Paper 011-2010

Zoom, Zoom: Get Your Document to Scale on All Paper Sizes

Daniel O'Connor, SAS® Institute Inc., Cary, NC

ABSTRACT

This paper will help you learn how to scale your large reports so they will always fit the paper size. The paper includes tips and tricks of the Output Delivery System (ODS) style that will make things fit without paneling your tables or clipping your graphs. Scaling to fit is a very difficult problem; learn how to tell SAS to "just make it fit."

INTRODUCTION

Creating custom printed output can be very time consuming and difficult. Designing a complicated document that looks great on the display and also having it print in a legible fashion can be a daunting task. Many third-party formats have some degree of photo copier "Reduce" and "Enlarge" capabilities which often do not produce the most desirable output; leaving output with large blank areas that are not centered, or are distorted to fit the document. The challenges of integrating the dynamic task of reporting with the ability to scale contents is far more complicated than authoring a document in your typical word processing application. SAS has already developed many advanced techniques to accommodate these challenges. This paper will give you insight into the techniques that are used by default, and will introduce you to some more advanced techniques to enhance your reporting capabilities.

DISPLAY VS. THE PHYSICAL PAGE

Hypertext Markup Language (HTML) output is designed to be displayed on a device that has navigational capabilities, such as your computer monitor, hand held device, or cell phone. Each display device has unique display characteristics like size and resolution, and provides an appealing interactive experience. However, those devices almost never print very well. Output types like Portable Document Format (PDF), Postscript, Printer Command Language (PCL), or RichText Format (RTF) are all based on a physical page or printer characteristics. The physical page has many more limitations than a display, such as paper size, orientation, pagination, and margins. These real-estate limiting constraints are what make producing dynamic reports very challenging. Before delving into the sophisticated techniques that SAS has developed to allow you to control your reporting, let's examine some simple SAS options that allow you to customize your printed output.

PAPERSIZE

Specifies the page size to use while initializing a physical page based output destination.

Syntax:

PAPERSIZE = PAPER_SIZE_NAME | (width, height)

PAPER_SIZE_NAME

Specifies a predefined paper size. The default is either LETTER or A4, depending on the locale.

All of the available PAPER_SIZE_NAME values can be found in the SAS Registry at the following location [CORE\PRINTING\PAPER_SIZES]. The registry can either be accessed by submitting PROC REGISTRY or by using the SAS REGEDIT window in interactive SAS.

Some of the more common paper sizes are LETTER, LEGAL, or "4x6 Card". When the predefined paper size contains a space, remember to enclose the name in quotes.

(width, height)

Specifies paper width and height as a positive floating-point value with or without units. Valid units are "in" for inches or "cm" for centimeters. A number with no units will default to "in".

If you provide a paper size that is not supported by your printer the default paper size will be used. Please consult your SAS documentation for more information on these limitations.

Example 1:

```
Options papersize=legal;

Options papersize=letter;

Options papersize=”4x6 Card”;

Options papersize=(“8in”, “10in”);
```

ORIENTATION

Specifies the paper orientation.

Syntax:

ORIENTATION = PORTRAIT | LANDSCAPE | REVERSEPORTRAIT | REVERSELANDSCAPE

PORTRAIT	paper is taller than wide. This is the default.
LANDSCAPE	paper is wider than tall.
REVERSEPORTRAIT	reverse portrait enables printing on paper with pre-punched holes. The reverse side of the page is printed upside down.
REVERSELANDSCAPE	reverse landscape enables printing on paper with pre-punched holes. The reverse side of the page is printed upside down.

In general setting the paper orientation should be done before opening a page based destination.

Example 2:

```
Options orientation=landscape;

Ods pdf file=”Example2.pdf”;
```

MARGINS (TOPMARGIN, BOTTOMMARGIN, LEFTMARGIN, RIGHTMARGIN)

Specifies the print margins for the appropriate side of the page.

SYNTAX:

TOPMARGIN = *margin-size*

Margin size is a positive floating-point value with or without units. Valid units are “in” for inches or “cm” for centimeters. A number with no units will default to “in”.

Margin defaults come from the default printer being used. For more information about changing the default printer please see the SAS documentation for the PRINTERPATH option.

Example 3:

```
Options topmargin=”.5in”
        bottommargin=”.25in”
        leftmargin=”.25in”
        rightmargin=”.25in”;
```

AVAILABLE PHYSICAL SPACE

The available real-estate for reporting purposes is not only constrained by the physical paper size, but also by the margins, as you can see in the following figure.

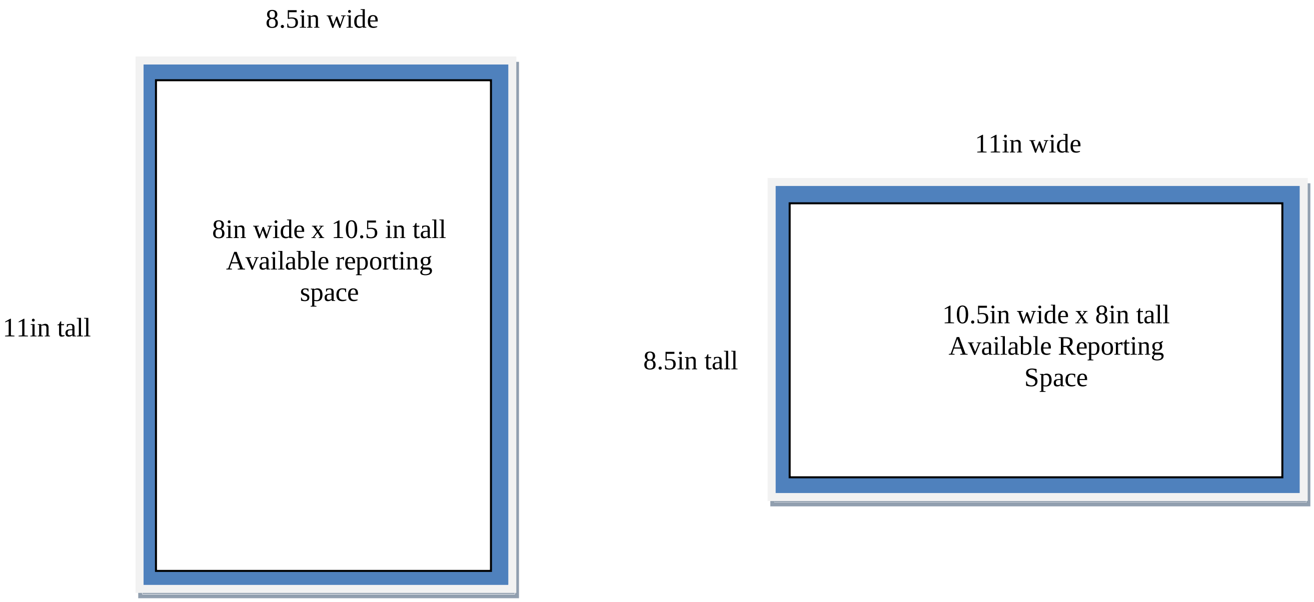


Figure 1. Letter Size Paper with .25in Margins with Portrait and Landscape Orientation

CONTROLLING DEFAULT PAGINATION

In an effort to allow the system to generate appropriate titles and footnotes for the output about to be generated, SAS begins a new page on every procedure boundary. SAS provides the STARTPAGE ODS option to allow customers to alter the default behavior and to better control pagination. Let's look at how the STARTPAGE option affects your output.

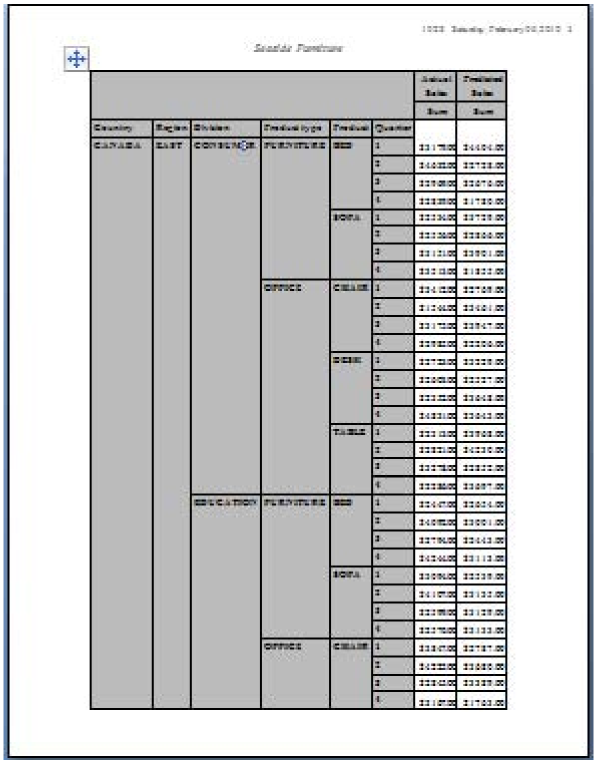
STARTPAGE = YES | NO | ON | OFF | NOW

YES | ON	inserts a new page at the beginning of each procedure. This is the default.
NO | OFF	no pages will be automatically inserted at the beginning of a procedure. Some PROCEDURES explicitly ask for a new page, which will still be honored, only the automatic pagination will be affected.
NOW	immediately start a new page.

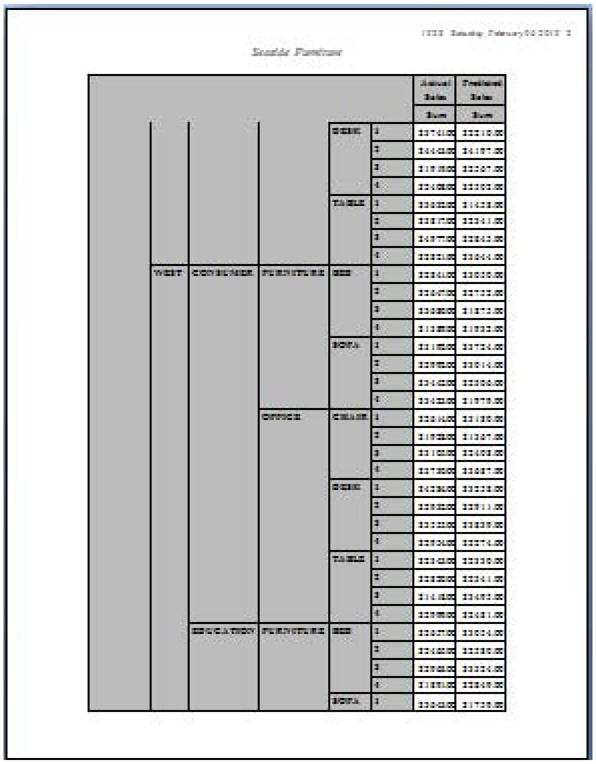
Despite the above heuristics, SAS always goes to the next page when the available vertical space has been exhausted. As SAS renders the output on a page it carefully keeps track of how much available space is left on the page. If ODS is in the middle of rendering a table and the available vertical space is exhausted, ODS automatically starts a new page, repeating the TITLES and FOONOTES, and regenerating any TABLE HEADER information about the subsequent page. ODS then continues rendering the remaining observations until every observation has been displayed.

Please note one exception to this rule. The Traditional RTF destination only does horizontal measurement, and the pagination is left up to Microsoft Word when the RTF file is displayed. SAS introduced a new RTF destination in SAS 9.2 called the TAGSETS RTF, that does both horizontal and vertical measurement. It is strongly recommended that customers wanting to produce RTF output for Microsoft Word integration use the more advanced TAGSETS

RTF destination. The transition from the Traditional RTF destination to the TAGSETS RTF destination is simple to make. Here are two quick examples:



Seaside Furniture



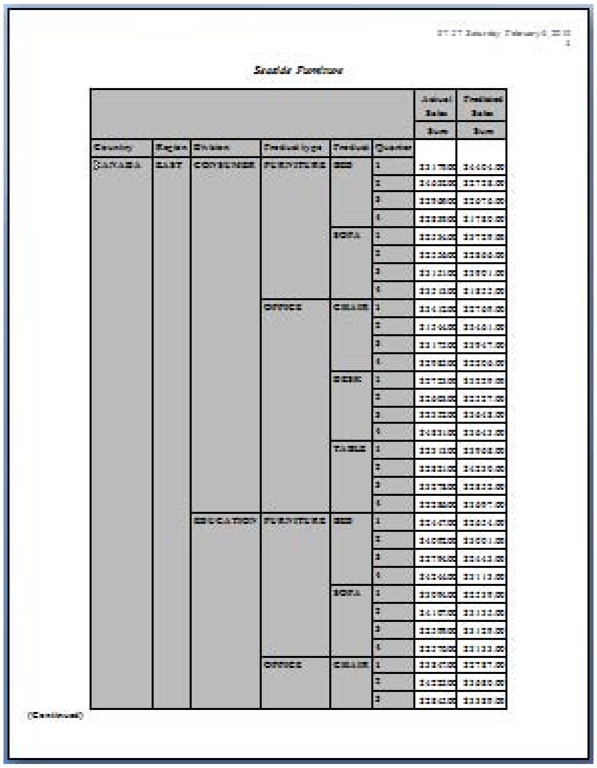
Seaside Furniture

Example 4:

```
title "Seaside Furniture";
ods rtf file="Example4.rtf";
proc tabulate data=sashelp.prdsale;
class country region division prodtype product quarter;
variable actual predict;
table country*region*division*prodtype*product*quarter,
      (actual*f=dollar8.2)*sum (predict*f=dollar8.2)*sum;
run;
ods rtf close;
```

Example 4 Traditional RTF Does Not Repeat Header Information

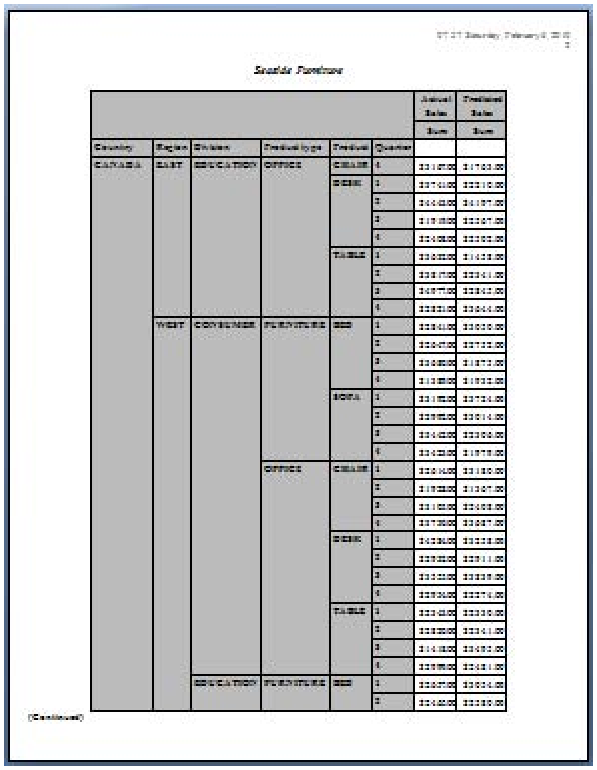
As mentioned in the previous paragraph, the traditional RTF destination leaves the pagination up to Microsoft Word, which does a very poor job of repeating row headers when table span multiple pages. In the above displays you will notice that when the table spans to page two, you lose context of the spanning columns 1-4. Let's compare the results of the traditional RTF output with the more sophisticated TAGSETS RTF destination in example 5.



Seaside Furniture

Country | Region | Division | Product type | Product | Quarter | Actual Sales Sum | Predicted Sales Sum

(Continued)



Seaside Furniture

Country | Region | Division | Product type | Product | Quarter | Actual Sales Sum | Predicted Sales Sum

(Continued)

Example 5:

```
title "Seaside Furniture";
ods tagsets.rtf file="Example5.rtf" uniform;
proc tabulate data=sashelp.prdsale;
class country region division prodtype product quarter;
variable actual predict;
table country*region*division*prodtype*product*quarter,
      (actual*f=dollar8.2)*sum (predict*f=dollar8.2)*sum;
run;
ods tagsets.rtf close;
```

Example 5 TAGSETS RTF Repeating Header Information

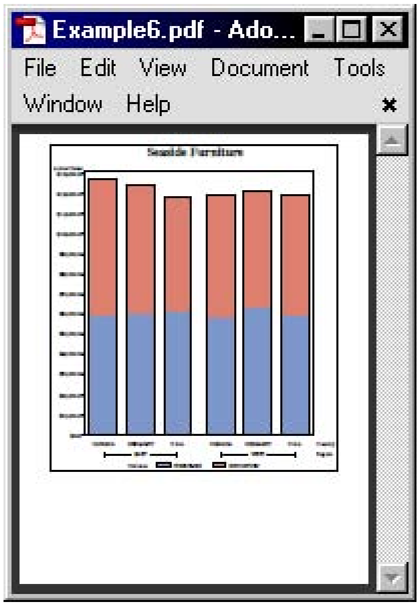
As you can see, the TAGSETS RTF destination does a really nice job of automatically repeating the column headings when a table spans multiple pages. This is also the default behavior for the PDF, Postscript, and PCL destinations. Note also that TAGSETS RTF generates the helpful continuation message when the table breaks pages. This is especially important for TABULATE, which can easily generate multiple tables. The continuation message distinguishes between two separate tables and a single table that has broken across pages.

HOW GRAPHS INTERACT

So what happens when a graph is being produced and there is insufficient vertical space on the page? You would never want to split a graph across a page, so ODS implicitly starts a new page and places the graph on the subsequent page. As of SAS 9.2, two unique graph architectures are available; the historical device-based graphics and the new ODS Graphics. These graphical architectures were developed for different purposes. Each has its own defaults and syntax for overriding the default graph dimensions. Here is a quick introduction and examples of how to alter the graph size using each architecture. For more information about either syntax, please refer to the *SAS/GRAPH 9.2: Reference,* or the *SAS/GRAPH Statistical Graphics Procedures Guide.*

DEVICE-BASED GRAPHICS (GRSEG)

The historical SAS/GRAPH® software procedures are all designed around a device definition. The device definition contains the default heights and width as well as many other relevant defaults. These are the defaults that are used for each output destination when no customer overrides have been specified. Use the GOPTIONS statement to override the default device setting.



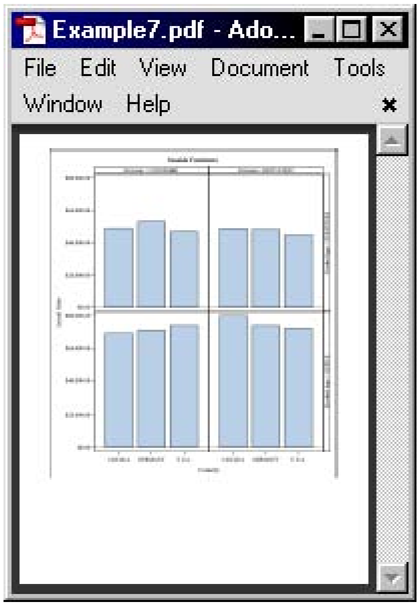
Example 6:

```
title "Seaside Furniture";
ods pdf file="Example6.pdf" notoc;

goptions hsize=7in vsize=8in border;

proc gchart data=sashelp.prdsale;
vbar country / group=region subgroup=division
sumvar=actual;
run;
ods pdf close;
```

Example 6 Using the GOPTIONS Statement to Resize a Graph

ODS GRAPHICS

ODS GRAPHICS is a new graphical architecture specifically designed to allow the creation of sophisticated, analytical, graphical displays. This architecture enhances the output from all analytical procedures and automatically produces the appropriate graphical display for the current analysis. Newly developed SAS/GRAPH software procedures further exploit this architecture by quickly producing custom graphical displays such as scatter plot matrixes, paneled displays, and sophisticated composite displays. With just a couple of statements you can produce stunning graphs that might have taken hundreds of lines of code in the past. Override the default ODS Graphics settings by using the “ODS graphics” statement.



Example 7:

```
title "Seaside Furniture";
ods pdf file="Example7.pdf" notoc;

ods graphics / width=7in height=8in;

proc sgpanel data=sashelp.prdsale;
panelby division prodtype/ layout=lattice;
vbar country / response=actual stat=sum;
run;
ods pdf close;
```

Example 7 Using the ODS Graphics Statement to Resize a Lattice Graphical Display

ODS AND GRAPH ENHANCEMENTS

A comprehensive effort was made in SAS 9.2 to completely integrate ODS with both the device-based Graphics and the ODS Graphics architectures, significantly improving the quality of our graphical output. One of the key enhancements was the addition of ODS style support. So what is an ODS style you might ask? ODS styles are powerful mechanisms to control the overall appearance of your output. A style definition controls the presentation characteristics such as colors, fonts, borders, and so on. ODS styles are a collection of directives that describe how portions of your output are to be presented. As of SAS 9.2, all tabular and graphical output share the same style definition, providing a consistent look and feel to your resulting document. If you are familiar with the Cascading Style Sheet (CSS) concept, an ODS style is very similar.

Now that we have established that all default colors, fonts, and symbols come from the active ODS style for tables and graphs, let’s examine some sample output using different styles, and see how to change the style associated with each output destination.

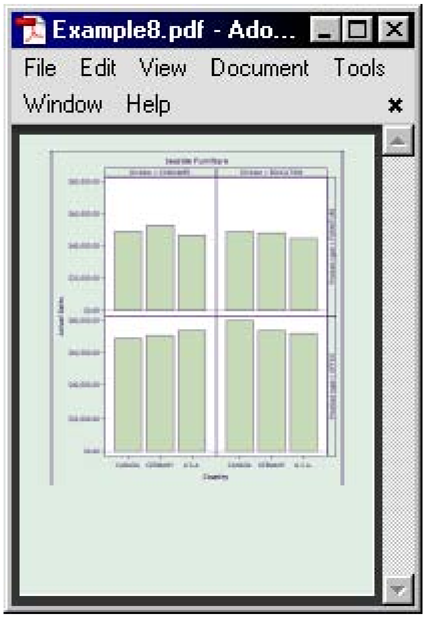


Figure Ocean style

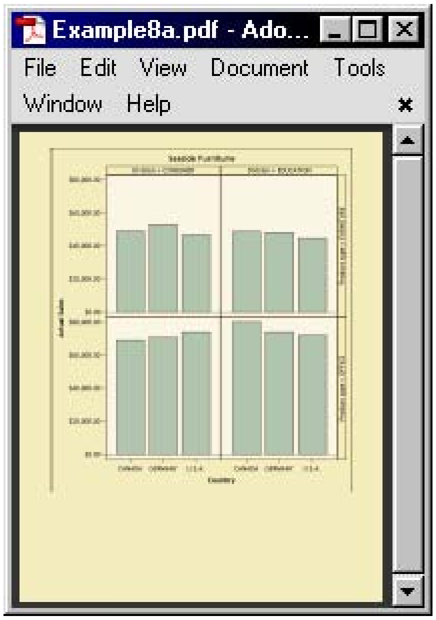


Figure Gears style

Examples 8a and 8b:

```
ods pdf file=”Example8.pdf
        style=Styles.Ocean;
```

```
ods pdf file=”Example8a.pdf”
        style=Styles.Gears;
```

Examples 8a and 8b Sample Output Using Different ODS Styles

DEFAULT STYLES

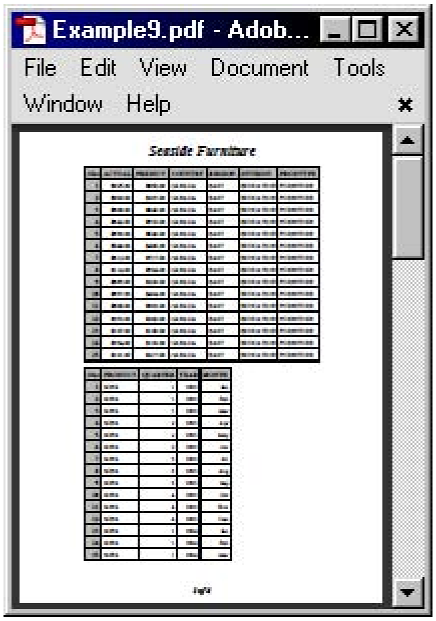
Whether you are preparing HTML for the Web, PDF files for printing, or RTF files for Microsoft Office integration, each third-party format has unique default expectations. Because output destinations serve different purposes, each has its own default style. ODS stores the default styles in a customer-modifiable hierarchical format called the SAS® Registry. For more information about how to alter the default style setting in the SAS Registry for a given output destination, please refer to the following paper “SAS® Graphics on ODS 9.2 Performance-Enhancing Steroids”(O’Connor 2008).

The following table lists the default styles for the most common output destinations.

Output Destination	Default Style
Listing	Listing (Graphics ONLY)
HTML	Default
RTF	RTF
Tagsets RTF	RTF
PDF	Printer
Postscript	Printer
PCL	Printer
Printer	Printer

MANAGING THE AVAILABLE SPACE

Thus far we have addressed how to control the paper size and what the default action is when a table or graph does not fit in the vertical direction, but what about the horizontal direction? What happens when the number of columns in your table does not fit within the available horizontal space? ODS does not have a natural ability to go to the next page as it does in the vertical direction. ODS has a table paneling technique that breaks your table up into panels of observations. This technique best displays the observations that would not fit, while repeating any table row and column heading information in order to supply each of these observations in a panel with the appropriate context. Let’s consider an example of how paneling can assist when the table is too wide for the page.



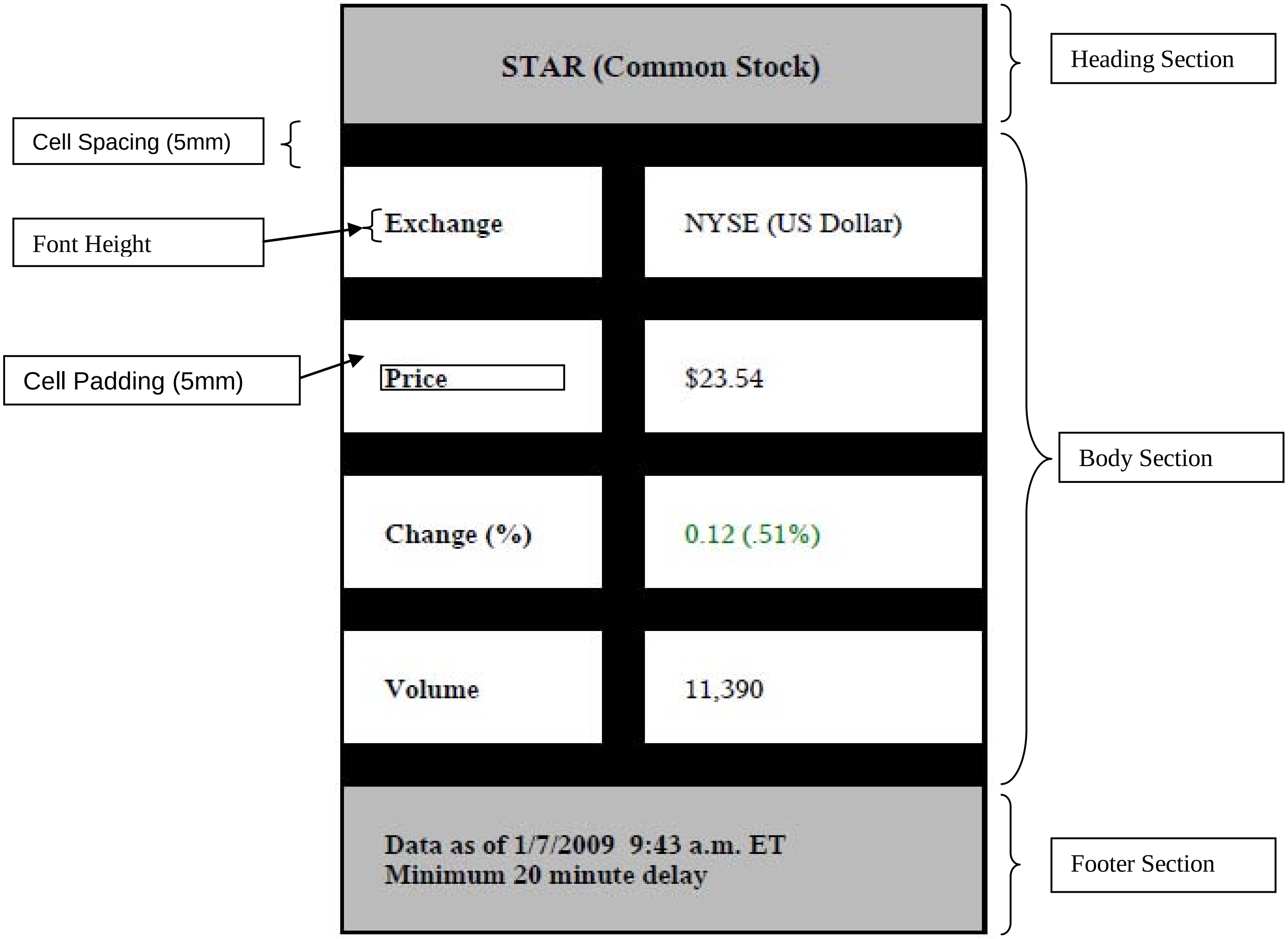
Seaside Furniture

Example 9 Paneled Output

Example 9:

```
Title "~{style [font_size=24pt]Seaside
Furniture}";
footnote"~{thispage} of ~{lastpage}";

ods pdf file="Example9.pdf" notoc;
proc print data=sashelp.prdsale
                (where=(country ='CANADA' and
                        region='EAST' and
                        division="EDUCATION" and
                        prodtype="FURNITURE"));
run;
ods pdf close;
```

One very common question is “how many rows (observations) can I fit on a single page”? The answer generally depends upon the size of the fonts, borders, cell padding, and cell spacing that is defined in the ODS style. Let’s quickly review the anatomy of a table to further understand how many observations will fit on a page.



STAR (Common Stock)	
Exchange	NYSE (US Dollar)
Price	$23.54
Change (%)	0.12 (.51%)
Volume	11,390
Data as of 1/7/2009 9:43 a.m. ET Minimum 20 minute delay	

Figure 2 Anatomy of a Table

As we can see in the preceding diagram it is more than just the font that can affect the number of observations or columns that can fit onto a page. It is the total accumulation of the font size for the data, the space inside the cells, the space between each cell, the font size used for any titles or footnotes, as well as the font size used for any heading or footing areas within a table itself. As we have learned, all of these setting come from the ODS style. The question is, how do you simultaneously scale all of these setting proportionally? It is actually quite simple conceptually; add a scale factor to the ODS style for every style attribute that affects size. To scale items in the ODS style, utilize the SAS SYSEVALF macro function, which evaluates an arithmetic expression using floating-point arithmetic. For convenience purposes I am providing the following small macro that will be used in our style modification code to implement our scaling.

Example 10:

```
%macro scale(size);
%if %symexist(factor) ne 1 %then
    %let factor = 1;
%let scaled=%sysevalf(&size * &factor);
&scaled.
%mend;
```

Example 10 Size-Scaling Macro

ODS styles are simply customizable SAS programs that are shipped with Base SAS®. SAS ships over fifty different ODS styles that are generally considered sample styles to show how various effects can be created. Use the PROC TEMPLATE procedure to define and modify ODS styles. Styles are organized by style elements with bundles of attributes, and these attributes apply to a particular section of the output being produced.

CONTROLLING TABLE ATTRIBUTES

The TABLE style element has a list of attributes that apply to all tables that are generated from SAS procedures. Let’s examine the TABLE style element that is defined in the PRINTER style.

Example 11:

```
 style Table from Output /
   rules = ALL
   cellpadding = 4pt
   cellspacing = 0.25pt
   borderwidth = 0.75pt;
```

Example 11 Table Style Element

In the preceding code sample we have four style attributes defined.

Style Attribute	***Description***
Rules	Controls the separator lines between rows and columns
Cellpadding	Specifies the amount of white space on each four sides of the text in a cell
Cellspacing	Specifies the thickness of the spacing between cells
Borderwidth	Specifies the width of the border of the table

SCALING TABLE ATTRIBUTES

Incorporating our new macro to allow the TABLE style elements to scale proportionally is as simple as doing the following.

Example 12:

```
style Table from Output /
  rules = ALL
  cellpadding = %scale(4)pt
  cellspacing = %scale(0.25)pt
  borderwidth = %scale(0.75)pt;
```

Example 12 Scaling the Table Style Element

CONTROLLING FONTS

Fonts are defined in two style element categories, ‘graph fonts’ and ‘non-graph fonts’. Let’s examine all the font definitions for the PRINTER style.

Example 13:

```
style fonts /
   'TitleFont2' = ("<MTserif>, Times Roman",12pt,bold italic)
   'TitleFont' = ("<MTserif>, Times Roman",13pt,bold italic)
   'StrongFont' = ("<MTserif>, Times Roman",10pt,bold)
   'EmphasisFont' = ("<MTserif>, Times Roman",10pt,italic)
   'FixedEmphasisFont' = ("<MTmonospace>, Courier",9pt,italic)
   'FixedStrongFont' = ("<MTmonospace>, Courier",9pt,bold)
   'FixedHeadingFont' = ("<MTmonospace>, Courier",9pt,bold)
   'BatchFixedFont' = ("SAS Monospace, <MTmonospace>, Courier",6.7pt)
   'FixedFont' = ("<MTmonospace>, Courier",9pt)
   'headingEmphasisFont' = ("<MTserif>, Times Roman",11pt,bold italic)
   'headingFont' = ("<MTserif>, Times Roman",11pt,bold)
   'docFont' = ("<MTserif>, Times Roman",10pt);
style GraphFonts /
   'GraphDataFont' = ("<MTserif>, <serif>",7pt)
   'GraphUnicodeFont' = ("<MTserif-unicode>",9pt)
   'GraphValueFont' = ("<MTserif>, <serif>",9pt)
   'GraphLabelFont' = ("<MTserif>, <serif>",10pt)
   'GraphFootnoteFont' = ("<MTserif>, <serif>",10pt)
   'GraphTitleFont' = ("<MTserif>, <serif>",11pt,bold)
   'GraphAnnoFont' = ("<MTserif>, <serif>",10pt);
```

Example 13 Table and Graph Style Definitions

Let's examine the 'docfont' style attribute font definition. Unlike the previous style attribute definitions, this one is a compound style attribute definition.

```
        'docFont' = ("<MTserif>, Times Roman",10pt)
```

The value on the left side of the '=' sign remains the style attribute name, however the font definition on the right side consists of a list of related font attributes.

("font-face-1, <… , font-face-n>", font-size, font-keywords)

"font-face" specifies the name of the font. ODS can now use any True Type font. The purpose of the list of font names is to provide flexibility and give the system an opportunity to use alternative font if the first one is not installed on the system. You will find that fonts can be operating system or even machine-specific. However, beginning in SAS, SAS 9.2 provides a redistributable font library that is available on all hosts to ensure great quality output.

<MTSerif> and <serif> aliases are dynamic registry references to entries located at [ODS\FONTS], which allow SAS to handle multiple locales without having to modify the production ODS style definitions. These "font-face" values could easily have been the literal font names.

"font-size" specifies the size of the font. Font-size is a nonnegative number optionally followed by a unit of measure like Point (PT), Percent (PCT), Inch (IN), or Millimeter (MM). For more information about font units, see the dimension unit of measure in the SAS Documentation.

"font-keywords" specifies the font weight, font style, and font width.

Font Weight	*Font Style*	*Font Width*
Medium	Italic	Normal
Bold	Roman	Compressed
Light	Slant	

So what does the 'docFont' style attribute control? In styles supplied by SAS, fonts are referenced a little differently than most style attributes. Let's examine the DATA style element that controls the data cells of a table.

Example 14:

```
style Data /
   font = Fonts(‘docFont’);
```

Example 14 Setting the font size for data area in a table

Notice that the “font” style attribute references our FONTS style element, and references the ‘docFont’ item that contains the composite font definition. All font references in a style are done indirectly into either the ‘fonts’ or the ‘GraphFonts’ style elements, which allows you to centralize font definitions.

SCALING FONTS

Incorporating the new macro into your ‘fonts’ and ‘GraphFonts’ style element definitions globally affects all subsequent font references. The indirect style element syntax shown in the previous example allows a simple incorporation.

Example 15:

```
style fonts /
    'TitleFont2' = ("<MTserif>, Times Roman",%scale(12)pt,bold italic)
    'TitleFont' = ("<MTserif>, Times Roman",%scale(13)pt,bold italic)
    'StrongFont' = ("<MTserif>, Times Roman",%scale(10)pt,bold)
    'EmphasisFont' = ("<MTserif>, Times Roman",%scale(10)pt,italic)
    'FixedEmphasisFont' = ("<MTmonospace>, Courier",%scale(9)pt,italic)
    'FixedStrongFont' = ("<MTmonospace>, Courier",%scale(9)pt,bold)
    'FixedHeadingFont' = ("<MTmonospace>, Courier",%scale(9)pt,bold)
    'BatchFixedFont' = ("SAS Monospace, <MTmonospace>, Courier",%scale(6.7)pt)
    'FixedFont' = ("<MTmonospace>, Courier",%scale(9)pt)
    'headingEmphasisFont' = ("<MTserif>, Times Roman",%scale(11)pt,bold italic)
    'headingFont' = ("<MTserif>, Times Roman",%scale(11)pt,bold)
    'docFont' = ("<MTserif>, Times Roman",%scale(10)pt);
style GraphFonts /
   'GraphDataFont' = ("<MTserif>, <serif>",%scale(7)pt)
   'GraphUnicodeFont' = ("<MTserif-unicode>",%scale(9)pt)
   'GraphValueFont' = ("<MTserif>, <serif>",%scale(9)pt)
   'GraphLabelFont' = ("<MTserif>, <serif>",%scale(10)pt)
   'GraphFootnoteFont' = ("<MTserif>, <serif>",%scale(10)pt)
   'GraphTitleFont' = ("<MTserif>, <serif>",%scale(11)pt,bold)
   'GraphAnnoFont' = ("<MTserif>, <serif>",%scale(10)pt);
```

Example 15 Scaling All Fonts in the PRINTER Style

GRAPHIC ATTRIBUTES

In general, most graph style elements need very little scaling. Starting in SAS 9.2 both graphic architectures are optimized to scale dynamically based on the overall graph dimensions. Notable exceptions are the border and spacing style attributes that globally affect the graphs as a whole.

Example 16:

```
style Graph from Graph /
  borderwidth = 0.75pt
  cellspacing = 0.25pt;
```

Example 16 Graph Style Element

SCALING GRAPHIC ATTRIBUTES

Incorporating your new macro to allow the GRAPH style element to scale proportionally is as simple as the following.

Example 17:

```
style Graph from Graph /
  borderwidth = %scale(0.75)pt
  cellspacing = %scale(0.25)pt;
```

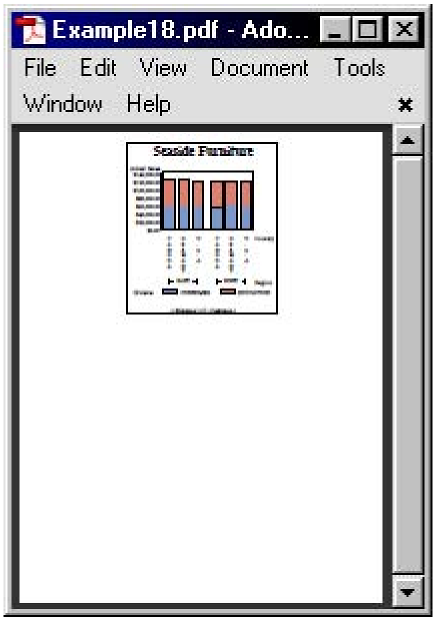
Example 17 Scaling the Graph Style Element

I have accumulated all of the previous style element changes for the ODS PRINTER style into a macro called “styles”, which is available in the Appendix. The %styles macro will be used in most of the remaining examples.

SCALING GRAPHIC OPTIONS

You are not limited to using the “scale” macro in just the ODS styles. It can be a very useful tool when providing sizing parameters to either of the two graphical architectures. Here is a simple example.

DEVICE-BASED GRAPHICS

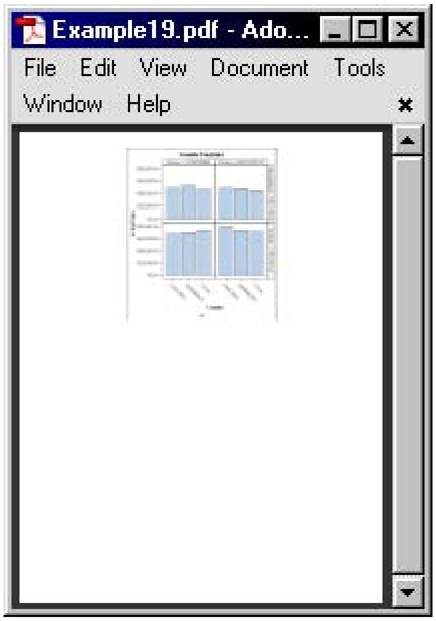


Example 18:

```
%let factor=.5;
%styles;

title "Seaside Furniture";
ods pdf file="Example18.pdf" notoc;

goptions hsize=%scale(7)in vsize=%scale(8)in
border;
proc gchart data=sashelp.prdsale;
vbar country / group=region subgroup=division
sumvar=actual;
run;

ods pdf close;
```

Example 18 Scaling the GOPTION Size Setting

ODS GRAPHICS



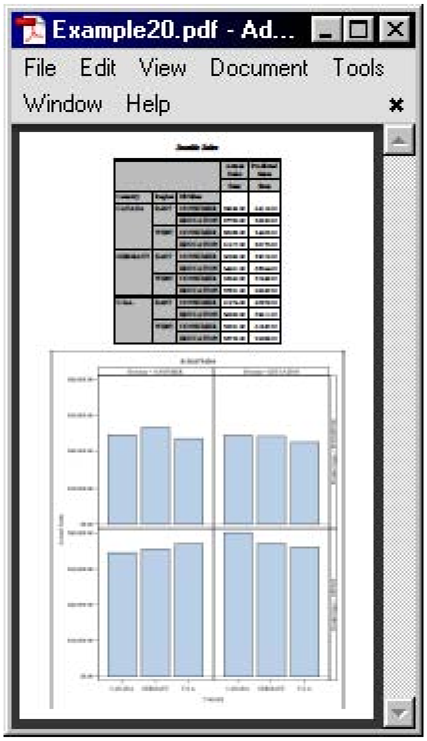
Example 19:

```
%let factor=.5;
%styles;

title "Seaside Furniture";
ods pdf file="Example19.pdf" notoc;

ods graphics / width=%scale(7)in
               height=%scale(8)in;
proc sgpanel data=sashelp.prdsale;
panelby division prodtype / layout=lattice;
vbar country / response=actual stat=sum;
run;

ods pdf close;
```

Example 19 Scaling the ODS Graphics Size Setting

SCALING TO THE PAPER SIZE

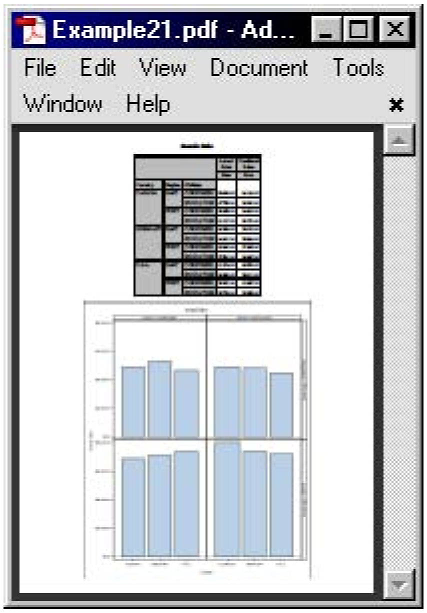
Consider the scenario where you have developed a report for a standard legal sized paper, and you prepare to print your report only to find out that you are out of legal-sized paper. What do you do? You need a hard copy of this report now! Using this simple little macro you can scale your content to use the identical number of pages on letter-sized paper by just applying the correct scale factor at the top of the job. When calculating your scale factor be sure to use the page size minus the margin settings. Assuming that you have .25in top and bottom borders for both the legal- and letter-size paper, you would compute your scale factor in the following manner: 10.5in/13.5in = .777. Change your scale factor and PAPERSIZE option, rerun the report, and you are done.



Example 20:

```
options papersize=legal;
ods pdf file="Example20.pdf" notoc startpage=off;

title "Seaside Sales";
proc tabulate data=sashelp.prdsale;
class country region division;
variable actual predict;
table country*region*division,
(actual*f=dollar8.2)*sum
(predict*f=dollar8.2)*sum;
;run;

title "Actual Sales";
ods graphics / width=7in height=8.5in;
proc sgpanel data=sashelp.prdsale;
panelby division prodtype/ layout=lattice;
vbar country / response=actual stat=sum;
run;

ods pdf close;
```

Example 20 Output on Legal Size Paper

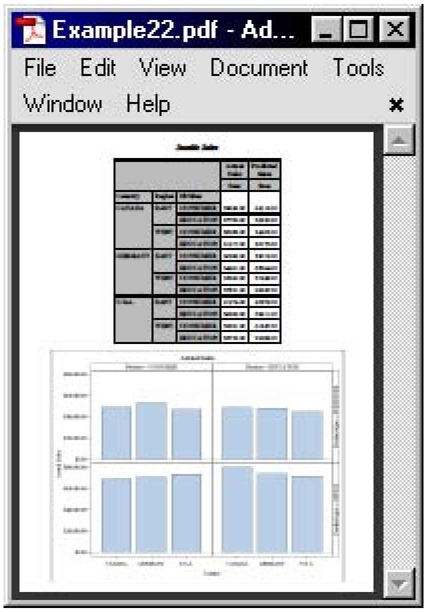


Example 21:

```
options papersize=letter;
%let factor=.777;
%styles;

ods pdf file="Example21.pdf" notoc startpage=off
style=Styles.ScaledPrinter;

title "Seaside Sales";
proc tabulate data=prdsale;
class country region division;
variable actual predict;
table country*region*division,
(actual*f=dollar8.2)*sum (predict*f=dollar8.2)*sum;
;run;

title "Actual Sales";
ods graphics / width=%scale(7)in
height=%scale(8.5)in;
proc sgpanel data=prdsale;
panelby division prodtype/ layout=lattice;
vbar country / response=actual stat=sum;
run;

ods pdf close;
```

Example 21 Legal Size Paper Scaled Down to Letter Size Paper

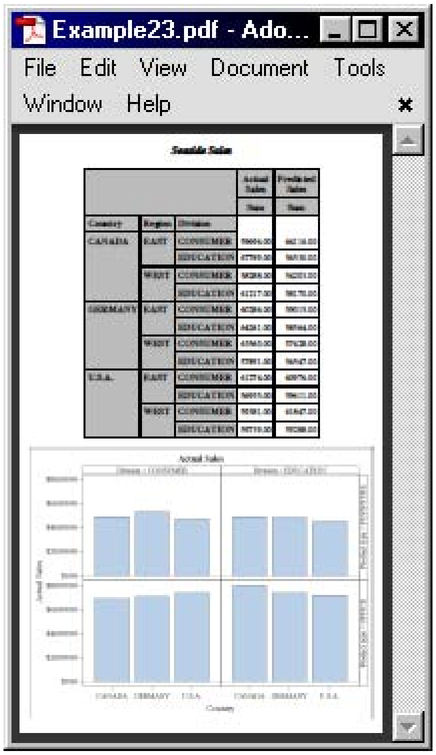
Conversely, you may want to retain the same number of pages of output and scale your report from letter- to legal-sized paper. Given the same margin assumption as in the previous example, you would compute your scale factor in the following manner: 13.5in/10.5in=1.28. Change your scale factor and PAPERSIZE option accordingly and you have scaled your report up to a larger paper size. Due to rounding issues when producing the report you will find that your scale factor might need to be adjusted by a small amount.



Example 22:

```
ods pdf file="Example22.pdf" notoc startpage=off;

title "Seaside Sales";
proc tabulate data=prdsale;
class country region division;
variable actual predict;
table country*region*division,
(actual*f=dollar8.2)*sum (predict*f=dollar8.2)*sum;
;run;

title "Actual Sales";
ods graphics / width=7in height=5.5in;
proc sgpanel data=prdsale;
panelby division prodtype/ layout=lattice;
vbar country / response=actual stat=sum;
run;

ods pdf close;
```

Example 22 Output on Letter Size Paper



Example 23:

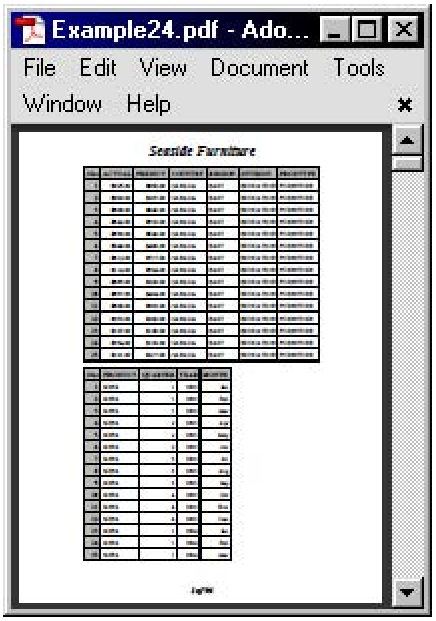
```
options papersize=legal;
%let factor=1.4;
%styles;

ods pdf file="Example23.pdf" notoc startpage=off
style=Styles.ScaledPrinter;

title "Seaside Sales";
proc tabulate data=prdsale;
class country region division;
variable actual predict;
table country*region*division,
(actual*f=dollar8.2)*sum (predict*f=dollar8.2)*sum;
;run;

title "Actual Sales";
ods graphics / width=%scale(7)in
               height=%scale(5.5)in;
proc sgpanel data=prdsale;
panelby division prodtype/ layout=lattice;
vbar country / response=actual stat=sum;
run;

ods pdf close;
```

Example 23 Letter Size Paper Scaled to Legal

The important distinction here is that you are not performing the photo copier “Reduce” and “Enlarge” approach. You are actually scaling the fonts, white space, and graphics proportionally to the new size while also optimally selecting all the best graphical markers, legend, axis attributes to ensure a professional graphic output appearance.

SCALING TO FIT MORE CONTENT ON A PAGE

Scaling your report to allow more observations or columns on the page might be a more powerful motivation for using this scaling solution. Let’s examine a report that just needs a little more horizontal space to keep all the tables from paneling. This report has 10 columns of information, but the final column did not fit in your letter-sized paper in a portrait orientation, and caused the entire report to panel and ultimately produce 96 pages of output. Not only does this make the report more difficult to read, but it produces twice as many pages and all because the last column did not fit.

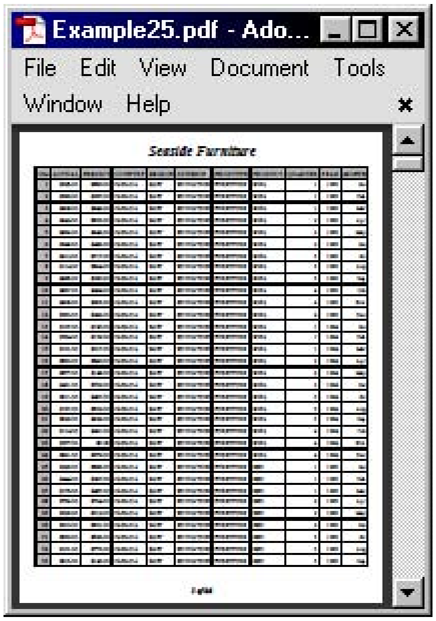


Example 24:

```
Title "~{style [font_size=24pt]Seaside
Furniture}";
footnote"~{thispage} of ~{lastpage}";

ods pdf file="Example24.pdf" notoc;
proc print data=sashelp.prdsale;run;
ods pdf close;
```

Example 24 Paneled Output That Almost Fit (Page 1 of 96)

Reducing the style characteristics by 6.5% allows all of the columns to fit horizontally in your letter paper size, reducing the number of pages produced in the report to only 44 pages. A very small adjustment here has a significant effect on readability and length of your report.



Example 25:

```
Title "~{style [font_size=24pt]Seaside
furniture}";
footnote"~{thispage} of ~{lastpage}";

%let factor=.935;  /* Reducing by 6.5% */
%styles;

ods pdf file="Example25.pdf" notoc
style=Styles.ScaledPrinter;
proc print data=sashelp.prdsale;run;
ods pdf close;
```

Example 25 Reduce the Output by 6.5% to Avoid Paneling (Page 1 of 44)

CONCLUSION

Creating sophisticated reports that convey your information in a clear and concise manner can be challenging. This paper has detailed the great efforts made by SAS to address the challenges of limited physical real-estate in printed reports. ODS styles provide a very powerful mechanism for centralizing default characteristics, and can easily be customized to dramatically alter your report's look and feel. Graphics have been enhanced to automatically scale all components, such as axis information, legends, and symbols, to easily produce professional looking graphics. It is also very easy to scale your reports to address various reporting challenges.

ACKNOWLEDGMENTS

The author would like to thank Scott Huntley, Chevell Parker, Bari Lawhorn, David Kelley, Kevin Smith, and Allison Crutchfield for their contributions to this paper.

REFERENCES

SAS Institute Inc. 2009. SAS/GRAPH 9.2: Reference. Cary, NC: SAS Institute Inc. Available http://support.sas.com/documentation/onlinedoc/graph/index.html.SAS Institute Inc. 2009. SAS/GRAPH Statistical Graphics Procedures Guide. Cary, NC: SAS Institute Inc. http://support.sas.com/documentation/onlinedoc/graph/index.html.

O'Connor, Daniel. 2008. "SAS® Graphics on ODS 9.2 Performance-Enhancing Steroids." *Proceedings of the SAS Global Forum Conference*. Cary, NC: SAS Institute Inc. Available at www2.sas.com/proceedings/forum2008/254-2008.pdf.

CONTACT INFORMATION

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

Daniel O'Connor
Building R, SAS Campus Drive
SAS Institute Inc.
Cary, NC 27513
E-mail: Dan.OConnor@sas.com

APPENDIX

SCALE MACRO

```
%macro scale(size);
%if %symexist(factor) ne 1 %then
    %let factor = 1;
%let scaled=%sysevalf(&size * &factor);
&scaled.
%mend;
```

SCALE PRINTER STYLE MACRO

```
%macro styles();
proc template;
define style Styles.ScaledPrinter;
parent = Styles.Printer;
      style fonts /
         'TitleFont2' = ("<MTserif>, Times Roman",%scale(12)pt,bold italic)
         'TitleFont' = ("<MTserif>, Times Roman",%scale(13)pt,bold italic)
         'StrongFont' = ("<MTserif>, Times Roman",%scale(10)pt,bold)
         'EmphasisFont' = ("<MTserif>, Times Roman",%scale(10)pt,italic)
         'FixedEmphasisFont' = ("<MTmonospace>, Courier",%scale(9)pt,italic)
         'FixedStrongFont' = ("<MTmonospace>, Courier",%scale(9)pt,bold)
         'FixedHeadingFont' = ("<MTmonospace>, Courier",%scale(9)pt,bold)
         'BatchFixedFont' = ("SAS Monospace, <MTmonospace>, Courier",%scale(6.7)pt)
         'FixedFont' = ("<MTmonospace>, Courier",%scale(9)pt)
         'headingEmphasisFont' = ("<MTserif>, Times Roman",%scale(11)pt,bold italic)
         'headingFont' = ("<MTserif>, Times Roman",%scale(11)pt,bold)
         'docFont' = ("<MTserif>, Times Roman",%scale(10)pt);
      style GraphFonts /
         'GraphDataFont' = ("<MTserif>, <serif>",%scale(7)pt)
         'GraphUnicodeFont' = ("<MTserif-unicode>",%scale(9)pt)
         'GraphValueFont' = ("<MTserif>, <serif>",%scale(9)pt)
         'GraphLabelFont' = ("<MTserif>, <serif>",%scale(10)pt)
         'GraphFootnoteFont' = ("<MTserif>, <serif>",%scale(10)pt)
         'GraphTitleFont' = ("<MTserif>, <serif>",%scale(11)pt,bold)
         'GraphAnnoFont' = ("<MTserif>, <serif>",%scale(10)pt);
      style Table from Output /
         bordercollapse = separate
         rules = ALL
         padding = %scale(4)pt
         borderspacing = %scale(0.25)pt
         borderwidth = %scale(0.75)pt;
      style Graph from Graph /
         borderwidth = %scale(0.75)pt
         borderspacing = %scale(0.25)pt;
      style data /
         font = Fonts('docFont');
 end;
run;
%mend;
```