Paper 172-26

Visual Styles for V9 SAS Output

Frank Wimmer, SAS® Institute
Ken Ellis, SAS® Institute

ABSTRACT

New development underway for SAS/GRAPH® version 9 targets new heights of visual style for SAS Output. The new Style Editor will allow you to easily create stylish presentation graphics. In addition to focusing upon ease of use, new graph rendering technology is being used to add sophisticated visual effects to both Java and ActiveX bar and pie charts. This paper discusses the development of these styles and the graphing enhancements being implemented in order to accomplish the new looks.

INTRODUCTION

A series of visual styles have been available to SAS users for some time through PROC TEMPLATE. New heights of visual styles are currently being implemented for version 9, which go far beyond what SAS users have seen in SAS Output to date.

CONTROLLING VISUAL OUTPUT IN SAS

The Output Delivery System (ODS) within SAS provides a very robust set of features that can be used to control the presentation aspects of output. A rich set of attributes are defined that allow for precise control over visual elements such as tables, graphs, titles and footnotes. These attributes are grouped together to form ODS Style definitions that allow users to customize entire SAS jobs or single documents.

SAS provides a set of default styles that users can apply to their output by using the *Style=* option on the ODS statement. The currently defined HTML styles are:

Default	D3D
Beige	Minimal
Brick	Printer
Brown	Statdoc

As you can see, this rather limited set lacks the breadth of stylistic choices that many of today's web and publication-centric mediums require.

PROC TEMPLATE TO THE RESCUE?

Early on, it was realized that most users would not be completely satisfied with the limited set of styles being provided with ODS. SAS would have to provide some mechanism by which users would be able to generate customized output, tailored to their particular needs.

For this reason, PROC TEMPLATE was created. PROC TEMPLATE allows users to view styles, modify styles, and even create their own new ones. With PROC TEMPLATE, you can control, in minute detail, every aspect of your output.

But wait, we were talking about styles. So, how does a template relate to a style? Actually, it is quite simple. ODS stores style definitions in a specific kind of item store (a special kind of SAS

file) called a template store. The default style definitions that SAS provides are stored in the template store SASHELP.TMPLMST.

A good way to understand what a template definition looks like is to view the contents of one. For instance, submitting the following statements…

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

displays the definition of the Default HTML style to the SAS log as follows: (partial listing)

```
1    proc template;
2        source styles.default;
define style Styles.Default;
   style fonts
     "Fonts used in the default style" /
     '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",2,Italic)
     'FixedStrongFont' = ("Courier",2,Bold)
     'FixedHeadingFont' = ("Courier",2)
     'BatchFixedFont' = ("SAS Monospace,
Courier",2)
     'FixedFont' = ("Courier",2)
     'headingEmphasisFont' = ("Arial,
Helvetica, Helv" ,4

,Bold Italic)
     'headingFont' = ("Arial, Helvetica,
Helv",4,Bold)
     'docFont' = ("Arial, Helvetica, Helv",3);
   style color_list
     "Colors used in the default style" /
     'fgB2' = cx0066AA
     'fgB1' = cx004488
     'fgA4' = cxAAFFAA
     'bgA4' = cx880000
     .
     .
     .
```

(followed by a few *hundred* more lines of output)

As you can see, the definition of a style can be quite complex and intimidating for all but the hard-core code warrior!

PROGRAMMING WITH PROC TEMPLATE CAN BE FUN!

PROC TEMPLATE's power (and resulting complexity) is a direct result of the granularity of control that it provides. Empirical evidence suggests that the amount of coding necessary to

achieve the required visual result has a direct impact on how willing users are to use the procedure. In other words, users commonly make simple modifications that don't require an in-depth understanding of minute details, while it is rare that they use the procedure to make wholesale stylistic changes.

An example can be worth more than words. In the following sample code, PROC TEMPLATE is used to define a table element, and a style element that will be used to modify the appearance of the table's cells.

```
/* setup ODS */
ods path sashelp.tmplmst(read)
        sasuser.templat(update);

/* Create the test dataset */
title;
options nodate pageno=1 linesize=72 pagesize=60;
data test;
    input country $ 1-13 grain $ 15-19;
    datalines;
Canada        Wheat
China         Rice
India         Rice
Mexico        Corn
United States Wheat;

/* Define a simple 2-column table */
proc template;
    define table mytable;
        column x y;
        define x;
            style=simplecell;
            dataname=country;
            header='Country';
        end;
        define y;
            style=simplecell;
            dataname=grain;
            header='Grain';
        end;
    end;
run;
```

Careful examination of the table definition reveals that it will expect to get data for its columns from two variables, 'country' and 'grain'. At the table definition level, no assumptions are made as to what dataset the variables will be supplied from. This detail is abstracted out of the table definition. Also, note the *style=* reference embedded in the table definition.

```
        style=simplecell;
```

We will need to supply a definition for simple cell, as one does not exist by default. The following code creates this definition in a user-defined style, 'example.style1'.

```
proc template;
    define style example.style1;
        style simplecell /
            font_face=arial
            background=CX2277AA
            foreground=white;
    end;
run;
```

Now by referencing our new style in the ODS statement, the output generated from my table can use our simple cell element…

```
ods html body='display2-body.htm'
        style=example.style1;
data _null_;
    set test;
    file print ods=(template='mytable');
    put _ods_;
run;
ods html close;
```

and generate the following output:

Figure 1

Is the amount of coding required to generate this output overly complex? That is a question that can only be answered by the individual charged with creating the final product. Bear in mind that this is a *very* simple example of how a user would typically use PROC TEMPLATE to generate output. In general terms, the amount of custom coding increases significantly with the desire to generate custom output.

MEET THE NEW GRAPH STYLES

SAS graph output has generally been considered the gold standard of accuracy and reliability for visualizing data. Recently, the idea was proposed that, in addition to being the industry leader in that respect, we should also be the industry leader in visual presentation. Rather than just providing visualization tools that provide unsurpassed accuracy, we could also provide graphs with visual flair that not only convey important information, but convey that information with great style. In particular, such graphs would be a boon for business presentations, when you really need to capture the attention of your audience.

Enter the 99 SAS Graph Styles. To test out this idea, a series of prototypes were created to provide ideas for the kinds of things that might be done with graphs.

PROTOTYPES FOR GRAPH STYLES

Using a bitmap paint package, a series of about 20 prototype images were created which suggested different graphic designs that might be emulated with our graph output. Some of these prototypes mimicked the appearance of various traditional artistic media, such as watercolor, pencil, and ink. Others had themes based upon certain industries, such as education, medicine, and manufacturing.

Next, these themes were narrowed down to 16 prototypes, which were then submitted to a committee for consideration. The goal was to narrow the designs to 10 styles that would be offered as new, user selectable graph styles. Following are two examples of the selected graph styles.
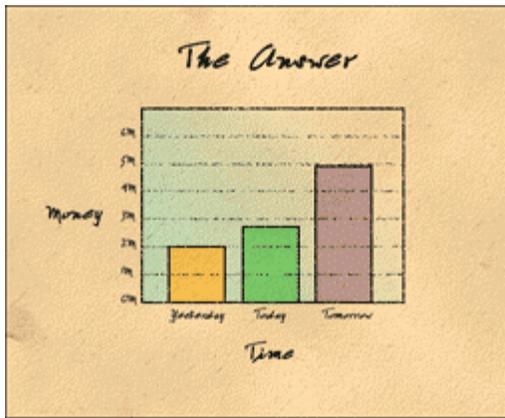
Figure 2

The "Watercolor" style above shows the use of transparency on the bars. The chart wall also shows the use of using a gradient that goes from one color (on the left) to transparency (on the right) which completely allows the background to show through. Gridlines exhibit a stippled appearance, allowing the lines to appear hand drawn. Anti-aliasing is also used on the fonts so that they appear more smooth.
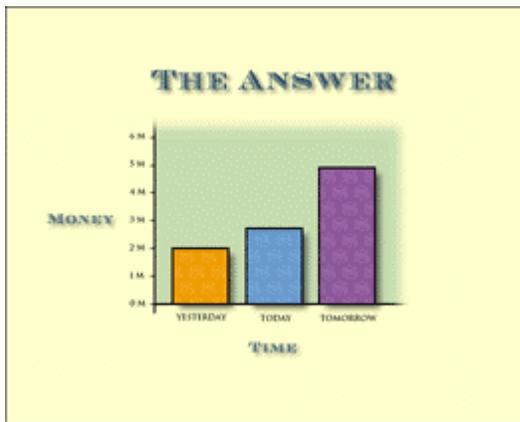

Figure 3

In addition to effects mentioned above, the "Banker" style shown here uses soft cast shadows behind the labels and bars. To evoke the banking theme, this image displays dollar-sign textures on the bars.

After the 10 final graph designs were chosen, the prototypes were compared to see what visual effects were shared. These shared effects included a variety of qualities such as texture maps, gradients, transparency, soft cast shadows, and anti-aliasing (to smooth jagged edges). These effects were then used as a list of new requirements for version 9 graph output.

EVOLUTION FROM GRAPH STYLES TO ODS OUTPUT

Once the prototypes for the graph styles were completed, the suggestion was made that these styles should be incorporated as a part of the ODS styles. In that way, these new visual styles would no longer be limited to just graphs. The same themes would be carried over into ODS output as well, so that tables and text output would have the same coordinated appearance.
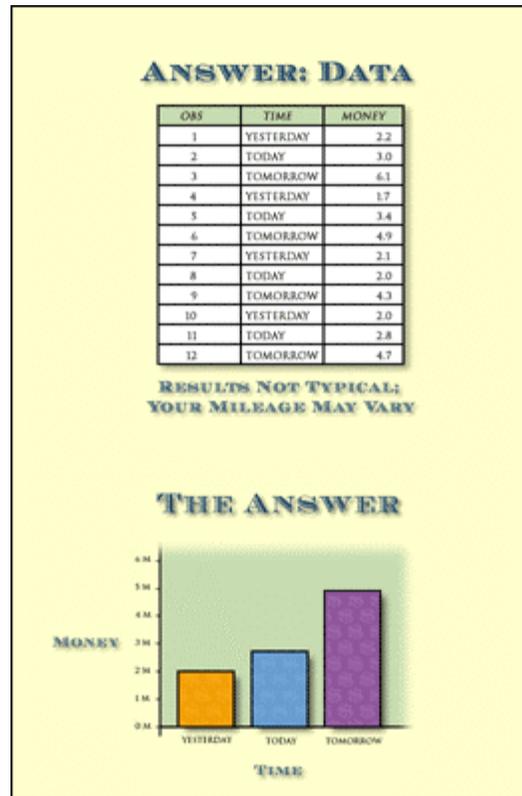

Figure 4

NEW VISUAL STYLES

The new visual styles, with new, enhanced visual effects in charts will be available for ODS output in version 9. This output provides users with 10 ready-made styles that offer new effects such as cast shadows, improved font support, transparencies and more. By selecting any of these 10 styles, users will see greatly enhanced output. In version 9, you will simply select which of the styles you want to apply and have your ODS output take on the selected appearance.

The addition of 10 new enhanced styles in ODS will go a long way toward meeting the needs of most SAS users. However, there may still be times when you will want to customize a style, or create an entirely new style, to fit a particular presentation objective. A project that is currently underway in the Data Visualization Research & Development division (DVR) could go a long way toward making it simple to do just that!

THE STYLE EDITOR

While PROC TEMPLATE is very powerful for modifying SAS Output, no doubt it can be a bit intimidating to some users. The Style Editor, currently under development, targets such users by providing a graphical user interface ODS output. Using the Style Editor, it will no longer be necessary for you to edit PROC TEMPLATE code in order to modify the visual style of ODS Output. However, the goal of the Style Editor is not simply a GUI to address ease of use.

The goal of the Style Editor GUI is to make it simple to generate great looking output. The Style Editor will emphasize the ability to adjust elements of the output based upon good graphic design principles. This is an important goal, since anyone should be able to easily generate great looking charts and tables, without

being required to have an art background or a graphic design degree.

The Style Editor will meet this challenge by allowing you to modify elements that are organized into meaningful groupings. For instance, from a graphic design standpoint, a table title and chart title on the same page should use the very same font. Similarly, key elements of your tables and charts should utilize the same colors.

The Style Editor will simply present you with choices for up to three fonts and apply those fonts to the appropriate text elements. Similarly, key colors that you select will be automatically applied to the logical elements of your charts and tables. This avoids excessive use of fonts and poor use of colors.

Similarly, applying great looking textures will be easier than ever before. In the past, in order to modify the color of a texture map for a chart, you had to edit the original texture, which required using a bitmap paint package to modify the colors of the image file. With the upcoming Style Editor, you can use new color-modulated texture maps, so that modifying the color of a texture simply means selecting any new color that you want for the texture.

For those who want complete control over their styles, there will always be PROC TEMPLATE. But for those who want a simple way to create great looking output, without a lot of fuss, watch for the upcoming Style Editor.

CONCLUSION

PROC TEMPLATE is currently available to give you control over the appearance of ODS Output. Version 9 of SAS will introduce new styles that provide visual enhancements not previously seen before.

Improvements in version 9 graphing components will provide new rendering capabilities such as anti-aliasing, soft cast shadows, transparency effects, and color-modulated texture mapping. These enhancements provide a level of visual flair never before seen in any data visualization tools. In addition to these technical embellishments, modifications to ODS will allow SAS output to show new visual styles that will be unparalleled in presentation graphics. Those new styles, along with the ease of creating customized appearances with the Style Editor, will allow SAS users to make a greater visual impact than ever before when presenting data.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Frank Wimmer
SAS Institute Inc
SAS Campus Drive
Cary, NC, 27513
Email: frank.wimmer@sas.com

Ken Ellis
SAS Institute Inc
SAS Campus Drive
Cary, NC, 27513
Email: ken.ellis@sas.com

TRADEMARK CITATION

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.