

Paper 035-2010

Getting the Right Report (Again): Your Compatibility Guide for ODS PDF 9.2

Scott Huntley and Bari Lawhorn, SAS Institute Inc., Cary, NC

ABSTRACT

ODS PDF in SAS® 9.2 looks and acts different from earlier versions. Why? What changed? How can I maintain the same look that I like? This paper written in collaboration with SAS® Technical Support and R&D will take commonly asked questions submitted to SAS Technical Support about ODS PDF and give you the answers. Once you understand how things work and why things changed, you'll have the tools you need to create that perfect report in SAS 9.2.

INTRODUCTION

You have received and installed SAS 9.2. Now you are anxious to see all the new features you've heard about at previous SAS Global Forums. You run some of your established ODS PDF jobs to see how they run on the new system. Good, the log shows no errors, but you notice the pagination is different from earlier releases. The fonts also look different. The file size is larger. What do you do? A normal first step is to call SAS Technical Support. If you ever have had these types of questions, then this is the paper for you. Bari Lawhorn in SAS Technical Support and Scott Huntley in SAS R&D will go through several main differences between SAS® 9.1.3 and SAS 9.2 in relation to ODS PDF output. This paper will be unique in the fact that each author will give you his perspective on the issues. We will dig into some of the big differences between SAS 9.1.3 and SAS 9.2 as they relate to ODS PDF. After that we will show off some interesting things we have learned along the way to make your life just a little easier.

CHANGE ... GOOD OR BAD?

Every new release of SAS has lots of cool new things. ODS PDF in SAS 9.2 is no exception. Requests from SAS Technical Support and the SASWare Ballot are incorporated into every release. Fixing known defects is always a high priority. One important thing SAS developers try hard to do is keep backward compatibility. Sometimes it's impossible when a new feature or defect fix makes the old way obsolete. We are going to tackle the changes that occurred in SAS 9.2 that made the output look totally different than it did in SAS 9.1.3. We went through the common questions and concerns that were brought to Technical Support about ODS PDF in SAS 9.2. Each of us will comment about each item. Scott from Research and Development will explain why the change was made and how it affects you. Bari from Technical Support will give you the tools to change your output to make it look like it did in the past. In doing this, you'll learn how things have improved in the new release and we'll give you the power to keep your existing reports looking the same.

IMAGES ... OBJECTS ARE BIGGER THAN THEY APPEAR

Issue: My images in SAS 9.2 appear bigger than they were in SAS 9.1.3. Now my output has shifted and my pagination is totally different.

R&D response

Using images is a great way to draw attention to your output. The way we processed images in SAS 9.2 changed and now some SAS 9.2 output does not match SAS 9.1.3. Image sizes in your output are critical because the whole appearance of your report depends on a certain image size. Also the size of the image greatly effects how much space is available in your report. A common way to put images in your output is to use the PREIMAGE style attribute. Using PROC TEMPLATE and updating the table style element, we can have an image appear before a table. The following code using SAS 9.2 runs and produces a simple table with an image above it:

```
options nodate nonumber;
proc template;
  define style styles.mystyle;
    parent=styles.printer;
    style table from table /
      preimage="c:\temp\sas_logo.jpg";
  end;
run;
```

```

title;
ods pdf file="&sysver..pdf" style=styles.mystyle;
proc print data=sashelp.class(obs=3);run;
ods _all_ close;

```



Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0

Now we will run the same code with SAS 9.1.3 to produce this output:



Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0

What happened to my image? Why is it smaller with SAS 9.1.3? To completely explain we need to understand some background information first. Let's talk about DPI (dots per inch). Basically having a higher image DPI will produce clearer and more detailed output. Now if you look at the properties of your image you can locate its DPI. The image that I'm using has a horizontal and vertical resolution set to 100 DPI.

Before SAS 9.2 we did not take into account the DPI of the image. We just grabbed the image, applied some internal math and placed it into the output. The image may have appeared smaller because the image's resolution was not the same as our file's resolution. We would update the image's resolution to work with the output file's resolution. Sometimes this would make the image size smaller.

Now in SAS 9.2 we made some internal changes. We correctly honor the size and the DPI of the image used. We obtain the image's DPI and perform internal calculations for placement and appearance. The image you see in your output in SAS 9.2 is the correct size and resolution. This change was an improvement.

Depending on your image's DPI, your initial output may look different, but there is a way to make your output in SAS 9.2 look just like it did in SAS 9.1.3. Creating your image with the desired DPI and size is a good start. Open the image in a tool such as Microsoft Paint. Print it out and you'll see the size of the image and how it will appear in your PDF output.

Technical Support response

Technical Support got many calls in SAS 9.1.3 reporting problems with large images that were being resized when they were included in ODS PDF output. In SAS 9.1.3 the only solution was to resize the image outside of SAS using an image tool and then include the newer (smaller) image in the PDF file. Now in SAS 9.2 we have the ability to *make* images with DPI control, and correspondingly, we take into account the DPI of any image referenced by the PREIMAGE=, POSTIMAGE=, or BACKGROUNDIMAGE= style attributes. Knowing the DPI of your image is key, and creating or recreating an image with a DPI of 150 is your best bet for matching output to SAS 9.1.3. There are several good image editing tools on the Web to help you in change your image's DPI.

TEXT STATEMENTS AND TITLES ... WHICH SHOULD COME FIRST?

Issue: I have some PDF text statements that were between my titles and my first procedure output. Now these text statements appear above the title statement.

R&D response

Using text statements is a great way to insert a text string into your output without using titles or footnotes. The text statement allows you to put a text string anywhere between objects (tables or graphs) in the output. A very common situation is the ability to put a text statement after the title on page 1 but before an object. For example, in the following code a simple text statement comes between a title and an object:

```
options nodate nonumber;
ods escapechar = '^';
title 'This is the title';
ods pdf file="&sysver..pdf" startpage=no;
ods pdf text = '^S={foreground=blue font_size=12pt}Is this text statement before the
title or the table';
proc print data=sashelp.class(obs=3);run;
ods _all_ close;.
```

I added some simple style attributes to enhance the appearance of the text statement. I added a blue color to the text as well as making it 12pt. Here is the output using SAS 9.1.3:

This is the title

Is this text statement before the title or the table

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0

Using STARTPAGE=NO, I prevented the table being moved to a new page. The title statement is first, then the text statement followed by the table. Now what happens if I run this same code using SAS 9.2?

Is this text statement before the title or the table

This is the title

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0

Wow! The text statement is now above the title statement. What happened here? Again we have a situation where output in SAS 9.1.3 doesn't look the same as it does in SAS 9.2. There were some requests in which customers wanted the ability to do just this. The explanation behind this is how we process a page of output in ODS PDF.

When I say "page of output" I don't mean a physical page. A page of output is how much information can fit within the specified dimensions before it overflows. Depending on several things such as margins, font size, and table widths, a page of output can hold more or less information. Looking over the SAS code, we see that we set STARTPAGE=NO to prevent a page break between the text statement and the table. So when the page processing started we did the title first and then the text statement. Since we told it not to go to a new page if room exists, we processed the table.

Now what happened in SAS 9.2? Basically the same thing except for one small little detail: now a text statement can be processed before page processing occurs. That is why you see the text statement before the title. The text statement isn't necessarily viewed as an output object, so the title lets it pass by first. The big question now is: How do I get my output to mimic SAS 9.1.3? It's quite simple to do. Let's examine this code :

```
options nodate nonumber;
ods escapechar = '^';
title 'This is the title';
ods pdf file="&sysver..pdf" startpage=now;
ods pdf startpage=no;
ods pdf text = '^{style [foreground=blue font_size=12pt]}Is this text statement before
the title or the table';
proc print data=sashelp.class(obs=3);run;
ods _all_ close;
```

I've made a few changes. First I changed the STARTPAGE attribute on the PDF statement. Its setting is NOW. This means "Give me a new page now, please." This will run the new page processing and give me my title now. Next I've reset the STARTPAGE setting to NO to prevent a page break between my text statement and my table. Finally, I've updated the STYLE setting in my text setting to show the new STYLE syntax available in SAS 9.2. Past SAS Global Forum papers have discussed the new STYLE syntax. Now the output is the same as before and you've learned a little about page processing in ODS PDF at the same time.

Technical Support response

You know that phrase "Be careful what you ask for"? This is one of those cases. Technical Support has asked the development team for years to "please be consistent with the ODS destinations." "Please make RTF and PDF behave similarly." Well, we asked and development responded. In the RTF destination, the TEXT= statement does not force a page break; it is truly "inline text." In SAS 9.1.3, code like the following creates *one* page of output, but similar code to create PDF output creates *two* pages of output:

```
ods rtf file="&sysver..rtf";
ods rtf text = 'text statement ';
proc print data=sashelp.class(obs=3);run;
ods _all_ close;
```

The change in pagination behavior contributed to the title/text differences, and we hope that this move toward consistency between the destinations is an improvement.

DID MY LAYOUT EXPAND TO MULTIPLE PAGES?

Issue: I know ODS LAYOUT is pre-production in SAS 9.1.3 and SAS 9.2, but my LAYOUT output has gone from one page to two pages using SAS 9.2.

R&D response

ODS LAYOUT is a great way to put your output on the page where you want it. LAYOUT is pre-production in SAS 9.1.3 and SAS 9.2, but that doesn't mean there aren't jobs out there using it to the fullest. In ODS PDF we worked hard on graph integration with PDF in SAS 9.2. Some of that work with graphs affected jobs that use LAYOUT. One common question is about LAYOUT with columns and graphs. Here is an example:

```
options nodate nonumber orientation=landscape;
title;
options reset=all device=sasprtc;
ods pdf file='layout_v913.pdf' notoc;

ods layout start columns=2 column_widths=(5in 5in);
ods region;
proc tabulate data=sashelp.prdsal2(where=(product in('BED' 'CHAIR')))
format=dollar12.2;
class country product;
var actual predict;
table country all="Total", product*(actual predict)*sum="";
run;

ods region;
Proc gchart data=sashelp.class;
    vbar name / sumvar=age description="";
Run;Quit;

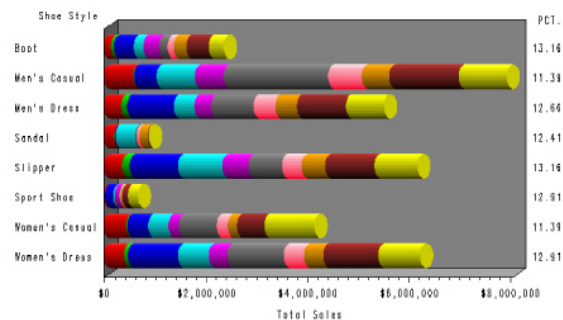
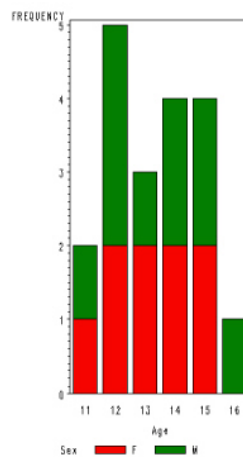
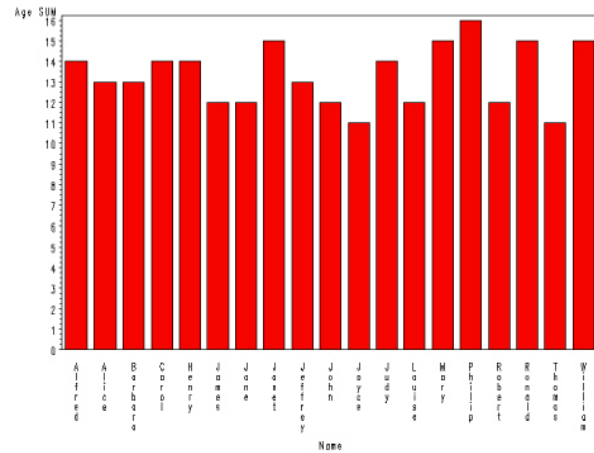
ods region;
Proc gchart data=sashelp.class;
    vbar age / subgroup=sex discrete cframe=white;
Run;Quit;

ods region;
proc gchart data=sashelp.shoes;
    hbar3d product / sumvar=sales
        subgroup=region
        shape=cylinder
        percent
        patternid=subgroup
        nolegend
    ;
    label product='Shoe Style';
run;
quit;

ods layout end;
ods pdf close;
```

The preceding code is putting a tabulate table and three graphs on the same page. Using LAYOUT with COLUMNS=2, all will fit on the same page. Here is how the output looks when running SAS 9.1.3:

	Product			
	BED		CHAIR	
	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales
Country				
Canada	\$790,506.00	\$850,600.80	\$782,128.80	\$847,395.00
Mexico	\$535,408.20	\$605,701.80	\$515,390.40	\$571,402.80
U.S.A.	\$2384337.64	\$2453193.80	\$2397864.42	\$2579405.29
Total	\$3710251.84	\$3909496.40	\$3695383.62	\$3998203.09



This output looks great and gives the desired results. When this code is run in SAS 9.2, you will only get the tabulate table on page 1. Page 2 of the output will have the first graph. Page 3 will be blank and page 4 will contain the final two graphs. What happened now? Why did my graphs get put on different pages?

As I mentioned before, this is caused by the work done with graph interaction with ODS PDF. ODS LAYOUT is pre-production. Pre-production means that code has been written, but that code hasn't been through the rigorous SAS testing procedures. Changes made to fix the graph and ODS PDF interaction affected LAYOUT. Does this mean my LAYOUT jobs won't run in SAS 9.2? All is not lost; we can still get that LAYOUT appearance you had in SAS 9.1.3. We need to tell LAYOUT more information about the region size or the graph size to fix the problem. Either change is easy to make, so let's explain how each change works. The first change is to tell LAYOUT more information about region sizes. Adding a HEIGHT attribute on the REGION statement will limit the graph height. Look at this code snippet:

```
ods region height=3.5in;
```

By adding the HEIGHT attribute to every REGION statement that is before the PROC GCHART statements, we limit the graph size inside that region. This prevents the graphs from getting too large and causing page breaks in SAS 9.2. The other solution is to use the graph options statement GOPTIONS to set the graph sizes. Here is the code snippet:

```
goptions reset=all device=sasprtc hsize=3.5in vsize=3.5in;
```

Using this statement will limit all graphs to the defined horizontal and vertical dimension. Again limiting the graph sizes will prevent incorrect page breaks. Adding some additional statements to your LAYOUT program allows you to get the same look in SAS 9.2 as you did in SAS 9.1.3.

Technical Support response

Until we launched SAS 9.2, we had *no* idea how many people were implementing pre-production code into their SAS 9.1.3 code. We certainly knew that people were asking for more flexibility to place graphs and tables together on a page, and that we were scrambling to find documentation of any sort to offer customers interested in ODS LAYOUT. Now that it's received more attention from the development team and from customers alike, we are realizing that this destination is truly amazing. Offering absolute and irregular placement of tables and graphs on a printed page is something to get excited about. Having to make people recode? Not so exciting. However, considering the new features and functionality that ODS LAYOUT offers, the HEIGHT= specification addition is hopefully a small price to pay for the ability to control tables and graphs on a printed page.

フォント – HOW TO REPRESENT DBCS CHARACTERS IN PDF, OR ARIAL, WHEREFORE ART THOU ARIAL?

Issue: What happened to my fonts? I ran my SAS 9.1.3 job with SAS 9.2 and everything looks different. My output tables look bigger.

R&D response

As mentioned in a past paper (Huntley 2006), SAS now licenses three font families from Agfa Monotype Corporation. Each of those font families has four members. In addition to these three font families, there are 10 other fonts that we license. Two are symbol fonts and eight have support for Asian languages. These typefaces will yield 100% metric compatibility to the Microsoft Core Fonts when used in the same computing environment:

Albany AMT = Arial
 Thorndale AMT = Times New Roman
 Cumberland AMT = Courier New

Starting in SAS 9.2, using these fonts allows SAS applications and output to display and print text consistently, portably, efficiently, and internationally. These typefaces can be rendered in weights and styles of normal, italic, bold and bold italic. These fonts are available on every platform that SAS ships.

The simple solution is to create a customized style with PROC TEMPLATE and use the fonts that were used in SAS 9.1.3. Bari explains how this is done in the following paragraphs.

Technical Support response

Fonts are important for how a business conveys and supports its message. They are also important for representing non-English characters. Many common fonts used by English speakers do not contain the glyphs necessary to represent the double-byte character sets (DBCS) used by our Asian neighbors. Technical Support fielded many calls asking for the ability to use and embed DBCS fonts in PDF files created by versions prior to SAS 9.2. When the SAS Monotype fonts were purchased, we cheered for our customers in China, Japan, and Korea who can now get their native language characters represented in PDF format directly with SAS. In order to deliver and showcase these fonts, the default styles and registry information makes use of the most common fonts: Thorndale AMT, Albany AMT and Cumberland AMT.

While an attempt was made to closely mimic Times, Arial, and Courier, the complication this poses to those who do not need Kanji characters in their output is that their beloved fonts Times, Arial and Courier are no longer in use in their files. If the spacing of the output file is very specific, this can also result in output that wraps differently across a page, or paginates differently across pages. The workaround is to change your PROC TEMPLATE style to use the fonts Times, Arial and Courier, as shown:


```

proc template;
  define style Styles.myPrinter;
    parent = styles.printer;
    style fonts /
      /* use TIMES if on non-Windows platform */
      'TitleFont2' = ("Times",12pt,bold italic)
      'TitleFont' = ("Times",13pt,bold italic)
      'StrongFont' = ("Times",10pt,bold)
      'EmphasisFont' = ("Times",10pt,italic)
      'FixedEmphasisFont' = ("Courier",9pt,italic)
      'FixedStrongFont' = ("Courier",9pt,bold)
      'FixedHeadingFont' = ("Courier",9pt,bold)
      'BatchFixedFont' = ("SAS Monospace, Courier",6.7pt)
      'FixedFont' = ("Courier",9pt)
      'headingEmphasisFont' = ("Times",11pt,bold italic)
      'headingFont' = ("Times",11pt,bold)
      'docFont' = ("Times",10pt);
  end;
run;

ods pdf file="times_92.pdf" style=styles.myprinter;
title "testing";
footnote "another test";
proc print data=sashelp.class;
run;
ods pdf close;

```

WHAT'S YOUR TYPE?

Issue: Looking at the font properties of the PDF file, I see I'm not using all the fonts in SAS 9.2 that I used in SAS 9.1.3. Am I missing something?

R&D response

Using Adobe Acrobat, you can see the fonts that are included in your file. Open the file and then press CTRL+D to open a properties window. Then click on the **Fonts** tab to see all the fonts in your file. You'll notice a few things that are different in SAS 9.2. As mentioned before, having fonts that are used globally is very important. Not every person who reads a PDF file will want the fonts to have 'single byte' or 'Latin' encoding. We need to be able to specify a font that can be language specific. Starting in SAS 9.2, Cumberland AMT or a Nationale-specific default font is the default font replacing Courier. Also you have the ability to set a language-specific font for all Universal Printers if required. The move to TrueType fonts from Type 1 was made because TrueType fonts have a more robust character set than Type 1 fonts. This lends them to better true global language support. In SAS 9.2, only the fonts actually used in the document are listed in the properties font list.

Technical Support response

We hear it a lot in Technical Support—fonts are a way for our customers to convey their message to the world with their unique "look." If that particular "look" is delivered via Type 1 fonts, then our use of TrueType fonts needs to be adjustable. Luckily it is. With changes to your SAS registry, you can adjust the appropriate keys to register and use the type 1 fonts that were in effect in SAS 9.1.3. A good start is to contact Technical Support for guidance in how to update your SAS registry. Together we can help you get back the "type" of output you expect.

THOSE YUMMY NEW FONTS MAKE MY FILE A LITTLE ...HEFTY.

Issue: My PDF output file is huge now. What increased my file size so much?

R&D response

As mentioned in a past paper (Huntley 2006), with these new TrueType fonts also comes the possibility of larger file sizes when embedding is enabled. In SAS 9.2 we will have four new options to control file sizes. Here is a brief description of the new options:

1. FONTEMBEDDING / NOFONTEMBEDDING This option will enable or disable font embedding. This option will allow more direct user control over font embedding and, therefore, output file size. For SAS 9.2 this option is only available for PDF and Scalable Vector Graphics (SVG) drivers. PostScript currently ignores this option and other drivers render the fonts as part of the image. The default for this option is FONTEMBEDDING.
2. UPRINTCOMPRESSION / NOUPRINTCOMPRESSION This option will enable or disable UP compression. Used with the next two options, this option is a master switch which enables or disables file compression. The default for this option is UPRINTCOMPRESSION. The alias is UPC / NOUPC.
3. DEFLATION= n , where n is 0 to 9 and 6 is the default. This option controls the level of data compression used by the Deflate compression method. It is only used by the PDF and SVG drivers. The alias is DEFLATE.
4. JPEQUALITY= n , where n ranges from 0 to 100 and 75 is the default. This option controls image quality and compression such that image quality decreases as compression increases. A value of 0 will produce the lowest quality image and highest level of compression.

These are the new options available to you in SAS 9.2. Again Bari has the solution to the issue below.

Technical Support response

The shouts of joy for the ability to use new fonts and embed them came with corresponding jabs of pain when Technical Support saw the size of the PDF files created in SAS 9.2. In order to ensure that the fonts we ship with SAS 9.2 are available to anyone creating PDF files, regardless of their operating system or their locale setting, and to guarantee that anyone opening the PDF files sees the same results, the fonts have to be embedded. In order to embed fonts, the size of the PDF files has grown exponentially. If you are not in need of the new fancy fonts and you are generating your PDF files with the Adobe default fonts of Times, Arial, and/or Courier, setting the option NOFONTEMBEDDING, valid on an OPTIONS statement or as part of your configuration file (SASv9.CFG) will restore your PDF files to their pre-9.2 file size.

```
options nofontembedding;
```

Note: Just using NOFONTEMBEDDING will drastically reduce the size of the PDF file, but if the fonts are not changed via PROC TEMPLATE, or with STYLE= overrides in the code, the fonts listed in the file will still be the new fonts like Cumberland AMT, Thorndale AMT, and Albany AMT.

Some of the extra overhead is the loading of the new default font, Cumberland AMT into the PDF file(s). To prevent this, use:

```
options sysprintfont='Courier';
```

WHO IS IN CONTROL OF MY MARGINS?

Issue: My margins look different. Did something internal change all my margin settings?

R&D response

There are two easy ways to control the page margins in your PDF file. You can use PROC TEMPLATE or the OPTIONS statement to set the margins. The default setting for all four margins (Top, Bottom, Right and Left) is set to .25 inch. But what happens when you use both ways to set the margin? Which method wins? Who is in control of the setting?

In SAS 9.1.3, the rule was that PROC TEMPLATE would override the OPTIONS statement if they conflicted. I'm going to show the output of the same code using two different versions of SAS. Here is the code:

```

options leftmargin=2in nocenter nodate nonumber;

proc template;
  define style styles.margin;
    parent = Styles.Printer;
    style body from Document /
      leftmargin=.5in;
  end;
run;

title 'Options statement says 2 inches while template says .5 inch';
ods pdf file="margin.pdf" style=styles.margin;
proc print data=sashelp.class(obs=3);run;
ods pdf close;

```

Notice I'm setting the left margin in the OPTIONS statement along with NOCENTER attribute. So I'm telling SAS I want my output to *not* to be centered, but I want a 2-inch left margin. Then using PROC TEMPLATE, I'm telling SAS I want my body style to have a left margin of .5 inch. Here is how the output looks in SAS 9.1.3:

Options statement says 2 inches while template says .5 inch

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0

It's hard to show margins in a paper, but when I took this snapshot I purposely went outside the viewer to help show you some defining lines. The gray lines on the top and left are not part of the document. They are there to provide a reference for you. Go ahead and cut and paste the above code into your SAS session and run it. Then print it out and get out a ruler and check to see if that half inch is there.

Now I'm going to run the same code in SAS 9.2 to see who controls the margin now. Here is the output:

Options statement says 2 inches while template says .5 inch

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0

Now the OPTIONS statement is controlling the margin setting, since the left margin is now 2 inches. So in SAS 9.2 if the OPTIONS statement margin settings conflicts with the PROC TEMPLATE margin setting, the higher of the two

values will be accepted. The margin settings are very important in PDF because all the measurements are based on them. Having conflicting settings causes problems in the code. We made a decision to take the maximum value between the two conflicting settings. It is very easy to solve this issue. Pick one method to set your margins.

Technical Support response

Margins have always posed a “which came first—the chicken or the egg?” question. Should the margins on an OPTIONS statement override what’s specified in PROC TEMPLATE, or should PROC TEMPLATE’s style information always prevail? In this case, you decide. The default margins from the PRINTER style are set to _UNDEF_ so that the ODS PRINTER (PCL/PDF/PS) document can grab the margins from the default printer setup (you can see this if you run PROC TEMPLATE; source styles.printer; run;). The options settings are shown in the log with the following code:

```
proc options group=odsprint; run;
```

If there are no explicit margins set for the printer (as shown from the preceding code) or the style, the information for the margins comes from the SAS registry with values of .25IN:

```
proc registry list startat="\CORE\PRINTING\PRINTERS\PDF\DEFAULT SETTINGS";
run;
```

Many customers have asked for margin settings of zero, which is not supported. In order to get very very close to the edge of the printed page, we suggest a setting of .001IN, i.e.:

```
options leftmargin=0.001in rightmargin=0.001in topmargin=0.001in bottommargin=0.001in;
```

THINGS DISCOVERED ALONG THE WAY

Along the way in solving customer inquiries, Bari and I have run into cool things that can make your jobs even better. These are not show stoppers, but they are nice things to have when you need them. You may already know about some of these but hopefully you will find some of these items useful.

TEXT ON IMAGES

A popular request is to have an image that spans across the page and perhaps have some text embedded in that image. The key is image construction. ODS PDF just grabs the image from a location and puts it into the file. If you can craft your image correctly, you can create output like this:



Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5

The code is quite simple. Just write a text statement with a background image. You can set the width and height of the cell to contain the image. Also you'll need to add the text, set the font size and the justification. Here is the code:

```
options nodate nonumber;
ods escapechar = '^';
title;
ods pdf file="test.pdf" startpage=no;

ods pdf text = '^{style [cellwidth=8in cellheight=0.5in foreground=white
font_size=18pt vjust=center just=right
backgroundimage="c:\temp\orionstarHeader.jpg"]South Region}';
proc print data=sashelp.class(obs=1);run;

ods pdf close;
```

The important piece is the image. Notice how the image has some empty space at the right for me to add the text. Here is the image without the text added:

```
options nodate nonumber;
ods escapechar = '^';
title;
ods pdf file="test.pdf" startpage=no;

ods pdf text = '^{style [preimage="c:\temp\orionstarHeader.jpg"]}' ;
proc print data=sashelp.class(obs=1) ;run;

ods pdf close;
```



You can see how the image was altered when put into a cell with the dimensions predefined. Looking back at the code, you can see how I've justified the text centered vertically and right horizontally to get it in the right place. There is no magic behind this other than that the image must be set up correctly before including it in the SAS program.

URLS IN PDF WITHOUT THE BLUE BOX

a uniform resource locator or URL is often used in PDF files to help users jump somewhere important. Here is a quick example of how a URL looks in PDF:

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0

Here is the link to SAS

<http://www.sas.com>

As you hover your mouse over the blue box, you'll see the mouse change from an arrow to a pointing finger. Clicking the URL will take you to the defined location. A common question is how to get rid of the blue box surrounding the words. This can be accomplished by using the style attribute LINKCOLOR. If you change LINKCOLOR to be the same as your background color, you can have a link with no box.

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0

Here is the [link](http://www.sas.com) to SAS

<http://www.sas.com>

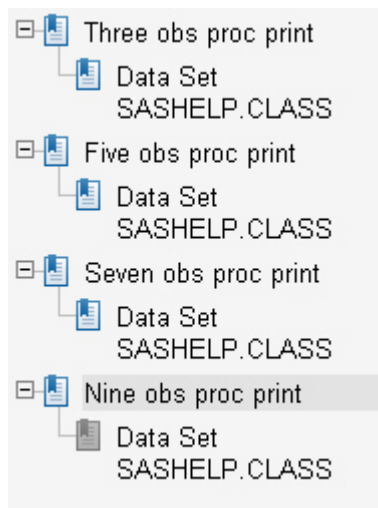
Here is the code to remove the blue box around the URL:

```
options nodate nonumber;
ods escapechar='^';
title;
ods pdf startpage=no;
proc print data=sashelp.class(obs=3);run;
ods pdf text = '^ {style [url="http://www.sas.com" linkcolor=white] Here is the
^ {style [color=blue]link} to SAS } ';
ods pdf close;
```

As you can see, the blue box is gone and the style attribute COLOR made the word "link" blue. The URL is still based on the cell, so putting the mouse anywhere on the cell will activate the URL code, but the blue word is a little nicer than the blue box.

CAN I PLEASE CONTROL MY PDF BOOKMARKS

A common request is the ability to control the PDF bookmarks inside your PDF output. These links allow a user to jump to any specified output object with a simple click. Recently Bari and I stumbled upon an easy way to control these bookmarks. Using the PDF setting BOOKMARKGEN, you can toggle it off and on between outputs. Here is a normal bookmark list containing four output objects (in this case they are PROC PRINT outputs):



I've used the statement ODS PROCLABEL to give descriptive names to my bookmarks. Here is my simple code :

```
ods escapechar="^";
ods pdf file='bookmark.pdf';

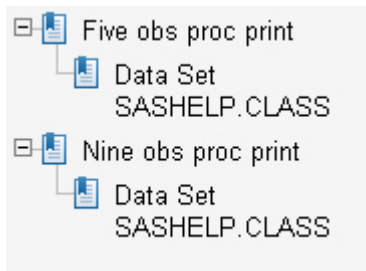
ods proclabel='Three obs proc print';
proc print data=sashelp.class(obs=3);run;

ods proclabel='Five obs proc print';
proc print data=sashelp.class(obs=5);run;

ods proclabel='Seven obs proc print';
proc print data=sashelp.class(obs=7);run;

ods proclabel='Nine obs proc print';
proc print data=sashelp.class(obs=9);run;
ods pdf close;
```

Okay, so now you really only want a bookmark for the 2nd and 4th entries. In the past you would probably use PROC DOCUMENT to remove the links and get the desired output you want. Using PROC DOCUMENT is a great alternative, but by using the BOOKMARKGEN option you can quickly get what you want.



Here is the code to produce those bookmarks:

```
ods escapechar="^";
ods pdf file='bookmark.pdf' bookmarkgen=off;

ods proclabel='Three obs proc print';
proc print data=sashelp.class(obs=3);run;

ods pdf bookmarkgen=on;
ods proclabel='Five obs proc print';
proc print data=sashelp.class(obs=5);run;

ods pdf bookmarkgen=off;
ods proclabel='Seven obs proc print';
proc print data=sashelp.class(obs=7);run;

ods pdf bookmarkgen=on;
ods proclabel='Nine obs proc print';
proc print data=sashelp.class(obs=9);run;
ods pdf close;
```

The only thing different is the BOOKMARKGEN setting being toggled off and on. You can see I started the PDF file with the BOOKMARKGEN set to off. I turned BOOKMARKGEN on for output objects #2 and #4, but turned it off for #3. This gives you limited control over your bookmarks.

NEW SPANROWS OPTION IN PROC REPORT

This isn't really something we discovered; this change was done on purpose. PROC REPORT added a new SPANROWS option in SAS 9.2. This was a top ten item on the 2007 SASWare Ballot. The SPANROWS option allows PDF, PRINTER, and measured RTF destinations to repeat group and order variable values across page breaks. It's a change in appearance that really looks great.

Here is some sample code to show what I'm talking about. First we start with SAS 9.1.3 and run this code:

```
ods pdf file = "repeat.pdf";

proc report data=sashelp.shoes nowd;
  col Region Product Subsidiary Stores Sales Inventory Returns;
  define Region / group;
  define Product / group;
  define Subsidiary / group;
  define Stores / group;
  define Sales / group;
  define Inventory / group;
  define Returns / group;
run;

ods pdf close;
```

This is the output:

Region	Product	Subsidiary	Number of Stores	Total Sales	Total Inventory	Total Returns
Africa	Boot	Addis Ababa	12	\$29,761	\$191,821	\$769
		Algiers	21	\$21,297	\$73,737	\$710
		Cairo	20	\$4,846	\$18,965	\$229
		Johannesburg	14	\$8,365	\$33,011	\$483
		Khartoum	24	\$19,282	\$105,370	\$700
		Kinshasa	16	\$13,921	\$70,736	\$553
		Luanda	8	\$6,081	\$51,572	\$325
		Nairobi	25	\$16,282	\$66,017	\$844
	Men's Casual	Addis Ababa	4	\$67,242	\$118,036	\$2,284
		Algiers	4	\$63,206	\$100,982	\$2,221
		Cairo	25	\$360,209	\$1,063,251	\$9,424
		Khartoum	1	\$9,244	\$16,230	\$478
		Luanda	3	\$62,893	\$149,864	\$1,701
		Addis Ababa	7	\$76,793	\$136,273	\$2,433
		Algiers	13	\$123,743	\$428,575	\$3,621

This is your standard gridded output from PROC REPORT. Once the column information is printed, it is not repeated on the next page. In this example, on the next page break the info stored in Region will not be repeated. Not having this information on the next page can make it difficult to remember where you were in your output.

By adding the SPANROWS option on the PROC REPORT statement like this:

```
proc report data=sashelp.shoes nowd spanrows;
```

the output now appears like this:

Region	Product	Subsidiary	Number of Stores	Total Sales	Total Inventory	Total Returns
Africa	Boot	Addis Ababa	12	\$29,761	\$191,821	\$769
		Algiers	21	\$21,297	\$73,737	\$710
		Cairo	20	\$4,846	\$18,965	\$229
		Johannesburg	14	\$8,365	\$33,011	\$483
		Khartoum	24	\$19,282	\$105,370	\$700
		Kinshasa	16	\$13,921	\$70,736	\$553
		Luanda	8	\$6,081	\$51,572	\$325
		Nairobi	25	\$16,282	\$66,017	\$844
	Men's Casual	Addis Ababa	4	\$67,242	\$118,036	\$2,284
		Algiers	4	\$63,206	\$100,982	\$2,221
		Cairo	25	\$360,209	\$1,063,251	\$9,424
		Khartoum	1	\$9,244	\$16,230	\$478
		Luanda	3	\$62,893	\$149,864	\$1,701
	Men's Dress	Addis Ababa	7	\$76,793	\$136,273	\$2,433
		Algiers	13	\$123,743	\$428,575	\$3,621

The difference is purely visual and it looks great. As you can see, the far left column which is showing Region is spanned across the Product column on its right. Then the Product column is spanned across the Subsidiary column on its right. What is not shown is that after a page breaks, all the columns are repeated on the next page so you can see exactly where you are in the report. This looks more like PROC TABULATE and better than the previous gridded output in SAS 9.1.3.

CONCLUSION

SAS 9.2 is full of new features and enhancements to make your output better. Some things have changed, and the look is different from SAS 9.1.3. As mentioned earlier, backward compatibility is very important and we strive to maintain that. We have shown you some major differences and explained how you can make some simple coding changes to get the look of your old SAS 9.1.3 output back. The continuous feedback offered to SAS developers via user group and technical support interactions has allowed the Base SAS product to develop into what we hope is the most powerful and flexible version of the software yet. Improvements to the offerings in the ODS PDF destination have led to changes that make the software better. Hopefully the explanations, sample code, and discussions offered in this paper help you get PDF output that is similar and *better* in SAS 9.2 than your PDF output generated by SAS 9.1.3.

REFERENCES

Huntley, Scott. 2006. "Let the ODS PRINTER Statement Take Your Output into the Twenty-First Century." *Proceedings of the SAS User's Group International Conference 2006*. Cary, NC: SAS Institute Inc. Available at <http://www2.sas.com/proceedings/sugi31/227-31.pdf>.

ACKNOWLEDGMENTS

The authors would like to thank Donna Antle, Tim Hunter, Kevin Smith, and Kathryn McLawhorn for their contributions to this paper.

CONTACT INFORMATION

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

Bari Lawhorn
SAS Institute Inc.
Building J, SAS Campus Drive
Cary, NC 27513
E-mail: Bari.Lawhorn@sas.com

Scott Huntley
SAS Institute Inc.
Building R, SAS Campus Drive
Cary, NC 27513
E-mail: Scott.Huntley@sas.com

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.