

ODS (Output Delivery System)

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.

```

Command ===>
00001 libname saved 'c:\sass';
00002 data _null_;
00003 set saved.demograf;
00004 file 'c:\data\demo1.html';
00005 if _n_ = 1 then do;
00006 put '<h1>SAS Data Set Demograf</h1>';
00007 put '<h2>from the Saved library</h2>';
00008 put '<br>';
00009 and;
00010 put '<table>';
00011 if x then put '<tr><td>GENDER</td><td>STATUS</td><td>CHILDREN</td><td>SALARY</td></tr>';
00012 run;
00013 x 'c:\program\intern\intern\explore.exe c:\data\demo1.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.

```

Command ===>
00001 libname saved 'c:\sass';
00002 data work.demograf;
00003 set saved.demograf;
00004 format salary dollar10.;
00005 run;
00006
00007 title1 'Data Set Table';
00008
00009 odshtml(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\intern\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 OUT2HTML

Any procedure's output can be captured and published with the OUT2HTML macro.

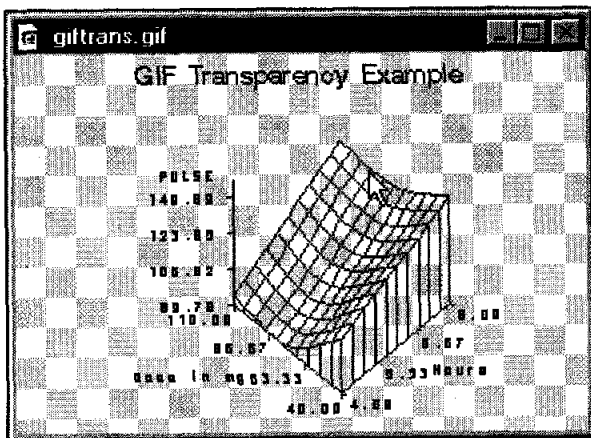
```

Command ===>
00001 libname saved 'c:\sass';
00002 out2html(capture=off);
00003
00004 title1 'Proc Report Output';
00005 odshtml(data=saved.demograf,
00006 where=status ^= 'SEP',
00007 var=gender status children salary,
00008 htmlfile=c:\test1.html,
00009 runmode=b);
00010
00011 x 'c:\program\intern\intern\explore.exe c:\test1.html';
00012

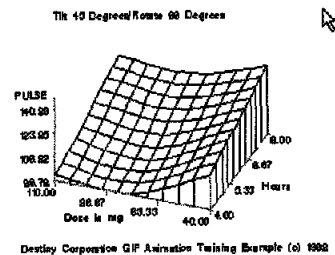
```

Proc Report Output

GENDER	STATUS	AGE	SALARY	CHILDREN	CARS
F	D	29	\$000	2	1
F	D	40	\$11,000	1	20
F	S	17	\$0000	0	2
F	SEP	23	\$0000	1	1
F	M	46	\$0000	1	1
F		25	\$13,000	0	25
F	D	35	\$0000	0	2
F	S	27	\$13,000	10	9
F	S	27	\$0000	0	2
F	SEP	50	\$2,000	2	1
F		31	\$5,600	25	15

[illegible][illegible]

(Look Mom, see what I can do!)



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

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.

```

000001 Command ==>
000002 body='c:\body.html'
000003 contents='c:\contents.html'
000004 page='c:\page.html'
000005 frame='c:\frame.html';
000006
000007 proc freq data=saved.demograf;
000008   table status*gender;
000009 run;
000010 ods html close;
000011

```

The FRED Procedure

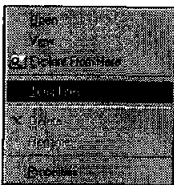
Frequency Category New Fed	Year of Activity by Education			
	High School	2	4	5+ yrs
V	280	15	5	8.91
	37.33	68.07		
	8.78	14.26		
M	11	5		10
	37.44	17.88		54.29
	68.40	31.60		
	94.56	43.88		
S	5	5		10
	16.28	14.29		28.67
	80.00	85.00		
	28.61	35.71		
None	5	1		7
	285	3.85		5.71
	36.60	5.00		
	5.00	1.00		
All	1	5		7
	7.40	68.00		2.90
	167.00	3.00		
	14.70	5.00		
Total	71	54		35
	66.50	46.00		200.00

Styles in HTML

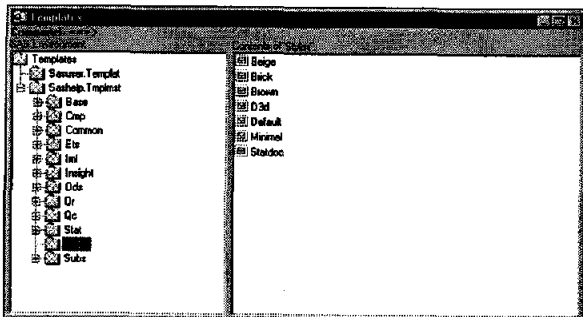
There are several HTML styles that come with SAS. They are designed to allow the user to create an HTML standard of colors, fonts and more across all HTML output. Any of the default ones can be used, or they can be modified to suit your needs. To see the existing styles, go to the Results window and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



Right click and select Templates.



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



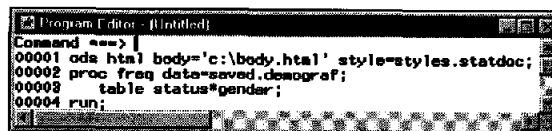
To examine a style, execute Proc Template with the Style name. Default is the default style used.



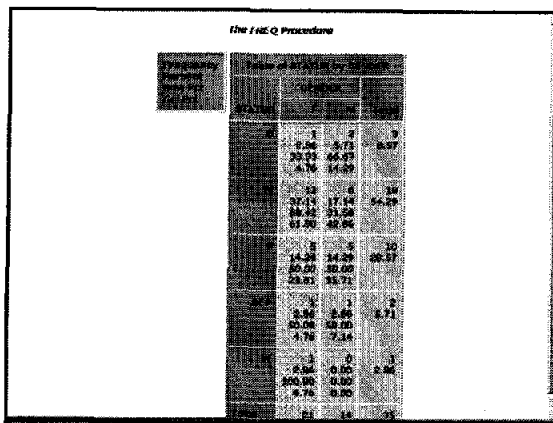
```
Command ==>
254 proc template;
255   source styles.default;
define style Styles.Default;
  style fonts /
    'TitleFont2' = ('Arial, Helvetica, Helv',4,Bold,Italic)
    'TitleFont' = ('Arial, Helvetica, Helv',5,Bold,Italic)
    'StrongFont' = ('Arial, Helvetica, Helv',4,Bold)
    'EmphasisFont' = ('Arial, Helvetica, Helv',3,Italic)
    'FixedEmphasisFont' = ('Courier,3,Italic)
    'FixedStrongFont' = ('Courier,4,Bold)
    'FixedHeadingFont' = ('Courier,4)
    'HeadingEmphasisFont' = ('Arial, Helvetica, Helv',4,Bold,Italic)
    'HeadingFont' = ('Arial, Helvetica, Helv',4,Bold)
    'docFont' = ('Arial, Helvetica, Helv',3);
  style color_list /
    'fg02' = c000000
    'fg01' = c000000
    'fg04' = c000000
    'bg01' = c000000
    'bg02' = c000000
    'bg03' = c000000
    'bg04' = c000000
    'fg01' = c000000
    'bg01' = c000000
    'fg01' = c000000
    'bg01' = c000000;
  style colors /
    'headerfontph' = color_list('fg02')
    'headerfontph' = color_list('bg02')
    'headerfontstrong' = color_list('fg02')
    'headerfontstrong' = color_list('bg02');
```

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)



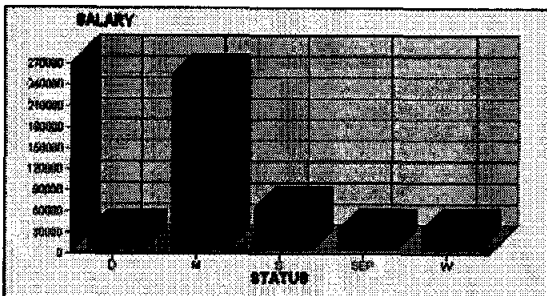
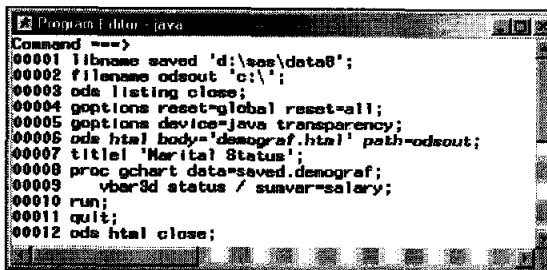
See the online help to learn more about changing styles and what the values mean.

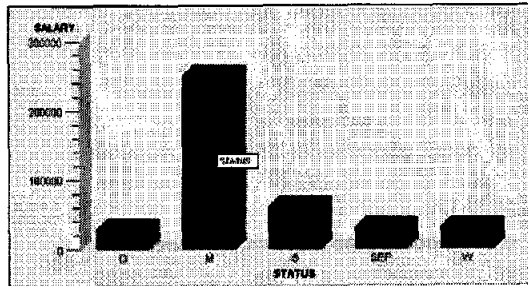
Graphical Support

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

Java Support

There is Java support. Consider the following driver.





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 --->
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 classvar status gender / s={foreground=yellow};
00011 var salary age;
00012 bygroup all sum / s={foreground=blue};
00013 bylabel all="Total";
00014 table status,gender*(s={foreground=white});
00015 run;
00016
00017 ods html close;
00018 ods listing;
00019

```

Tabulate Color Example using styles

GENDER	
STATUS	Sum
D	29
M	42932071
S	171
SEP	23
W	46
Total	845

Additional examples use background and font_style.

```

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=white foreground=blue};
00010 classvar status gender / s={background=white};
00011 var salary age;
00012 bygroup all sum / s={background=white};
00013 bylabel all="Total";
00014 table status,gender*(s={background=white font_style=italic})$salary;
00015 run;
00016
00017 ods html close;
00018 ods listing;
00019

```

Tabulate Color Example using styles

GENDER	
STATUS	Sum
D	29
M	42932071
S	171
SEP	23
W	46
Total	845

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	6900
	M	42932071	15600
	S	171	43000
	SEP	23	19000
	W	46	50000
M	D	845	20000
	M	26900000	116500
	S	103	27000
	SEP	55	12000

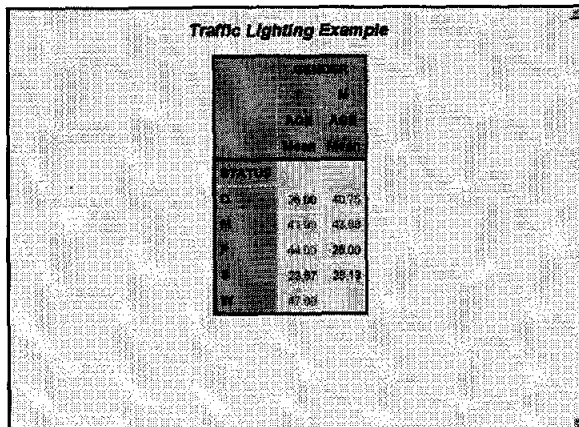
Traffic Lighting

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.

```

Command ===>
00001 ods html body="c:\traffic";
00002 ods listing close;
00003
00004 title 'Traffic Lighting Example';
00005 proc format;
00006   value agefat
00007     1-18 = 'ck4AGEE'
00008     19-30 = 'Blue'
00009     31-45 = 'Green'
00010     46-65 = 'Red'
00011     other = 'Gray';
00012 run;
00013
00014 proc tabulate data=saved.demograf style=[foreground=agefat.];
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

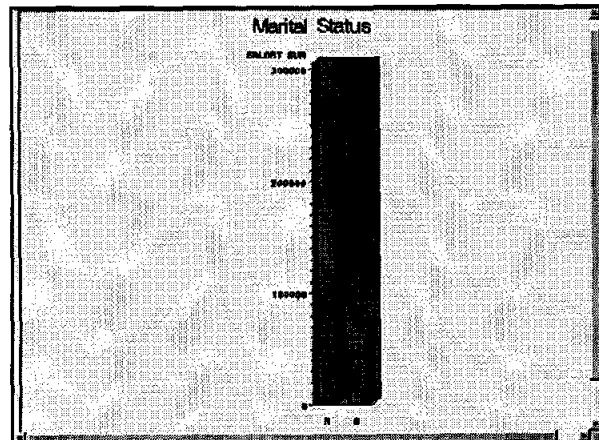
In Version 8 of SAS Software, HTML examples that link to each other are possible. Wouldn't it be great to create a graph and then be able to click on the items in the graph to branch to appropriate detail about those items?

The following code could create that result.

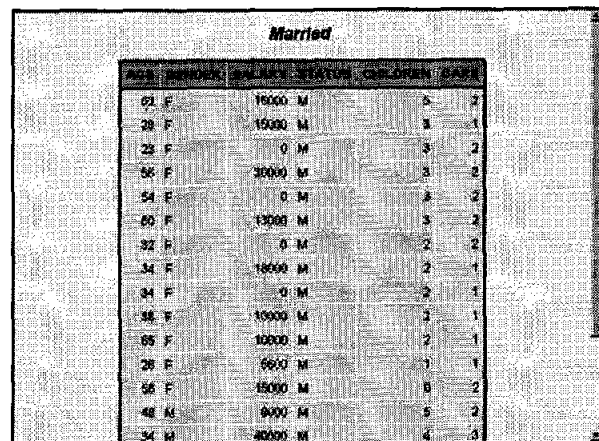
```

Command ===>
00001 libname saved 'd:\sase\data8';
00002 filename odsout 'c:\';
00003 ods listing close;
00004 options reset=global;
00005 data work.demograf;
00006   length linkas $ 40;
00007   set saved.demograf;
00008   if status='S' then linkas = 'href="single.html"';
00009   else if status='M' then linkas = '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   vbar2 status / sumvar=salary;
00016   html=linkas;
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=linkas) 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=linkas) 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	CARE
12	F	0	S	0	0
16	F	0	S	0	0
19	F	0	S	0	0
22	F	10000	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
24	M	7000	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 ==>
00001 ods html body="c:\traffic";
00002 ods listing close;
00003 title 'Hyperlinks in Procedures';
00004 proc format;
00005   value agefst
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=agefst.};
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

STATUS		
	AGE	AGE
	Mean	Mean
Divorced	28.00	40.75
Married	41.00	43.88
Widowed	44.00	26.00
Single	23.67	26.13
Separated	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.