:pulse/smooth; spline;
00009 run;
00010
00011 %macro animate;
00012   %let first=1;
00013   %do %eval(1 + 360 * %xby 1);
00014     %if %first = 1 %then %do;
00015       goptions reset=all;
00016       filename out 'c:\animate.gif';
00017       goptions dev=gifanim getnames=out;
00018       gformat=propic4 float=width;
00019       ht=2; gcpl=0; %wait;
00020       %oback=1;
00021       colors=(BLUE MAGENTA DRABRY BRDN);
00022       footnote h=2 j=right "Destiny Corporation GIF Animation Training Example (c) 1998";
00023       %let first=0;
00024     %end;
00025     %do %eval(2 + 360 * %xby 1);
00026       goptions getmode=append;
00027     %end;
00028     %if %eval(1 + 360 * %xby 1) = 360 %then %do;
00029       goptions getmode="360";
00030     %end;
00031     title4 "Tilt 45 Degrees/Rotate 90 Degrees";
00032     proc g3d data=work.smooth;
00033       label dose="Dose in mg" regular="Hours";
00034       plot dose=regular:pulse / tilt=45 rotate=svalue side;
00035     run;
00036   %mend animate;
00037   goptions reset=all reset=global;
00038 %macro animate;
00039 %animate
00040
00041 data _null_;
00042 file 'c:\animate.html';
00043 put <html><title>Animated GIF HTML Example</title>;
00044   <h2>(Look Mom, see what I can do!)</h2>;
00045   <img src='c:\animate.gif' />;
00046 run;
00047 x 'c:\program\intern\iexplore.exe c:\animate.html';
00048
00049

```



### 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.demogrf;
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.HTML file pulls them all together.

The FREQ Procedure

Table of STATUS by GENDER

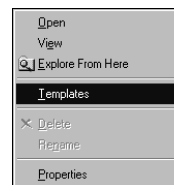
STATUS	GENDER			Total
	F	M		
D	1	2	3	8.57
	2.86	5.71		33.33
	4.76	14.29		61.90
M	13	6	19	54.29
	37.14	17.14		68.42
	61.90	31.58		119.00
S	5	5	10	28.57
	14.29	14.29		50.00
	23.81	35.71		100.00
SEP	1	1	2	5.71
	2.86	2.86		50.00
	4.76	7.14		100.00
W	1	0	1	2.86
	100.00	0.00		47.60
	60.00	40.00		100.00
Total	21	14	35	100.00

### Styles in HTML

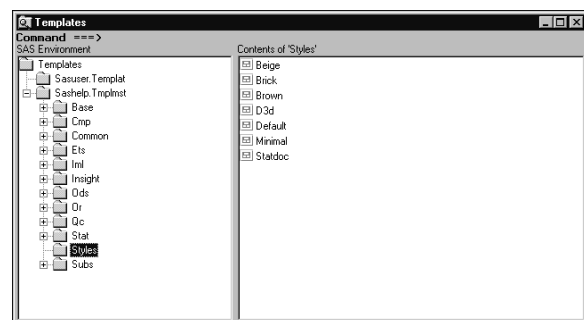
There are several HTML styles that come with SAS. They are designed to allow the user to create an HTML standard of colors, fonts and more across all HTML output. Any of the default ones can be used, or they can be modified to suit your needs. To see the existing styles, go to the Results window and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

```

Program Editor - (Untitled)
Command ==>
00001 proc template;
00002   source styles.default;
00003 run;
00004

```

```

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',4,Bold)
    'docFont' = ('Arial, Helvetica, Helv',3);
  style color_list /
    'fgB2' = cx0066AA
    'fgB1' = cx004488
    'fgA4' = cxAAFFAA
    'bgA4' = cx880000
    'bgA3' = cxD3D3D3
    'fgA2' = cx0033AA
    'bgA2' = cx808080
    'fgA1' = cx000000
    'bgA1' = cxFF9900
    'fgA' = cx002288
    'bgA' = cxE0E0E0;
  style colors /
    'headerfgenph' = color_list('fgA2')
    'headerbgenph' = color_list('bgA2')
    'headerfstrong' = color_list('fgA2')
    'headerbstrong' = 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

```

Program Editor - (Untitled)
Command ==> |
00001 ods html body='c:\body.html' style=styles.statdoc;
00002 proc freq data=saved.demograf;
00003   table status*gender;
00004 run;

```

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

The FREQ Procedure

STATUS	GENDER		Total
	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	5	5	10
	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	
	4.76	7.14	
W	1	0	1
	2.86	0.00	2.86
	100.00	0.00	
	4.76	0.00	
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.

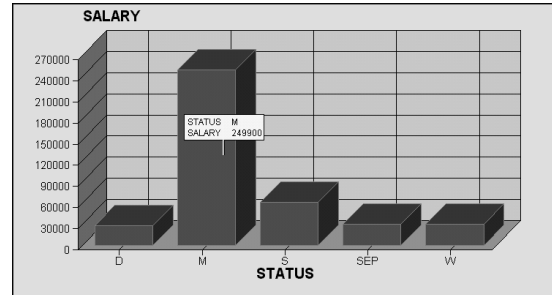
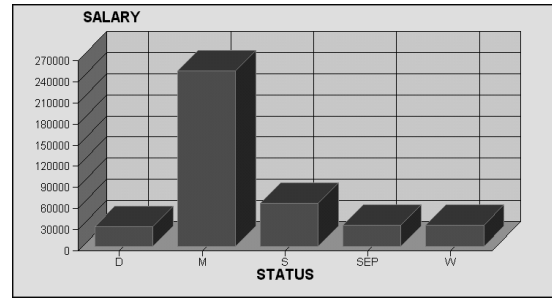
**Java Support**

There is Java support. Consider the following driver.

```

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

```



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

- Graph ▶
- Options ▶
- Tools ▶
- Variables ▶
- About ▶

- 2D ▶
- Backdrop ▶
- Colors ▶
- Image ▶
- Legend ▶
- Grid ▶
- Shape ▶
- Style ▶
- Type ▶

- Drilldown ▶
- Tips ▶

- Print... ▶
- Save... ▶

- Dependent ▶
- Depth ▶
- Group ▶
- Independent ▶
- Subgroup ▶

- Applet ▶
- SAS Institute ▶

Experiment with each to understand the capabilities.

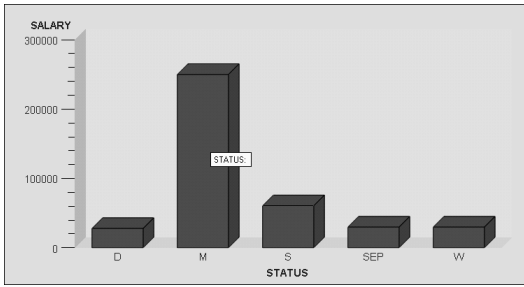
**Active X Support**

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 title '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.

- File
  - View
  - Graph
  - Axis
  - Legend
  - Explore
  - Variables
  - About
  - Properties...
- SaveAs...
  - Print...
  - CopyToClipboard
  - PasteFromClipboard
- 2D View
  - Reset Eyepoint
- Appearance
  - Lighting...
  - Type
- Labels
  - Orientation
  - Shape
  - Wall...
  - Background...
  - Color Scheme...
- Bar
  - Box
  - Contour
  - High Low
  - Map
  - Plot
  - Pipe
  - Surface
- Horizontal
  - Vertical
  - Depth
- Border
  - Position
  - Title...
  - Text...
  - Visible
- Chart Tips
  - Drilldown
  - Scrollbars
  - Mouse
- Variables...

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}

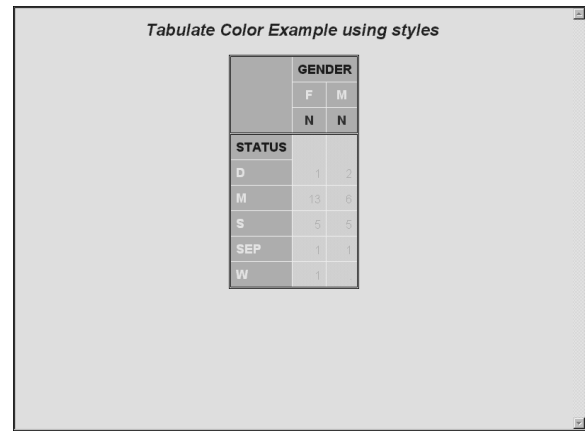
At various locations to specify colors.

Current supported Procedures are Tabulate and Report.

### Styles in Proc Tabulate

```

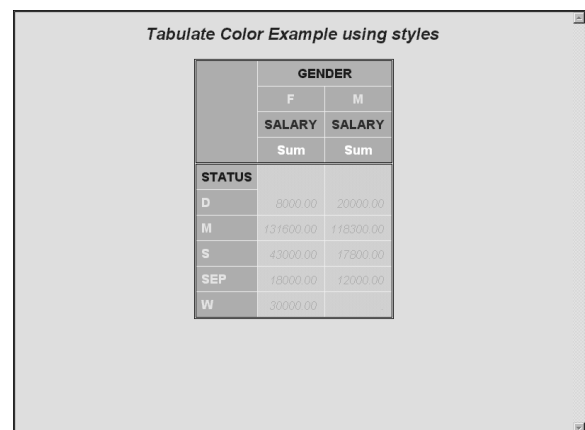
Program Editor - tabcolor
Command ==>
00001 libname saved 'd:\sas\data8';
00002
00003 ods html body='c:\tabcolor.html';
00004 ods listing close;
00005
00006 title 'Tabulate Color Example using styles';
00007 footnote;
00008 proc tabulate data=saved.demograf;
00009 class status gender / s={foreground=blue};
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
    
```



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
    
```



### 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
M	SEP	23	18000
	W	46	30000
	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 = 'x4DAGEE'
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;
  
```

**Traffic Lighting Example**

	GENDER	
	F	M
STATUS	AGE	AGE
	Mean	Mean
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

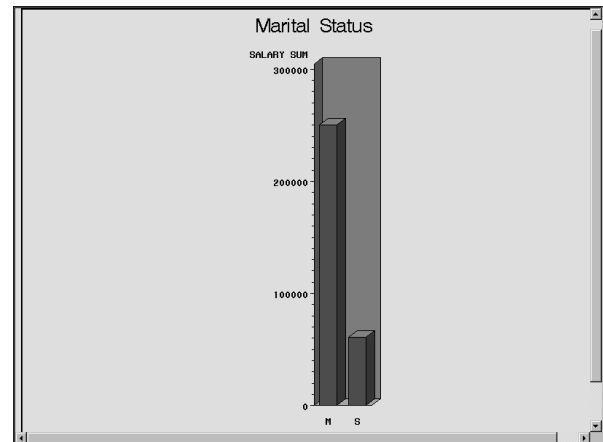
#### Hyperlinks in Graphs

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.

```

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   html=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.

**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

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

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.

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 hyperlinks. 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 = 'c:\4DAGEE'
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

```

	GENDER	
	F	M
AGE	AGE	
	Mean	Mean
STATUS		
<a href="#">Divorced</a>	26.00	40.75
<a href="#">Married</a>	41.90	43.68
<a href="#">Separated</a>	44.00	26.00
<a href="#">Single</a>	23.67	28.13
<a href="#">Widowed</a>	47.00	.