

But Our Office *Isn't* Paperless—

Can I get all those nifty styles, too?

Or

Using ODS to Create High-quality Hardcopy in SAS v8

Brian T. Schellenberger, SAS Institute Inc., Cary, NC

[The following is a table, showing the fonts normally available to ODS PRINTER code, that was produced directly by SAS, with PROC REPORT using the ODS PRINTER destination. Not exactly a green-bar listing, is it?]

font_face	plain	<i>italic</i>	bold	<i>bold_italic</i>
Times	plain ABCdef123	<i>italic ABCdef123</i>	bold ABCdef123	<i>bold-italic ABCdef12</i>
Helvetica	plain ABCdef123	<i>italic ABCdef123</i>	bold ABCdef123	<i>bold-italic ABCdef12</i>
Courier	plain ABCdef123	<i>italic ABCdef123</i>	bold ABCdef123	<i>bold-italic ABCdef12</i>
Σψμβολ	πλαιν ABXδεφ123	ιταλιχ ABXδεφ123	βολδ ABXδεφ123	βολδ-ιταλιχ ABXδεφ12
Bookman	plain ABCdef123	<i>italic ABCdef123</i>	bold ABCdef123	<i>bold-italic ABCdef12</i>
New Century Schoolbook	plain ABCdef123	<i>italic ABCdef123</i>	bold ABCdef123	<i>bold-italic ABCdef12</i>
Palatino	plain ABCdef123	<i>italic ABCdef123</i>	bold ABCdef123	<i>bold-italic ABCdef12</i>
<i>Chancery</i>	<i>plain ABCdef123</i>	<i>italic ABCdef123</i>	<i>bold ABCdef123</i>	<i>bold-italic ABCdef12</i>
✦✧✪✫✬✭✮✯	☐●●✪✫	✪▼●●✪✫	☐●●✪✫	☐●●✪✫▼●●✪✫
	✧✦✧✪✫✬✭✮✯	✧✦✧✪✫✬✭✮✯	✧✦✧✪✫✬✭✮✯	✧✦✧✪✫✬✭✮✯

INTRODUCTION

Many of you have seen the nifty styles, highlighting, and other effects available from the Output Delivery System, a new feature of the SAS® System in v7. When we demonstrate this capability, we almost always show it off using HTML in a web browser. There are a number of reasons for this—it's a very popular technology, it looks good on the computer screen, we have an embedded browser window in the Windows version of SAS, and so forth.

But for many of you, it doesn't solve a lot of the problems you face in real life, because you need to produce actual hardcopy or "hardcopy-like" output, such as PDF or PostScript documents. This paper examines alternatives available to you to get the sort of quality output you need in the formats you require while still being able to get the "stylish" sorts of results you've seen in the HTML-based demos.

SOLUTIONS

One possible solution, of course, is to simply use the v7 HTML capabilities to put the output into the browser, and then use the browser itself to save as PostScript. This actually works reasonably well for small tables, but the browsers do not do a good job when the table is too long or too wide.

Newly available in production quality for version 8 of the SAS System is the ODS PRINTER output destination, which allows you produce PostScript or PCL directly. In addition, when using the SAS System under the Windows operating system, ODS PRINTER code will use the Windows printers installed on your system. If you install Adobe Acrobat (available commercially from Adobe), then it will produce PDF files by using the PDFWriter and/or Acrobat Distiller components of Adobe Acrobat.

In addition, there are three experimental destinations that may be helpful for producing hardcopy output: RTF for Microsoft Word, LaTeX and XML destinations.

Regardless of which of these destinations you choose, the Output Delivery System uses *table templates* and *style templates* to control the appearance of the output.

Browsers

As any of you who have used the printing capabilities of today's browsers are no doubt aware, they have

some severe limitations when it comes to the production of hardcopy output. For example, here is a rather large table as rendered by Internet Explorer.

Obs	value1	value2	value3	value4	value5	value6	value7	value8	value9	value10
1	101	102	103	104	105	106	107	108	109	110
2	201	202	203	204	205	206	207	208	209	210
3	301	302	303	304	305	306	307	308	309	310
4	401	402	403	404	405	406	407	408	409	410
5	501	502	503	504	505	506	507	508	509	510
6	601	602	603	604	605	606	607	608	609	610
7	701	702	703	704	705	706	707	708	709	710
8	801	802	803	804	805	806	807	808	809	810
9	901	902	903	904	905	906	907	908	909	910
10	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010
11	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110
12	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210
13	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310
14	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410
15	1501	1502	1503	1504	1505	1506	1507	1508	1509	1510
16	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610
17	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710
18	1801	1802	1803	1804	1805	1806	1807	1808	1809	1810
19	1901	1902	1903	1904	1905	1906	1907	1908	1909	1910
20	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
21	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110
22	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210
23	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310
24	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410
25	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510
26	2601	2602	2603	2604	2605	2606	2607	2608	2609	2610
27	2701	2702	2703	2704	2705	2706	2707	2708	2709	2710
28	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810
29	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910
30	3001	3002	3003	3004	3005	3006	3007	3008	3009	3010
31	3101	3102	3103	3104	3105	3106	3107	3108	3109	3110
32	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210
33	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310
34	3401	3402	3403	3404	3405	3406	3407	3408	3409	3410
35	3501	3502	3503	3504	3505	3506	3507	3508	3509	3510
36	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610
37	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710
38	3801	3802	3803	3804	3805	3806	3807	3808	3809	3810
39	3901	3902	3903	3904	3905	3906	3907	3908	3909	3910
40	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010
41	4101	4102	4103	4104	4105	4106	4107	4108	4109	4110
42	4201	4202	4203	4204	4205	4206	4207	4208	4209	4210
43	4301	4302	4303	4304	4305	4306	4307	4308	4309	4310
44	4401	4402	4403	4404	4405	4406	4407	4408	4409	4410
45	4501	4502	4503	4504	4505	4506	4507	4508	4509	4510
46	4601	4602	4603	4604	4605	4606	4607	4608	4609	4610
47	4701	4702	4703	4704	4705	4706	4707	4708	4709	4710
48	4801	4802	4803	4804	4805	4806	4807	4808	4809	4810
49	4901	4902	4903	4904	4905	4906	4907	4908	4909	4910
50	5001	5002	5003	5004	5005	5006	5007	5008	5009	5010
51	5101	5102	5103	5104	5105	5106	5107	5108	5109	5110
52	5201	5202	5203	5204	5205	5206	5207	5208	5209	5210
53	5301	5302	5303	5304	5305	5306	5307	5308	5309	5310
54	5401	5402	5403	5404	5405	5406	5407	5408	5409	5410
55	5501	5502	5503	5504	5505	5506	5507	5508	5509	5510
56	5601	5602	5603	5604	5605	5606	5607	5608	5609	5610
57	5701	5702	5703	5704	5705	5706	5707	5708	5709	5710
58	5801	5802	5803	5804	5805	5806	5807	5808	5809	5810
59	5901	5902	5903	5904	5905	5906	5907	5908	5909	5910
60	6001	6002	6003	6004	6005	6006	6007	6008	6009	6010

Not only are various rules missing, but the titles fail to repeat on the second page as one would expect. Even worse, most of the columns of this wide table are missing entirely.

ODS Printer

New in v8 is the ODS PRINTER destination. (Actually, it is also available experimentally in v7.) This destination is designed for printing and produces results like this (the first and last pages are shown):

The SAS System 13:03 Wednesday, May 26, 1999 1

Obs	value1	value2	value3	value4	value5	value6	value7	value8	value9	value10	value11	value12	value13
1	101	102	103	104	105	106	107	108	109	110	111	112	113
2	201	202	203	204	205	206	207	208	209	210	211	212	213
3	301	302	303	304	305	306	307	308	309	310	311	312	313
4	401	402	403	404	405	406	407	408	409	410	411	412	413
5	501	502	503	504	505	506	507	508	509	510	511	512	513
6	601	602	603	604	605	606	607	608	609	610	611	612	613

Obs	value14	value15	value16	value17	value18	value19	value20	value21	value22	value23	value24	value25
1	114	115	116	117	118	119	120	121	122	123	124	125
2	214	215	216	217	218	219	220	221	222	223	224	225
3	314	315	316	317	318	319	320	321	322	323	324	325
4	414	415	416	417	418	419	420	421	422	423	424	425
5	514	515	516	517	518	519	520	521	522	523	524	525
6	614	615	616	617	618	619	620	621	622	623	624	625

Obs	value26	value27	value28	value29	value30	value31	value32	value33	value34	value35	value36	value37
1	126	127	128	129	130	131	132	133	134	135	136	137
2	226	227	228	229	230	231	232	233	234	235	236	237
3	326	327	328	329	330	331	332	333	334	335	336	337
4	426	427	428	429	430	431	432	433	434	435	436	437
5	526	527	528	529	530	531	532	533	534	535	536	537
6	626	627	628	629	630	631	632	633	634	635	636	637

Obs	value38	value39	value40	value41	value42	value43	value44	value45	value46	value47	value48	value49
1	138	139	140	141	142	143	144	145	146	147	148	149
2	238	239	240	241	242	243	244	245	246	247	248	249
3	338	339	340	341	342	343	344	345	346	347	348	349
4	438	439	440	441	442	443	444	445	446	447	448	449
5	538	539	540	541	542	543	544	545	546	547	548	549
6	638	639	640	641	642	643	644	645	646	647	648	649

Obs	value50	value51	value52	value53	value54	value55	value56	value57	value58	value59	value60
1	150	151	152	153	154	155	156	157	158	159	160
2	250	251	252	253	254	255	256	257	258	259	260
3	350	351	352	353	354	355	356	357	358	359	360
4	450	451	452	453	454	455	456	457	458	459	460
5	550	551	552	553	554	555	556	557	558	559	560
6	650	651	652	653	654	655	656	657	658	659	660

The SAS System 13:03 Wednesday, May 26, 1999 10

Obs	value1	value2	value3	value4	value5	value6	value7	value8	value9	value10	value11	value12	value13
55	5501	5502	5503	5504	5505	5506	5507	5508	5509	5510	5511	5512	5513
56	5601	5602	5603	5604	5605	5606	5607	5608	5609	5610	5611	5612	5613
57	5701	5702	5703	5704	5705	5706	5707	5708	5709	5710	5711	5712	5713
58	5801	5802	5803	5804	5805	5806	5807	5808	5809	5810	5811	5812	5813
59	5901	5902	5903	5904	5905	5906	5907	5908	5909	5910	5911	5912	5913
60	6001	6002	6003	6004	6005	6006	6007	6008	6009	6010	6011	6012	6013

Obs	value14	value15	value16	value17	value18	value19	value20	value21	value22	value23	value24	value25
55	5514	5515	5516	5517	5518	5519	5520	5521	5522	5523	5524	5525
56	5614	5615	5616	5617	5618	5619	5620	5621	5622	5623	5624	5625
57	5714	5715	5716	5717	5718	5719	5720	5721	5722	5723	5724	5725
58	5814	5815	5816	5817	5818	5819	5820	5821	5822	5823	5824	5825
59	5914	5915	5916	5917	5918	5919	5920	5921	5922	5923	5924	5925
60	6014	6015	6016	6017	6018	6019	6020	6021	6022	6023	6024	6025

Obs	value26	value27	value28	value29	value30	value31	value32	value33	value34	value35	value36	value37
55	5526	5527	5528	5529	5530	5531	5532	5533	5534	5535	5536	5537
56	5626	5627	5628	5629	5630	5631	5632	5633	5634	5635	5636	5637
57	5726	5727	5728	5729	5730	5731	5732	5733	5734	5735	5736	5737
58	5826	5827	5828	5829	5830	5831	5832	5833	5834	5835	5836	5837
59	5926	5927	5928	5929	5930	5931	5932	5933	5934	5935	5936	5937
60	6026	6027	6028	6029	6030	6031	6032	6033	6034	6035	6036	6037

Obs	value38	value39	value40	value41	value42	value43	value44	value45	value46	value47	value48	value49
55	5538	5539	5540	5541	5542	5543	5544	5545	5546	5547	5548	5549
56	5638	5639	5640	5641	5642	5643	5644	5645	5646	5647	5648	5649
57	5738	5739	5740	5741	5742	5743	5744	5745	5746	5747	5748	5749
58	5838	5839	5840	5841	5842	5843	5844	5845	5846	5847	5848	5849
59	5938	5939	5940	5941	5942	5943	5944	5945	5946	5947	5948	5949
60	6038	6039	6040	6041	6042	6043	6044	6045	6046	6047	6048	6049

Obs	value50	value51	value52	value53	value54	value55	value56	value57	value58	value59	value60
55	5550	5551	5552	5553	5554	5555	5556	5557	5558	5559	5560
56	5650	5651	5652	5653	5654	5655	5656	5657	5658	5659	5660
57	5750	5751	5752	5753	5754	5755	5756	5757	5758	5759	5760
58	5850	5851	5852	5853	5854	5855	5856	5857	5858	5859	5860
59	5950	5951	5952	5953	5954	5955	5956	5957	5958	5959	5960
60	6050	6051	6052	6053	6054	6055	6056	6057	6058	6059	6060

Note that the table is split into panels because it is wide, and that titles repeat at the top of the next page.

The majority of this paper will focus on examples using the ODS PRINTER destination.

RTF

Available experimentally in v8 is the RTF destination. This is the format used by Microsoft Word to store documents. Since it is experimental, the RTF support cannot be guaranteed always to work as expected, but we are trying to make it pretty solid. The biggest advantage of this format is that it is relatively easy to add text and/or to modify the appearance of the output visually. A small example of RTF output is shown here.

The STANDARD Procedure

Name	Mean	Standard Deviation	N
Age	13.315789	1.492672	19
Height	62.336842	5.127075	19
Weight	100.026316	22.773933	19

XML and LaTeX

The XML and LaTeX destinations are also available experimentally in v8, but are expected to be somewhat more experimental than RTF. XML is an emerging standard for object exchange between applications. An increasing number of applications understand this format, so it will probably be useful for producing hard copy in the future, but as of this writing, web browsers and data bases are the primary applications that support it.

LaTeX is an open-source language-based formatter for producing “desktop typesetting.” It is most popular in academia, particular among mathematicians. It is in widespread use in the Applications department at SAS Institute by the folks working on the mathematically-intensive parts of the SAS System (SAS/STAT, SAS/OR, SAS/QC, and so forth). LaTeX produces high-quality hardcopy output for a number of printers, but is probably the solution you are looking for only if you are already familiar with it.

ODS AND STYLES

For those who are not familiar with the basics of the Output Delivery System I’m going to give a brief synopsis of ODS, the table templates, and styles.

The Output Delivery System

In previous versions of the SAS System (prior to v7), each procedure formatted its own output by directly specifying how to print the data, in much the same way that you might format the data yourself in a data step.

With the introduction of the Output Delivery System in SAS v7, we've changed this completely. Instead of formatting the text itself, the procedure produces a *data object* that describes the data in its full-precision internal representation—more like a SAS data set than a listing file—and the name of the *template* to use to format the data.

All of the formatting information is contained in the *table template*. Although a default table template is supplied by the procedure writer, the template is specified with SAS code and can be modified by the local site or individual user.

In theory, the table template could supply detailed information about how to format each aspect of the individual table columns, but in practice it usually leaves most of the presentation details to the *style template*. Thus, in GLM, the template specifies that the "Class" column is in the RowHeader style rather than specifying that it's a bold font.

Templates describe the output of most, but not all, procedures. Some of the procedures, such PLOT and GRAPH, produce output which is not tabular at all. For these procedures, ODS simply reproduces the listing output in a fixed-size font. There are also four procedures which produce tables that are quirky in some way that makes it impossible to apply the templates to them. These are procs PRINT, FREQ (N-way tables only), TABULATE, and REPORT.

Of these, however, TABULATE and REPORT more than compensate by having syntax directly in the procedure syntax that allows you to specify formatting information for both the table and its individual components.

The remainder of this paper will use the Table Templates, special features of procs REPORT and TABULATE, and the Style Templates to demonstrate what can be done to produce stylish output intended for printers.

TABLE TEMPLATES

With a very few exceptions, the tables produced by the SAS System are described by templates, and the templates are the means to modify the appearance of an individual table. I will use as the example here the

table from PROC STANDARD, simply because it's one of the simplest tables in the system.

The Original Table Template

The first step is to determine the original table template so that we can then proceed to modify it as we want. The easiest way to do this is probably by using the GUI, but I'm going to show how to do it with SAS code anyway. The first step is to use "ODS TRACE " to find out what the table's name is:

```
7? ods trace on;
8? proc standard print mean=0; run;
```

Output Added:

```
-----
Name:          Standard
Label:         Standardized data set variables.
Template:     Base.Standard
Path:         Standard.Standard
-----
```

Then we get the source code by using PROC TEMPLATE:

```
11? proc template;
12? source base.standard;
define table Base.Standard;
    notes "Table definition for PROC
Standard.";
    column name mean std n label;
    define name;
        header = "Name";
        varname = 'Name';
        style = RowHeader;
    end;
    define mean;
        header = "Mean";
        format = D12.;
        varname = 'Mean';
    end;
    define std;
        header = "/Standard/Devi ation";
        format = D12.;
        varname = 'stdDev';
    end;
    define n;
        header = "N";
        format = best.;
    end;
    define label;
        header = "Label";
        varname = 'Label';
    end;
    requi red_space = 3;
```

```

byline;
wrap;
end;

```

This produces a table like this:

The STANDARD Procedure

Name	Mean	Standard Deviation	N
Age	13.315789	1.492672	19
Height	62.336842	5.127075	19
Weight	100.026316	22.773933	19

Modifying the Template

Let us suppose that, for some reason, we wanted to make the names come out all in italic, and the means in "reverse video." Then we would modify the template as follows (only the changed parts are shown below but the entire template must be supplied):

```

proc template;
  define table Base.Standard;
    notes "Table definition for PROC
Standard.";
    column name mean std n label;
    define name;
      header = "Name";
      varname = 'Name';
      style =
        RowHeader{font_style=italic};
    end;
    define mean;
      header = "Mean";
      format = D12.;
      varname = 'Mean';
      style = {background=black
        foreground=white};
    end;
  /* remainder same as before */

```

Producing a table like this:

The STANDARD Procedure

Name	Mean	Standard Deviation	N
<i>Age</i>	13.315789	1.492672	19
<i>Height</i>	62.336842	5.127075	19
<i>Weight</i>	100.026316	22.773933	19

PROCS REPORT AND TABULATE

There are two procedures in SAS with special syntax especially for altering the style. This example uses PROC REPORT to produce a font table. In addition to showing how this special syntax works, it also shows the fonts and styles available by default when using the SAS-supplied PostScript driver, which will come in handy when you are trying to make stylistic changes to your own reports. Here is the code:

```

title 'ODS Printer Font Table';

data;
  length font_face $100 plain $20
    italic $20 bold $20
    bold_italic $20;
  plain = "plain ABCdef123";
  italic = "italic ABCdef123";
  bold = "bold ABCdef123";
  bold_italic = "bold-italic ABCdef123";
  font_face = "Times"; output;
  font_face = "Helvetica"; output;
  font_face = "Courier"; output;
  font_face = "Symbol"; output;
  font_face = "Bookman"; output;
  font_face = "New Century Schoolbook";
    output;
  font_face = "Palatino"; output;
  font_face = "Chancery"; output;
  font_face = "Dingbats"; output;
run;

ods printer;

proc report;
  column font_face plain italic bold
    bold_italic;

  define plain /
    style={font_style=roman
      font_weight=medium};
  define italic /
    style={font_style=italic

```

```

        font_weight=medium);
define bold /
    style={font_style=roman
        font_weight=bold};
define bold_italic /
    style={font_style=italic
        font_weight=bold};

compute font_face;
    call define(_ROW_, 'STYLE',
        "style={font_face=' " ||
            trim(font_face) || "'}");
    endcomp;
run;

ods printer close;

```

This code first sets up a data set that looks like this:

Obs	font_face	plain	italic	bold	bold_italic
1	Times	plain ABCdef123	italic ABCdef123	bold ABCdef123	bold-italic ABCdef123
2	Helvetica	plain ABCdef123	italic ABCdef123	bold ABCdef123	bold-italic ABCdef123
3	Courier	plain ABCdef123	italic ABCdef123	bold ABCdef123	bold-italic ABCdef123
4	Symbol	plain ABCdef123	italic ABCdef123	bold ABCdef123	bold-italic ABCdef123
5	Bookman	plain ABCdef123	italic ABCdef123	bold ABCdef123	bold-italic ABCdef123
6	New Century Schoolbook	plain ABCdef123	italic ABCdef123	bold ABCdef123	bold-italic ABCdef123
7	Palatino	plain ABCdef123	italic ABCdef123	bold ABCdef123	bold-italic ABCdef123
8	Chancery	plain ABCdef123	italic ABCdef123	bold ABCdef123	bold-italic ABCdef123
9	Dingbats	plain ABCdef123	italic ABCdef123	bold ABCdef123	bold-italic ABCdef123

And then defines columns in PROC REPORT, setting the attributes of (for example) the “italic” column to be italic. For the “font face” column, it defines the style of the entire row using a call define statement so that it can use the values from the data set itself to define the set of fonts to display. The final table produced is the one shown on the title page.

OVERALL DOCUMENT STYLE

In addition to the *data object* supplied by the procedure and the *table template* we have already examined, there is a third element that determines the appearance of the table: the overall document style, controlled by a *style template*. The style template comes into play because rather than specifying each aspect of the presentation directly, the table templates instead specify the name of a style defined in the Style Template. For example, the GLM.ClassLevels table template contains this line:

```

define Class; id header="Class"
    style=RowHeader; end;

```

This means that the appearance of the Class column will be determined by looking in the Style Template for the style attributes. If no style is supplied in the table template, then the default “Data” style is used.

SAS-supplied styles

The easiest way to change the appearance of your document is to use one of the styles that SAS supplies “out of the box.” To see the list of styles available, do this:

```
Proc template; list styles; run;
```

As of this writing, the styles available are: Default (actually the HTML default style), Printer (the default Printer style), Beige, Brown, D3d, Minimal, Printer, Statdoc, fancyPrinter, sansPrinter, sasdDocPrinter, and serifPrinter. More may be added later, and your site may supply local styles as well.

For example, this is some output produced with a simple “ods printer” statement:

The STANDARD Procedure

Name	Mean	Standard Deviation	N
Age	13.315789	1.492672	19
Height	62.336842	5.127075	19
Weight	100.026316	22.773933	19

Here is the same output with the specification “ods printer style=fancyPrinter”

The STANDARD Procedure

Name	Mean	Standard Deviation	N
Age	13.315789	1.492672	19
Height	62.336842	5.127075	19
Weight	100.026316	22.773933	19

And here again with “ods printer style=brick”, which is one of the “HTML-oriented” styles:

The STANDARD Procedure

Name	Mean	Standard Deviation	N
Age	13.315789	1.492672	19
Height	62.336842	5.127075	19
Weight	100.026316	22.773933	19

Color Issues

As you may have noticed, the last example didn't really look all that great. The problem is that it is using a style designed for HTML browsers, where color is a given, but it is being reproduced here in black-and-white (well, grayscale, actually).

This is perhaps the biggest difference between HTML styles and Printer styles: stylistic distinctions must be made without the use of color. (Actually, this is good advice anyway, for the benefit of any color-impaired viewers your work might have.)

Moreover, while the printers theoretically support "grayscale," they all in fact dither the grayscale out of little dots. This works just fine for filled backgrounds, and in the default printer style, but it makes narrow rules look more dotted than gray. Similarly, all but the largest fonts look terrible when set in any color other than black on a typical 300 dpi office printer. So all the examples in the remainder of this paper will avoid using colors with differing hues entirely, and will use even grayscale only for backgrounds.

If you are lucky enough to have a 600 dpi or higher, or a color printer available, you will have more freedom. Indeed, if you have color available, you can use all of the HTML-oriented styles can get perfectly pleasing results.

Creating your own style

It is possible to create your own style, and customize to your heart's content. The "base" style for the entire system is called "default," and you can print it out by using

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

However, it is enormous and only rarely will you need to deal with all the elements in that style. Most styles simply want to alter the colors and/or fonts used by the style, and perhaps change one or two individual elements. Conveniently, the default ODS PRINTER style, "styles.printer", does exactly this,

and is a template of much more manageable size, so for most purposes you will probably do this:

```
Proc template; source styles.printer; run;
```

To see the default printer style, and use that as a starting point for your own styles. If you wish to change the appearance of an individual element of the document, such as the page number, the date, system titles, or all data cells, though, you will need to use the "styles.default" as your starting point for altering those elements. For example, here is a document style in which we start with elements copied and modified from "styles.printer", and then add the TitlesAndFooters element from styles.default, so that we get this code:

```
proc template;
define style Styles.mystyle;
  parent = styles.Printer;

  /*--- from styles.Printer ---*/
  replace fonts /
    'TitleFont2' =
      ("Helvetica", 12pt, Bold)
    'TitleFont' =
      ("Helvetica", 13pt, Bold)
    'StrongFont' =
      ("Times", 10pt, Bold)
    'EmphasisFont' =
      ("Times", 10pt, Italic)
    'FixedEmphasisFont' =
      ("Courier", 9pt, Italic)
    'FixedStrongFont' =
      ("Courier", 9pt, Bold)
    'FixedHeadingFont' =
      ("Courier", 9pt, Bold)
    'BatchFixedFont' =
      ("Courier", 7pt)
    'FixedFont' =
      ("Courier", 9pt)
    'headingEmphasisFont' =
      ("Chancery", 13pt, Bold Italic)
    'headingFont' =
      ("Chancery", 11pt, Bold)
    'docFont' =
      ("Times", 10pt, italic);
  style Table from Output /
    rules = ROWS
    frame = BOX
    cellpadding = 4pt
    cellspacing = 0.50pt
    borderwidth = 3pt
    BorderColor = gray99
    Background = _undef_;
  replace color_list
    'bgH' = white
    'fg' = black
```

```

'bg' = white;

/*--- from styles.default ---*/
replace TitlesAndFooters
  from Container /
  background = gray33
  foreground = white;

end;
run;

ods printer style=mystyle file="p7.ps";
proc standard print; run;
ods printer close;

```

A couple subtleties should be noted here. First, in the Table definition, the “background=_undef_” is necessary; the default table background is black, which effectively forces rules on everywhere, since the table background color “shows through” in the spacing between the table cells. Second, it is necessary to completely re-define any element if you want to replace part of it; thus, we had to list all of the fonts even though we were only changing a few of them.

This is the result:

The SAS System

The STANDARD Procedure

Name	Mean	Standard Deviation	N
Age	13.315789	1.492672	19
Height	62.336842	5.127075	19
Weight	100.026316	22.773933	19

Admittedly, this is not likely to win any design awards, but it does show the significant changes that can be made with styles.

MODIFYING HTML STYLE EXAMPLES

If you have seen other examples of “stylish” SAS output which uses HTML, there are a few points that you need to be aware of if you want to get nice-looking printer output:

- As previously mentioned, colors are not available on most office printers. Some colors happen to

work well when translated to grayscale; others do not.

- On some printers, even the grayscale which ODS PRINTER code uses by default does not look very good. This is particularly true if you are going to turn around and FAX or photocopy the results. If this is the case for your printer, use the NOCOLOR option for ODS PRINTER and it will not draw any of the background colors.
- ODS PRINTER does not support images in v8. This feature *will* be supported for v9. If you must have images in printed output, you should use HTML and print from the browser. Of course, that won’t work well for large tables, but I doubt that there are very many large tables with images that need to be printed. (I hope not, anyway.)

I am going to take a couple of the examples from the “Your Base SAS Dollars At Work” paper and re-work them for the printer. All the examples that modify table templates to alter the arrangement of data, or that show off new features of the procedures, of course, work just fine as they stand.

PROC REPORT “traffic-lighting”

For example, here is a PROC REPORT example done in an HTML style. Only part of the table is shown in order to make the results visible.

Retail Wholesale Amounts per Region						
		Saletype				
		Retail		Wholesale		
Citysize	Quantity	Amount	Quantity	Amount	Comments	
E			2,272	\$45,440	Retail amount is missing.	
M	1,066	\$26,600	1,066	\$21,320		
S	472	\$11,800	472	\$9,440		
	1,538	\$38,400	3,810	\$76,200	Wholesale amount greater than re	
MC Region – Sales Need Improvement						

The style elements can be changed to rely less on color changes and more on other distinctions; for example, I took the original specification

```

style(COLUMN)={ foreground=yellow
  font_weight=bold };

```

and change it into this:

```

style(COLUMN) =
  {font=("New century school book",
    10pt, bold italic)};

```

Similarly, I took the specification for the missing data and changed the pink background into this:

```

call define('COMMENTS', "STYLE",

```

```
"style(CALLDEF)={just=c
fillrulewidth=0.5pt}");
```

(The fillrulewidth specification causes the line to be filled in with a rule.) The result looks like this:

Wholesale Amounts per Region						
		Saletype				
		Retail		Wholesale		
Citysize	Quantity	Amount	Quantity	Amount	Comments	
L	.	.	2,272	\$45,440	Retail amount is mi	
M	1,066	\$26,600	1,066	\$21,320		
S	472	\$11,800	472	\$9,440		
L	1,538	\$38,400	3,810	\$76,200	Wholesale amount greater than re	
NC Region - Sales Need Improvement						

Of course, in practice you may find that the HTML styles only need a little tweaking to work just fine for your printing purposes. For example, in converting the HTML to grayscale for the screen, everything but the yellow actually comes out just fine. It is important, however, to try looking at the results on a printer before assuming that the styles that work well for HTML will work well for PostScript as well.

A PROC TABULATE example

Similarly, the PROC TABULATE example from the "Base Dollars" paper has this specification:

```
proc tabulate s={foreground=green};
class region citysize saletype
  / s={foreground=blue};
classlev region citysize saletype
  / s={foreground=yellow};
var quantity amount
  / s={foreground=black};
keyword all sum / s={foreground=white};
format saletype $salefmt.;
label region="Region" citysize="Citysize"
  saletype="Saletype";
label quantity="Quantity"
  amount="Amount";
keylabel all="Total";

table all={label = "All Products"
  s={foreground=orange}},
  (region all)*(citysize
  all*{s={foreground=#002288}}),
  (saletype all)*(quantity*f=COMMA6.
  amount*f=dollar10.)
  / s={background=red}
  missertext={label="Missing"
  s={foreground=brown}}
  box={label="Region by Citysize by
  Saletype" s={foreground=purple}};
```

which produces a table like this:

All Products							
Region by Citysize by Saletype		Saletype				Total	
		Retail		Wholesale			
		Quantity	Amount	Quantity	Amount	Quantity	Amount
		Sum	Sum	Sum	Sum	Sum	Sum
Region	Citysize						
L		Missing	Missing	2,272	\$45,440	2,272	\$45,440
M		1,066	\$26,600	1,066	\$21,320	2,132	\$47,920
S		472	\$11,800	472	\$9,440	944	\$21,240
	Total	1,538	\$38,400	3,810	\$76,200	5,348	\$114,600

By changing the colors into font specifications, we can transform it into a form where the same distinctions can be seen even in the extreme case where no grayscales can be used at all. Here is the new tabulate code:

```
proc tabulate s={font_face=bookman};
class region citysize saletype
  / s={font_style=italic};
classlev region citysize saletype
  / s={font_style=italic
  font_weight=bold};
var quantity amount
  / s={font_weight=bold};
keyword all sum
  / s={font=("helvetica",
  10pt, bold italic)};
format saletype $salefmt.;
label region="Region" citysize="Citysize"
  saletype="Saletype";
label quantity="Quantity"
  amount="Amount";
keylabel all="Total";

table all={label = "All Products"
  s={font_face=chancery}},
  (region all)*(citysize
  all*{s={font_face=helvetica
  font_weight=bold}}),
  (saletype all)*(quantity*f=COMMA6.
  amount*f=dollar10.)
  / s={fillrulewidth=1pt}
  missertext={label="Missing"
  s={font_face=courier
  font_style=italic}}
  box={label="Region by Citysize by
  Saletype" s={font_face=helvetica
  font_style=italic}};

run;
```

Which produces a table like this (in NOCOLOR mode):

All Products

Region by Citysize by Saletype		Saletype				Total	
		Retail		Wholesale			
		Quantity	Amount	Quantity	Amount	Quantity	Amount
		Sum	Sum	Sum	Sum	Sum	Sum
Region	Citysize						
NC	L	Missing	Missing	2,272	\$45,440	2,272	\$45,440
	M	1,066	\$26,600	1,066	\$21,320	2,132	\$47,920
	S	472	\$11,800	472	\$9,440	944	\$21,240
	Total	1,538	\$38,400	3,810	\$76,200	5,348	\$114,600

CONCLUSION

There are a number of ways to control and alter the appearance of the SAS output for all of the ODS destinations other than LISTING and OUTPUT. Most of the stylistic elements will work for any destination, but the best-looking results will be obtained if the style used is sensitive to the intended destination. This paper has only scratched the surface of how to best get “stylish” output for printers from the SAS system, and I hope that this topic will be more extensively explored as more people get familiar with the new ODS capabilities.