

Paper 112-31

PROC TEMPLATE: The Basics

Lauren Haworth, Genentech, Inc., South San Francisco, CA

ABSTRACT:

So you haven't gotten around to reading the entire PROC TEMPLATE chapter in Online Doc? The TEMPLATE procedure is incredibly powerful, but it's also incredibly complex, and this chapter of the documentation is not an easy read. If you don't have time to learn everything, but you want to be able to make basic changes to the appearance of your output, this is the workshop for you.

We'll start out with a few syntax basics, and then focus on a number of shortcuts that will make customizing the style of your output quick and easy. Topics include viewing the source code for an Output Delivery System style, a point-and-click approach to modifying styles, and plenty of sample code snippets to use in your own programs. In no time you'll be impressing your boss and coworkers with beautiful reports. The presentation will include easy-to-use techniques for enhancing HTML, RTF, and PDF output.

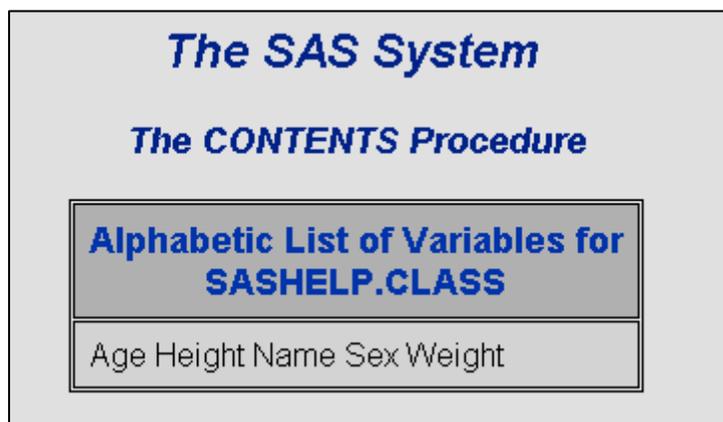
INTRODUCTION:

We will start by working through a series of examples using the HTML destination. The reason for this is that HTML is the easiest destination to deal with in this workshop environment. Once we've gone over the basics, we'll switch to RTF and PDF to do a few additional examples. However, all of the techniques shown here today will work with any destination.

First, we need to review a few syntax and terminology basics. When you create a report using the SAS Output Delivery System (ODS), SAS takes the raw procedure output and applies a **style template** to control how the results are displayed. This includes the fonts, colors, and borders. By default, SAS has a style template it will use for each output destination. If you request HTML output using the following code:

```
* sample program 1 ;
ods html file='sample1.html' ;
  proc contents data=sashelp.class short ;
  run ;
ods html close ;
```

the resulting file will look like this:



The code for this output is in sample program 1. Submit this code yourself and review the results. Depending on the setup of your system, the results may pop up in a browser within the SAS display manager, or you may need to go to the Results window, navigate the tree structure until you see the HTML file, and then double click on the file to open it up.

SAS creates this blue and gray style, with its Arial fonts, by using specifications contained in **style definitions** that make up a style template. Since the code above did not request a specific style, the default style for this destination was used. For the HTML destination, the default style is called Default. When you submit an ODS HTML statement without naming a style, it's equivalent to submitting the following code.

```
ods html file='sample1.html' style=default ;
```

To see what a style template looks like, we can use the TEMPLATE procedure to print out a style's source code. The syntax is as follows:

```
* sample program 2 ;
proc template;
  source styles.default;
run;
```

Go ahead and run this code yourself, and view the results (in the log window, not the output window). This code is in sample program 2. It produces an extremely long listing of source code. Appendix A contains a printout of selected style definitions. If you skim through the items in the source code, you'll see that a number of styles are defined, for everything from fonts to titles to tables. A template this large can be overwhelming to read, so let's look at another tool that will help you understand the style template.

USING THE POPUP STYLE TO EXPLORE THE ELEMENTS OF A STYLE TEMPLATE:

The following code is the secret to quickly making modifications to style template. It gives you an easy way to see which bits of code control which parts of your output.

```
* sample program 3 ;
ods markup file="sample2.html" tagset=tagsets.style_popup;

title 'Class List';
footnote 'See www.laurenhaworth.com for more ODS papers and examples';
proc means data=sashelp.class min median max maxdec=1;
run;

ods markup close;
```

This bit of code creates sample HTML output with some special features. When you hover over part of the output, it tells you the name of the style element that controls that attribute. Even better, when you click on a part of the output, it pops up the actual source code for that style element. For example, if you wanted to know what part of the style template source code controls the text in the table headers, by clicking on a cell in the header, you would see the following information:

The screenshot shows a SAS Results window with a popup window titled "Style Definition" and a table titled "Class List". The popup window displays the following style definition:

```
STYLE Header /
  FONT_FACE = "Arial, Helvetica, sans-serif"
  FONT_SIZE = 4
  FONT_WEIGHT = bold
  FONT_STYLE = roman
  FOREGROUND = cx0033aa
  BACKGROUND = cxb0b0b0
;
```

The "Class List" table is as follows:

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

Below the table, there is a footnote: [laurenhaworth.com](http://www.laurenhaworth.com) for more ODS papers and examples

From this output, we know that the text in this cell is controlled by the "STYLE Header" definition. If we want to change the header, we now know where to go look in the PROC TEMPLATE source.

This popup output is created using the MARKUP destination and a specialized tagset. The use of this destination and tagsets is a workshop in itself, and will not be covered in detail. For now, think of this as a useful tool, and don't worry about the details. The main thing to keep in mind is that when you want to modify your output, using these ODS MARKUP statements around your PROC will let you see the style elements in use for your specific output example.

Go ahead and run this code yourself. It is included in sample program 3. Then go to the results window, and expand the tree until you see the HTML icon for the PROC PRINT, and click on it to open it in the viewer. By clicking on one of the row headers in the HTML output, find the code you would need to change in order to alter the font in the table title.

Now rerun the PROC TEMPLATE code with the SOURCE statement, review the log file, and find the part of the PROC TEMPLATE code that matches the code in the popup window. The code is sample program 2.

In the PROC TEMPLATE code, the entry for "STYLE Header" looks different:

```
style Header from HeadersAndFooters
  "Controls the headers of a table.";
```

This style definition has no values for the FONT_FACE, FONT_SIZE, and other attributes listed in the definition that came from the popup window. That's because in the Default style, this element (Header) inherits its properties from another element (HeadersAndFooters). If you go look at that element, you see:

```

style HeadersAndFooters from Cell
  "Abstract. Controls table headers and footers." /
  font = fonts('HeadingFont')
  foreground = colors('headerfg')
  background = colors('headerbg');

```

This code looks closer, but it still doesn't match the popup. That's because it's inheriting its style attributes and values from other style elements (Cell), and some lookup lists: fonts() and colors(). To figure out the style, we could keep backtracking through the PROC TEMPLATE code until we found the actual settings.

But the beauty of the approach I'll be teaching today is that you don't need to do this. You can use the code in the popup window to create a new custom style, without ever having to know how the original style works. The popup window already has all of the values you need to change.

The rest of this workshop will focus on how to make these changes. We will first work with HTML, because that destination is easier to work with in a workshop environment, but then we'll look briefly at the same techniques for RTF and PDF.

HTML EXAMPLES:

CHANGING THE TITLE FONT

In the previous example, we used the popup style output to find out that the title at the top of the output was controlled by the SystemTitle style element. If we want to change the font, we need to edit the font selection in the template source code. Rather than edit the Default style, we'll create a new custom style.

The first step in doing this is to start our own style source code, using PROC TEMPLATE. However, this does not need to be as complex as the source code that we've been viewing for the Default style. To make it easier to modify styles, SAS created a PROC TEMPLATE statement called PARENT=. This allows the creation of a new style template based on the source code of an existing style template, so you don't have to retype any existing source code.

So, to start our new style, we need to set up the following PROC TEMPLATE code:

```

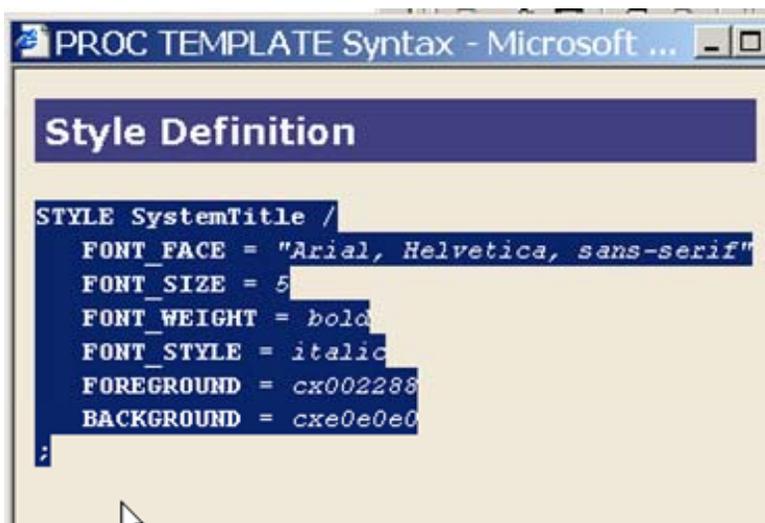
* sample program 4 ;
proc template;
  define style Styles.Custom;
    parent = Styles.Default;

    end;
run;

```

This code starts a new style called Custom, which will be created based on the Default style. If no other code was added to this PROC TEMPLATE, then we'd be creating a new style called Custom that looks exactly like the Default style. Now we are going to make some changes.

Open Sample Program 4. The PROC TEMPLATE code we just reviewed has already been typed in for you. To change the title font, we are going to edit the part of the source code that controls the title. Using the output from the previous example, click on the title again, and copy the code from the popup window. You can pull up the previous output by going to the Results window and clicking on the appropriate result to reopen the HTML file.



Take this code and paste it into the PROC TEMPLATE code in Sample Program 4, as follows:

```

proc template;
  define style Styles.Custom;
    parent = Styles.Default;
    STYLE SystemTitle /
      FONT_FACE = "Arial, Helvetica, sans-serif"
      FONT_SIZE = 5
      FONT_WEIGHT = bold
      FONT_STYLE = italic
      FOREGROUND = cx002288
      BACKGROUND = cxe0e0e0;
  end;
run;

```

Again, if we just ran the template code as is, our style would look just like the Default style. Now we can edit the style. To change the font used in the titles, we need to edit the font choices. In the style attribute FONT_FACE, there is a quoted string that lists three fonts. These are the 3 choices of font being requested, in order of preference. HTML allows you to specify multiple fonts, in case the person viewing your output in the browser does not have your first choice font.

To change the font, you simply edit the list of font names in the FONT_FACE style attribute line. Keep in mind that the person receiving your output will need to have the same fonts in order to view the web page, RTF file, or PDF file properly. You want to pick fonts that are commonly available. Appendix B lists some good font combinations to try.

Type over the text *"Arial, Helvetica, sans-serif"* with your choice of fonts. To use this new style, we need to add an option to the ODS HTML call. See the following code at the bottom of Sample Program 4. Notice that the option `"style=Custom"` has been added to the ODS HTML statement.

```

* sample program 4 (cont.) ;
ods html file="sample3.html" style=Custom;

title 'Class List';
footnote 'See www.laurenhaworth.com for more ODS papers and examples';
proc means data=sashelp.class min median max maxdec=1;
run;

ods html close;

```

Submit the program. The program will create your output using the new style. As an example, if the font is changed to Comic Sans MS (`FONT_FACE = "Comic Sans MS, Helvetica, sans-serif"`) then the new output will look like this.

<i>Class List</i>			
Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

See www.laurenhaworth.com for more ODS papers and examples

CHANGING THE FOOTNOTE FONT AND SIZE

In the Default style, ODS uses the same font size for both titles and footnotes. While big, bold titles can be attractive in your output, generally you want the footnote to be more discreet. This next example will show how to create a small italic footnote. At the same time, the italics will be removed from the title.

We already know the style source code for the title, so now we need to find the same code for the footnote. The approach is the same. Open the popup style example, click on the footnote, and copy the resulting code into the new PROC TEMPLATE.

The screenshot shows two windows from the SAS interface. On the left is a window titled "PROC TEMPLATE Syntax - Microsoft ...". It contains a "Style Definition" section with the following code:

```
STYLE SystemFooter /
  FONT_FACE = "Arial, Helvetica, sans-serif"
  FONT_SIZE = 5
  FONT_WEIGHT = bold
  FONT_STYLE = italic
  FOREGROUND = cx002288
  BACKGROUND = cxe0e0e0
;
```

On the right is a window titled "Class List" containing a table with the following data:

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

Below the table is a red banner with the text: aurenhaworth.com for more ODS papers and examples.

With this code added, the new PROC TEMPLATE should look like this:

```
proc template;
  define style Styles.Custom;
    parent = Styles.Default;
    STYLE SystemTitle /
      FONT_FACE = "Comic Sans MS, Helvetica, sans-serif"
      FONT_SIZE = 5
      FONT_WEIGHT = bold
      FONT_STYLE = italic
      FOREGROUND = cx002288
      BACKGROUND = cxe0e0e0;
    STYLE SystemFooter /
      FONT_FACE = "Arial, Helvetica, sans-serif"
      FONT_SIZE = 5
      FONT_WEIGHT = bold
      FONT_STYLE = italic
      FOREGROUND = cx002288
      BACKGROUND = cxe0e0e0;
  end;
run;
```

To change the footnote font size for the footnote, change the value of FONT_SIZE from 5 to something smaller. These values represent the HTML font sizes built into each browser. For more control, you can also specify a font size in points (e.g. 16pt). To change the font used for the footer, type over the text in the SystemFooter attribute FONT_FACE ("Arial, Helvetica, sans-serif") with some other font selection.

Finally, if you'd like to remove the italics from the title, change the line "FONT_STYLE = italic" from the SystemTitle section of the PROC TEMPLATE to specify *roman* instead of *italic*.

The following output was produced by changing the footnote font size to 2, the footnote font to Comic Sans MS, and removing the italics from the title.

Class List

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

See www.laurenhaworth.com for more ODS papers and examples

CHANGING THE COLUMN HEADER FONT AND WEIGHT

If you repeat the above process for the column headers you can make similar changes. Try this, changing the FONT_FACE, FONT_SIZE, FONT_WEIGHT, and FONT_STYLE attributes.

First, get the style definition for the column headers from the popup style:

The screenshot shows a Microsoft Word window titled "PROC TEMPLATE Syntax - Microsoft ...". On the left, a "Style Definition" pane shows the following code:

```

STYLE Header /
  FONT_FACE = "Arial, Helvetica, sans-serif"
  FONT_SIZE = 4
  FONT_WEIGHT = bold
  FONT_STYLE = roman
  FOREGROUND = cx0033aa
  BACKGROUND = cxb0b0b0
;

```

On the right, the "Class List" table is displayed. The "Minimum" column header is highlighted in red. The table data is as follows:

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

laurenhaworth.com for more ODS papers and examples

Next, paste this code into the PROC TEMPLATE, just as in the previous examples. Now we can change the font and weight of the column headers. To change the font, edit the FONT_FACE attribute. To change the weight by removing the bold face, change the FONT_WEIGHT attribute from *bold* to *medium*. If the font is changed to *Times New Roman*, and the weight to *medium*, the resulting output is as follows:

Class List

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

See www.laurenhaworth.com for more ODS papers and examples

CHANGING THE FONT IN THE TABLE CELLS

If you repeat the above process for the data in the table cells you can make similar changes. Try this, changing the FONT_FACE, FONT_SIZE, FONT_WEIGHT, and FONT_STYLE attributes.

First, get the style definition for the table cells from the popup style:

The screenshot shows a window titled "PROC TEMPLATE Syntax - Microsoft ...". On the left, a "Style Definition" popup is open, displaying the following code:

```
STYLE Data /
  FONT_FACE = "Arial, Helvetica, sans-serif"
  FONT_SIZE = 3
  FONT_WEIGHT = medium
  FONT_STYLE = roman
  FOREGROUND = cx000000
  BACKGROUND = cxd3d3d3
;
```

On the right, a "Class List" table is displayed with the following data:

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

Below the table, the text "See www.laurenhaworth.com for more ODS papers and examples" is displayed in a blue, italicized font.

Next, paste this code into the PROC TEMPLATE. Now we can change the font face and size used in the table data cells. *Times New Roman*, and the size to 2, the resulting output is as follows:

The screenshot shows the resulting ODS output. The "Class List" title is in a large blue font. Below it is a table with the following data:

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

Below the table, the text "See www.laurenhaworth.com for more ODS papers and examples" is displayed in a blue, italicized font.

CHANGING THE TABLE BORDERS AND RULES

Next, we move on to the table that is used to display the data. In ODS output, tables are important, as virtually all ODS output is displayed in the form of a table. In this example, we'll work with the lines around and inside the table. These are called borders and rules. Once again, we will use the popup style to find the code we need to modify. This time, it's a little trickier, as it's hard to click on the rules. The secret is to hover the mouse over the table until the borders and rules turn red. Then click to display the popup.

The screenshot shows a Microsoft Word window titled "PROC TEMPLATE Syntax - Microsoft ...". The document is divided into two sections:

Style Definition

```

STYLE Table /
  FONT_FACE = "Arial, Helvetica, sans-serif"
  FONT_SIZE = 3
  FONT_WEIGHT = medium
  FONT_STYLE = roman
  FOREGROUND = cx002288
  BACKGROUND = cx10f0f0
  CELLSPACING = 1
  CELLPADDING = 7
  FRAME = BOX
  RULES = GROUPS
;

```

Class List

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

Below the table, there is a watermark: www.laurenhaworth.com for more ODS papers and examples.

Again, we'll copy this code into our PROC TEMPLATE. Before working with the rules and borders, we can simplify our code a bit. The Table style definition contains four font attributes, but because we are controlling the data, header, title, and footnote fonts separately. These have no effect. You can delete these 4 lines of code.

To make changes to the rules and borders, there are two key attributes. The first one is called FRAME. It controls the border that runs around the outside of the table. It is currently set to BOX, which draws a border around the entire table. To find out what other settings you can use, open OnlineDoc, then browse the Guide to the Output Delivery System. In this document, navigate to: TEMPLATE Procedure: Creating a Style Definition : DEFINE STYLE Statement. On this page, you will find details about each of the style attributes and their values. For FRAME, you find:

Value for <i>frame-type</i> ...	Frame type
ABOVE	a border at the top
BELOW	a border at the bottom
BOX	borders at the top, bottom, and both sides
HSIDES	borders at the top and bottom
LHS	a border at the left side
RHS	a border at the right side
VOID	no borders
VSIDES	borders at the left and right sides

The second attribute that controls rules and borders is called RULES. Its values can be looked up in OnlineDoc as well:

This value of <i>rule</i> ...	Creates rules in these locations
ALL	between all rows and columns
COLS	between all columns
GROUPS	between the table header and the table and between the table and the table footer, if there is one
NONE	no rules anywhere
ROWS	between all rows

Now you can pick the look you want, and edit the PROC TEMPLATE code accordingly. If you change the FRAME setting to HSIDES, and the RULES setting to NONE, you can create the following output:

Class List

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

See www.laurenhaworth.com for more ODS papers and examples

Notice the heavier lines at the top and bottom of the table. These are created by the FRAME attribute. However, there are still lines inside the table. What happened? These are not rules, but rather the space between the table cells allowing the background color to show through. These lines can be removed by setting the CELSPACING option to 0. With that change, this is the new output:

Class List

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

See www.laurenhaworth.com for more ODS papers and examples

The final change you can make to the table structure is to adjust the spacing between the table cell contents and the table cell borders. This is a great trick to squeezing an oversized table onto the page. To reduce the space used by each table cell, set the CELLPADDING option to something smaller. In this example, the value has been reduced from 7 to 3.

Class List

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

See www.laurenhaworth.com for more ODS papers and examples

It's a little hard to see in this example, but the table did get a bit smaller.

CHANGING THE COLORS

In the previous examples, you may have noticed that each of the style elements (SystemTitle, SystemFooter, Header, Data, and Table) each contained style attributes called FOREGROUND and BACKGROUND. These control the colors used in the style. FOREGROUND is the text color, and BACKGROUND is the background color behind the text.

To change these values for any part of the output, all you have to do is type in a new color. Currently, the colors are set using hex codes like "cxE0E0E0". Unless you can look at that value and know that it stands for grey, you will probably find it easier to

work with named colors, at least at first.

For example, if you wanted to change the title text from blue to red, you would change the value "FOREGROUND = cx002288" in the SystemTitle style element to "FOREGROUND = red". By running this code, you would get the following output:

The screenshot shows a report titled "Class List" in red text. Below the title is a table with blue text. The table has four columns: Variable, Minimum, Median, and Maximum. The rows are Age, Height, and Weight. Below the table is a footer in blue text: "See www.laurenhaworth.com for more ODS papers and examples".

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

See www.laurenhaworth.com for more ODS papers and examples

See Appendix C for a table showing all of the named colors that are available to you. Using this list, edit your custom style to change the color scheme. For example, I can create a style to match my corporate logo colors.

The screenshot shows a report titled "Class List" in blue text. Below the title is a table with blue text. The table has four columns: Variable, Minimum, Median, and Maximum. The rows are Age, Height, and Weight. Below the table is a footer in blue text: "See www.laurenhaworth.com for more ODS papers and examples".

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

See www.laurenhaworth.com for more ODS papers and examples

This code used Midnightblue or Black for each of the FOREGROUND settings, and White for each of the BACKGROUND settings. But the background is still gray in many places. This is because the style definition is incomplete. To fix the problem, it is necessary to go back to the popup style and figure out what style element is missing.

The screenshot shows a window titled "PROC TEMPLATE Syntax - Microsoft ...". On the left is a "Style Definition" pane with the following code:

```
STYLE Body /
  FONT_FACE = "Arial, Helvetica, sans-serif"
  FONT_SIZE = 3
  FONT_WEIGHT = medium
  FONT_STYLE = roman
  FOREGROUND = cx002288
  BACKGROUND = cxe0e0e0
;
```

On the right is the resulting ODS output. The title "Class List" is in blue. The table below it has blue text. The footer is in blue text: "See www.laurenhaworth.com for more ODS papers and examples". The background of the report is red.

Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

See www.laurenhaworth.com for more ODS papers and examples

It turns out the missing code is for a style element called Body. By adding this code to the PROC TEMPLATE, and setting the BACKGROUND to white, the problem is solved. The new output no longer has the gray:

Class List			
Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

See www.laurenhaworth.com for more ODS papers and examples

There's still one problem remaining: the faint gray boxes around the title and footnote. These are output tables created by ODS. Just as the main results are a table, ODS creates a table to hold the title, and a table to hold the footnote. To make the boxes go away, the FRAME and CELLSPACING needs to be turned off for the title and footnote style elements (SystemTitle and SystemFooter). Currently, there are no FRAME or CELLSPACING attributes in these style elements. If you add them, there is no effect. That's because the borders and spacing are controlled by a different style element. This is a good example of the limitations of using the popup style to create new styles. Attributes controlled by style elements that you can't click on are hard to identify and modify.

To find the right style element, you can look at the HTML code. The table that contains the title is preceded in the HTML file by the following table tag:

```
<table class="SysTitleAndFooterContainer" width="100%" cellpadding="1" cellspacing="1"
rules="none" frame="void" border="0" summary="Page Layout">
```

Notice the value of class. This is the name of the style element that is controlling these tables. And notice that while the FRAME is set to "void", there is a value of 1 for CELLSPACING. This 1 is the culprit. By resetting it to 0, the grey boxes will go away. The syntax to add is:

```
STYLE SysTitleAndFooterContainer /
CELLSPACING=0;
```

The complete code is in Appendix D, and the output is shown below:

Class List			
Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

See www.laurenhaworth.com for more ODS papers and examples

CHANGING RTF OUTPUT:

Each of the preceding examples was based on creating HTML output. You can do the same thing for RTF output, there are just a few differences to note.

1. Instead of using the Default style as the parent, use the RTF style.
2. RTF by default places the date and page number at the top right of every page. To remove them, add the system options NODATE and NONUMBER.
3. RTF by default has very narrow page margins (0.25"). To change them, add the style attributes TOPMARGIN, BOTTOMMARGIN, LEFTMARGIN, and RIGHTMARGIN to the definition of the Body style element. Set them to a larger value (e.g. 1in).
4. You do not need to list multiple font names, as Word processors only use the first font you specify, unlike web browsers that can substitute when a font is not available.
5. If you are rerunning an ODS RTF program, you need to close the Results window before resubmitting the job,

otherwise you will get an error message as you can't write to an RTF while it's already open.

- Titles and footnotes will appear in the page header and footer, rather than in the body of the document above and below the results table, unless you add the option BODYTITLE to your ODS RTF statement.

The complete code is in Appendix E, and the output is shown below:

Class List			
Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

See www.laurenhaworth.com for more ODS papers and examples

CHANGING PDF OUTPUT:

Just as with RTF, there are a few things that will be different when using this approach to create PDF output.

- Instead of using the Default style as the parent, use the Printer style.
- PDF by default places the date and page number at the top right of every page. To remove them, add the system options NODATE and NONUMBER.
- If you get a WARNING message "Font specification "<fontname>" failed", you may need to register the fonts on your system. Execute the following code to fix the problem:


```
proc fontreg;
  fontpath "%sysget(systemroot)\fonts";
run;
```
- If you are rerunning an ODS PDF program, you need to close the Results window before resubmitting the job, otherwise you will get an error message as you can't write to an PDF while it's already open.
- Footnotes will appear at the bottom of the page, rather than right below the results table.

The complete code is in Appendix F, and the output is shown below:

Class List			
Variable	Minimum	Median	Maximum
Age	11.0	13.0	16.0
Height	51.3	62.8	72.0
Weight	50.5	99.5	150.0

CONCLUSION

Using this point and click approach is a simple way to create a custom style template. Fonts, borders, and colors are easy to change using this approach. Eventually, a programmer who uses ODS a lot may want to learn PROC TEMPLATE in more detail by reading the manual or taking a course.

CONTACT INFORMATION

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

Name: Lauren Haworth

E-mail: admin@laurenhaworth.com

Web: <http://www.laurenhaworth.com/publications.htm>

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 trademarks of their respective companies.

Appendix A: Source code (selected elements) for Default style

NOTE: Path 'Styles.Default' is in: SASHELP.TMPLMST.

```
define style Styles.Default;
  style fonts
    "Fonts used in the default style" /
    'TitleFont2' = ("Arial, Helvetica, sans-serif",4,Bold Italic)
    'TitleFont' = ("Arial, Helvetica, sans-serif",5,Bold Italic)
    'StrongFont' = ("Arial, Helvetica, sans-serif",4,Bold)
    'EmphasisFont' = ("Arial, Helvetica, sans-serif",3,Italic)
    'FixedEmphasisFont' = ("Courier New, Courier, monospace",2,Italic)
    'FixedStrongFont' = ("Courier New, Courier, monospace",2,Bold)
    'FixedHeadingFont' = ("Courier New, Courier, monospace",2)
    'BatchFixedFont' = ("SAS Monospace, Courier New, Courier, monospace",2)
    'FixedFont' = ("Courier",2)
    'headingEmphasisFont' = ("Arial, Helvetica, sans-serif",4,Bold Italic)
    'headingFont' = ("Arial, Helvetica, sans-serif",4,Bold)
    'docFont' = ("Arial, Helvetica, sans-serif",3);
  style color_list
    "Colors used in the default style" /
    'fgB2' = cx0066AA
    'fgB1' = cx004488
    'fgA4' = cxAAFFAA
    'bgA4' = cx880000
    'bgA3' = cxD3D3D3
    'fgA2' = cx0033AA
    'bgA2' = cxB0B0B0
    'fgA1' = cx000000
    'bgA1' = cxF0F0F0
    'fgA' = cx002288
    'bgA' = cxE0E0E0;
  style colors
    "Abstract colors used in the default style" /
    'headerfgemph' = color_list('fgA2')
    'headerbgemph' = color_list('bgA2')
    'headerfgstrong' = color_list('fgA2')
    'headerbgstrong' = color_list('bgA2')
    'headerfg' = color_list('fgA2')
    'headerbg' = color_list('bgA2')
    'datafgemph' = color_list('fgA1')
    'databgemph' = color_list('bgA3')
    'datafgstrong' = color_list('fgA1')
    'databgstrong' = color_list('bgA3')
    'datafg' = color_list('fgA1')
    'databg' = color_list('bgA3')
    'batchfg' = color_list('fgA1')
    'batchbg' = color_list('bgA3')
    'tableborder' = color_list('fgA1')
    'tablebg' = color_list('bgA1')
    'notefg' = color_list('fgA')
    'notebg' = color_list('bgA')
    'bylinefg' = color_list('fgA2')
    'bylinebg' = color_list('bgA2')
    'captionfg' = color_list('fgA1')
    'captionbg' = color_list('bgA')
    'proctitlefg' = color_list('fgA')
    'proctitlebg' = color_list('bgA')
    'titlefg' = color_list('fgA')
    'titlebg' = color_list('bgA')
    'systitlefg' = color_list('fgA')
    'systitlebg' = color_list('bgA')
    'Conentryfg' = color_list('fgA2')
    'Confolderfg' = color_list('fgA')
```

```

'Contitlefg' = color_list('fgA')
'link2' = color_list('fgB2')
'link1' = color_list('fgB1')
'contentfg' = color_list('fgA2')
'contentbg' = color_list('bgA2')
'docfg' = color_list('fgA')
'docbg' = color_list('bgA');
style Container
  "Abstract. Controls all container oriented elements." /
  font = Fonts('DocFont')
  foreground = colors('docfg')
  background = colors('docbg');
style Document from Container
  "Abstract. Controls the various document bodies." /
  htmldoctype = "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">"
  htmlcontenttype = "text/html"
  protectspecialchars = auto
  linkcolor = colors('link2')
  activelinkcolor = colors('link1')
  visitedlinkcolor = colors('link1');
style Body from Document
  "Controls the Body file." /
  rightmargin = 8
  leftmargin = 8
  pagebreakhtml = html('PageBreakLine');
style SysTitleAndFooterContainer from Container
  "Controls container for system page title and system page footer." /
  rules = NONE
  frame = VOID
  outputwidth = 100%
  cellpadding = 1
  cellspacing = 1
  borderwidth = 0;
style TitleAndNoteContainer from Container
  "Controls container for procedure defined titles and notes." /
  rules = NONE
  frame = VOID
  outputwidth = 100%
  cellpadding = 1
  cellspacing = 1
  borderwidth = 0;
style TitlesAndFooters from Container
  "Abstract. Controls system page title text and system page footer text." /
  font = Fonts('TitleFont2')
  background = colors('systitlebg')
  foreground = colors('systitlefg');
style SystemTitle from TitlesAndFooters
  "Controls system title text." /
  font = Fonts('TitleFont');
style SystemFooter from TitlesAndFooters
  "Controls system footer text." /
  font = Fonts('TitleFont');
style PageNo from TitlesAndFooters
  "Controls page numbers for printer" /
  vjust = T
  just = R
  cellspacing = 0
  cellpadding = 0
  font = fonts('strongFont');
style ProcTitle from TitlesAndFooters
  "Controls procedure title text." /
  background = colors('proctitlebg')
  foreground = colors('proctitlefg');
style Output from Container

```

```

"Abstract. Controls basic output forms." /
background = colors('tablebg')
rules = GROUPS
frame = BOX
cellpadding = 7
cellspacing = 1
bordercolor = colors('tableborder')
borderwidth = 1;
style Table from Output
"Controls overall table style.";
style UserText from Note
"Controls the TEXT= style";
style Cell from Container
"Abstract. Controls general cells.";
style Data from Cell
"Default style for data cells in columns." /
foreground = colors('datafg')
background = colors('datavg');
style HeadersAndFooters from Cell
"Abstract. Controls table headers and footers." /
font = fonts('HeadingFont')
foreground = colors('headerfg')
background = colors('headerbg');
style Header from HeadersAndFooters
"Controls the headers of a table.";
style RowHeader from Header
"Controls row headers.";
style Footer from HeadersAndFooters
"Controls table footers.";
style RowFooter from Footer
"Controls a row footer (label).";
end;
run;

```

Appendix B: “Safe” Fonts

“Safe” fonts ^{1,2}
Times New Roman, Times
Arial, Helvetica
Arial Black, Arial, Helvetica
Book Antigua, Times New Roman, Times
Courier New, Courier
Comic Sans MS, Arial, Helvetica
Verdana, Arial, Helvetica
Impact , Arial Black, Helvetica
Georgia, Times New Roman, Times
News Gothic MT, Arial, Helvetica
Tahoma, Arial, Helvetica
Trebuchet MS, Arial, Helvetica

¹ Although these fonts are fairly safe, good alternate fonts have also been listed for each item. Also, this list is based on standard Windows fonts. If you have a lot of Mac users, you may want to list some Mac fonts like Chicago, Geneva, Helvetica, Monaco, New York, Times and Palatino as alternatives.

² Two very unsafe fonts are SAS Monospace and SAS Monospace Bold.

Appendix C: Color Names

White	Cornsilk	Antiquewhite	Seashell	Linen	Ivory	Floralwhite
Snow	Azure	Mintcream	Ghostwhite	Honeydew	Aliceblue	Beige
Oldlace	Bisque	Moccasin	Wheat	Navajowhite	Blanchedalmond	Tan
Gray	Lightgrey	Darkgray	Dimgray	Gainsboro	Silver	Whitesmoke
Black	Darkslategray	Slategray	Lightslategray	Lemonchiffon	Khaki	Darkkhaki
Brown	Sienna	Chocolate	Saddlebrown	Sandybrown	Burlywood	Peru
Red	Tomato	Darkred	Indianred	Mistyrose	Lavenderblush	Firebrick
Crimson	Maroon	Peachpuff	Goldenrod	Darkgoldenrod	Palegoldenrod	Lavender
Orange	Darkorange	Orangered	Forestgreen	Greenyellow	Lime	Lightgoldenrodyellow
Yellow	Lightyellow	Gold	Springgreen	Darkolivegreen	Olive	Limegreen
Green	Lightgreen	Darkgreen	Mediumseagreen	Mediumspringgreen	Palegreen	Olivedrab
Lawngreen	Chartreuse	Yellowgreen	Paleturquoise	Darkseagreen	Aquamarine	Mediumaquamarine
Teal	Lightseagreen	Seagreen	Darkblue	Mediumturquoise	Turquoise	Darkturquoise
Darkcyan	Cyan	Lightcyan	Mediumslateblue	Lightskyblue	Skyblue	Deepskyblue
Blue	Lightblue	Mediumblue	Steelblue	Darkslateblue	Powderblue	Cornflowerblue
Royalblue	Dodgerblue	Slateblue	Plum	Cadetblue	Mediumorchid	Darkorchid
Navy	Midnightblue	Lightsteelblue	Mediumvioletred	Orchid	Thistle	Rosybrown
Purple	Mediumpurple	Indigo	Fuchsia	Palevioletred	Magenta	Darkmagenta
Violet	Darkviolet	Blueviolet	Salmon	Deeppink	Coral	Lightcoral
Pink	Lightpink	Hotpink	Lightsalmon	Darksalmon		

Appendix D: Final HTML program

```

proc template;
  define style Styles.Custom;
    parent = Styles.Default;
    STYLE SystemTitle /
      FONT_FACE = "Comic Sans MS, Helvetica, sans-serif"
      FONT_SIZE = 5
      FONT_WEIGHT = bold
      FONT_STYLE = roman
      FOREGROUND = midnightblue
      BACKGROUND = white;
    STYLE SystemFooter /
      FONT_FACE = "Comic Sans MS, Helvetica, sans-serif"
      FONT_SIZE = 2
      FONT_WEIGHT = bold
      FONT_STYLE = italic
      FOREGROUND = midnightblue
      BACKGROUND = white;
    STYLE Header /
      FONT_FACE = "Times New Roman, Helvetica, sans-serif"
      FONT_SIZE = 4
      FONT_WEIGHT = medium
      FONT_STYLE = roman
      FOREGROUND = midnightblue
      BACKGROUND = white;
    STYLE Data /
      FONT_FACE = "Times New Roman, Helvetica, sans-serif"
      FONT_SIZE = 2
      FONT_WEIGHT = medium
      FONT_STYLE = roman
      FOREGROUND = black
      BACKGROUND = white;
    STYLE Table /
      FOREGROUND = black
      BACKGROUND = white
      CELLSPACING = 0
      CELLPADDING = 3
      FRAME = HSIDES
      RULES = NONE;
    STYLE Body /
      FONT_FACE = "Arial, Helvetica, sans-serif"
      FONT_SIZE = 3
      FONT_WEIGHT = medium
      FONT_STYLE = roman
      FOREGROUND = black
      BACKGROUND = white;
    STYLE SysTitleAndFooterContainer /
      CELLSPACING=0;
  end;
run;

ods listing close;
ods noptitle;

ods html file="sample4j.html" style=Custom;

  title 'Class List';
  footnote 'See www.laurenhaworth.com for more ODS papers and examples';
  proc means data=sashelp.class min median max maxdec=1;
  run;

ods html close;

```

Appendix E: Final RTF program

```

proc template;
  define style Styles.Custom;
    parent = Styles.RTF;
    STYLE SystemTitle /
      FONT_FACE = "Comic Sans MS"
      FONT_SIZE = 5
      FONT_WEIGHT = bold
      FONT_STYLE = roman
      FOREGROUND = midnightblue
      BACKGROUND = white;
    STYLE SystemFooter /
      FONT_FACE = "Comic Sans MS"
      FONT_SIZE = 2
      FONT_WEIGHT = bold
      FONT_STYLE = italic
      FOREGROUND = midnightblue
      BACKGROUND = white;
    STYLE Header /
      FONT_FACE = "Times New Roman"
      FONT_SIZE = 4
      FONT_WEIGHT = medium
      FONT_STYLE = roman
      FOREGROUND = midnightblue
      BACKGROUND = white;
    STYLE Data /
      FONT_FACE = "Times New Roman"
      FONT_SIZE = 2
      FONT_WEIGHT = medium
      FONT_STYLE = roman
      FOREGROUND = black
      BACKGROUND = white;
    STYLE Table /
      FOREGROUND = black
      BACKGROUND = white
      CELLSPACING = 0
      CELLPADDING = 3
      FRAME = HSIDES
      RULES = NONE;
    STYLE Body /
      FOREGROUND = black
      BACKGROUND = white
      TOPMARGIN=1in
      BOTTOMMARGIN=1in
      LEFTMARGIN = 1in
      RIGHTMARGIN = 1in;
    STYLE SysTitleAndFooterContainer /
      CELLSPACING=0;
  end;
run;

ods listing close;
ods noptitle;
options nodate nonumber;

ods rtf file="sample5.rtf" style=Custom bodytitle;

  title 'Class List';
  footnote 'See www.laurenhaworth.com for more ODS papers and examples';
  proc means data=sashelp.class min median max maxdec=1;
  run;

ods rtf close;

```

Appendix F: Final PDF program

```

proc template;
  define style Styles.Custom;
    parent = Styles.Printer;
    STYLE SystemTitle /
      FONT_FACE = "Comic Sans MS"
      FONT_SIZE = 5
      FONT_WEIGHT = bold
      FONT_STYLE = roman
      FOREGROUND = midnightblue
      BACKGROUND = white;
    STYLE SystemFooter /
      FONT_FACE = "Comic Sans MS"
      FONT_SIZE = 2
      FONT_WEIGHT = bold
      FONT_STYLE = italic
      FOREGROUND = midnightblue
      BACKGROUND = white;
    STYLE Header /
      FONT_FACE = "Times New Roman"
      FONT_SIZE = 4
      FONT_WEIGHT = medium
      FONT_STYLE = roman
      FOREGROUND = midnightblue
      BACKGROUND = white;
    STYLE Data /
      FONT_FACE = "Times New Roman"
      FONT_SIZE = 2
      FONT_WEIGHT = medium
      FONT_STYLE = roman
      FOREGROUND = black
      BACKGROUND = white;
    STYLE Table /
      FOREGROUND = black
      BACKGROUND = white
      CELLSPACING = 0
      CELLPADDING = 3
      FRAME = HSIDES
      RULES = NONE;
    STYLE Body /
      FOREGROUND = black
      BACKGROUND = white;
    STYLE SysTitleAndFooterContainer /
      CELLSPACING=0;
  end;
run;

ods listing close;
ods noptitle;
options nodate nonumber;

ods pdf file="sample5.pdf" style=Custom notoc;

  title 'Class List';
  footnote 'See www.laurenhaworth.com for more ODS papers and examples';
  proc means data=sashelp.class min median max maxdec=1;
  run;

ods pdf close;

```