

Paper 099-2007

Funny ^Stuff~ in My Code: Using ODS ESCAPECHAR

Cynthia L. Zender, SAS Institute Inc., Cary, NC

ABSTRACT

Have you ever wanted to insert a superscript or subscript into your SAS[®] output? Or, have you ever needed to make one part of a text string bold or a different color? Do you wonder how to get RTF or PDF output to have Page X of Y page numbering in a footnote?

This paper introduces you to the wonders of the ODS ESCAPECHAR statement (ODS ESCAPECHAR=). Discussed is how to define and then use a specific escape character to perform in-line formatting. Concrete examples show you how to change style within one table cell (or a SAS title or footnote), how to insert Page X of Y page numbers into your output, how to insert line feeds into long text strings, and how to insert destination-specific "raw text" into your HTML or RTF output. Learn how to use ODS ESCAPECHAR= and ESCAPECHAR control strings effectively and make the funny characters work for you.

A short glossary of the terms used in this paper appears at the end of the paper.

INTRODUCTION

The ability to insert control characters into your text strings and variable values was first introduced in SAS[®] 8.2. Not much has changed in newer versions of SAS, except that more functionality has been added to what ODS ESCAPECHAR= can do in your Output Delivery System (ODS) output. The fundamental explanation for how ODS ESCAPECHAR= works might be easier to understand if you remember the "old" days of DOS word processing programs, like WordStar or WordPerfect. In these early days of word processing, computer screens could not show fonts, colors, bold face, or italics. Documents were displayed in plain text on the screen with control characters added that indicated what kind of formatting should be used to print the document.

Frequently, these control characters were inserted into the document with the ESC key or the CTRL key. So, for example, ESC + B might mean "use a bold typeface for whatever character or text follows until a 'turn off formatting' control sequence or string is encountered." Microsoft Word changed the world of word processing with the introduction of WYSIWYG (What You See Is What You Get) word processing software. The document that you created was displayed on the screen in the font that you wanted and when you applied bold formatting to a word, the word appeared in **BOLD** type and you no longer saw the underlying control characters or escape characters.

SAS code has a lot in common with DOS word processors. Your SAS data does not have any inherent font or color or superscript, when stored as data. Therefore, when you produce reports based on your data, any "extra" information, such as superscripts, subscripts, bolding, or italicizing must be added programmatically. Additionally, of course, there are no styles or typographic niceties available in the SAS Output window or LISTING destination. In the world of ODS output, only destinations that support style characteristics will allow you to present your output with bolding or italicizing or font changes in the final ODS result file.

Consider the following example sentence: "The **quick** brown fox jumped over the lazy red dog." Most of the sentence is in the Arial font, except for the words "fox" and "dog", which are displayed in the Courier New font. The word "quick" is in boldface; the word "jumped" is in italics. The words "brown" and "red" are displayed in their respective colors. So, in the line of text, inside Microsoft Word, some words are formatted differently from the rest of the line. Within SAS code, you would have to use ODS ESCAPECHAR= control strings to achieve this same kind of "in-line" formatting in the final output file.

SPECIFYING AN ESCAPE CHARACTER FOR ODS

Before you can apply or perform in-line formatting with ODS, however, you must first tell ODS what character will be used as the ODS ESCAPECHAR character value (ESCAPECHAR). There are two default escape character sequences: (*ESC*) and '03'x. The default (*ESC*) was new in SAS[®]9, while '03'x was available starting in SAS 8.2; however, both of those ESCAPECHAR values have too many keystrokes for me. I prefer setting a single character as my ESCAPECHAR, such as:

```
ods escapechar='^';  
ods escapechar='~';  
ods escapechar='#';
```

If I used the first statement in my code, this means that anytime ODS encounters the caret (^) it will interpret whatever follows the caret (^) to be a sequence or string of control characters that will have an impact on the final output file, as created by ODS. Of course, you should avoid characters that might have meaning in your data or might be used for different purposes. For example, in general I avoid using the back slash (\), forward slash (/), ampersand (&), semicolon(;), colon (:), less than (<), greater than (>), and common punctuation marks (such as comma, question mark, exclamation point, dollar sign, percent sign, and so on) for use as the ODS ESCAPECHAR character value (ESCAPECHAR).

SETTING AND USING ODS ESCAPECHAR= TO CHANGE STYLE ATTRIBUTES

Consider the following program. INLINE1.SAS. It is a DATA step program that creates five observations and just one variable in the WORK.USE_ESC data set. The variable, var1, is a text string that has different ESCAPECHAR control strings specified in observations 2 through 5. The program is shown first, followed by the HTML output that it produces (Figure 1). The use of the ODS ESCAPECHAR character (^) is highlighted for each observation. The ODS invocation is self-explanatory, except for the use of the NOTOC_DATA option which is an experimental ODS RTF option. The NOTOC_DATA option suppresses any hidden RTF control strings which are put into the table headers just in case you want to build a table of contents in Word. This option was introduced in SAS 9.1.3, Service Pack 4.

PROGRAM: INLINE1.SAS

```
data use_esc;
  length var1 $200;
  var1 = "The quick brown fox";
  output;
  var1 = "The ^S={font_weight=bold}quick ^S={} brown fox";
  output;
  var1 = "The quick ^S={foreground=brown}brown ^S={} fox";
  output;
  var1 = "The quick brown ^S={font_face='Courier New'}fox ^S={}";
  output;
  var1 = "The ^S={font_weight=bold}quick ^S={} " ||
        "^S={foreground=brown}brown ^S={} " ||
        "^S={font_face='Courier New'}fox ^S={} ";
  output;
run;

ods html file='inlined.html' style=sasweb;
ods rtf file='inlined.rtf' notoc_data;
ods pdf file='inlined.pdf';
ods escapechar='^';

proc print data=use_esc;
run;

ods _all_ close;
```

In Figure 1, you can see that Observation 1 does not have any special formatting within the var1 cell. However, for Observation 2, the word "quick" is in bold; for Observation 3, the word "brown" is brown; for Observation 4, the word "fox" is in the Courier New font; and for Observation 5, all of those changes have been combined together and only the word "The" remains unchanged in the data cell.

Obs	var1
1	The quick brown fox
2	The quick brown fox
3	The quick brown fox
4	The quick brown fox
5	The quick brown fox

Figure 1: Output from Program INLINE1.SAS

This example illustrates three important things about using ODS ESCAPECHAR= to change style characteristics:

1. The escape character that you declare in the ODS ESCAPECHAR= statement must be the same one you use in your code.
2. One of the things that you can do with the ODS ESCAPECHAR control string is to specify style characteristics using the syntax `ESCAPECHAR + S={attribute=value}`.
3. If you are making style changes, a null style specification (`ESCAPECHAR + S={}`) terminates the "halo" effect of the style attribute override.

This was just one example of performing in-line formatting with ODS ESCAPECHAR=. Let's move on to more of the basics.

ODS ESCAPECHAR= BASICS

ODS ESCAPECHAR functionality can be used four ways: Style Attribute insertion, Function insertion, line break and other character insertion, and Raw Text insertion. The first example above illustrated Style Attribute insertion. These four capabilities are described in the table below. None of these capabilities will be available to you, however, unless you first declare what your ESCAPECHAR will be. In the table below, you can assume that the ESCAPECHAR has been defined as the caret (^).

Category	Description and Usage (Assume ODS ESCAPECHAR='^'; in effect)
Style Attribute insertion	<code>^S={attribute1=value1 attribute2=value2 attributen=valuen}</code> will set part of the value in a current cell (or the whole value of a cell) until null style specification <code>^S={}</code> is encountered.
Function insertion	Uses predefined functions to insert special characters (like the dagger or superscripts and subscripts or Page X of Y) into a cell. <code>^{super 2}</code> would cause a superscripted numeral 2 to appear in the output. <code>^{pageof}</code> inserts the current page and last page into an RTF document title or footnote. <code>^{dagger}</code> inserts a dagger (†) character into a cell.
Line break, line wrap, non-breaking space, and other special codes	Inserts generally unprintable characters into the cell <u>or</u> inserts characters which may be "unformatted" in the code, but will have the desired formatting in the output file. But the characters perform a specified function (breaking a line (<code>^n</code>), marking a place to wrap the line (<code>^m</code>), inserting a non-breaking space (<code>^_</code>), and so on) in the output file.
Raw Text insertion (only for HTML and RTF)	If you know HTML or RTF control strings, then you can use the <code>^R</code> capability to insert "native" control strings into the output file. When you use the raw text capability, you are passing the control string directly into the output file for rendering; therefore, this capability should be used by those who are very familiar with ODS and very familiar with the control language of either HTML or RTF.

More style attribute examples are provided at the end of this paper.

Remember that if the ODS ESCAPECHAR= value had been declared as a tilde (~), then all the above examples would need to use the tilde as the ESCAPECHAR: for example, ~{super 2}, ~{pageof}, or ~{dagger}. The next section covers the use of ODS ESCAPECHAR= Functions.

SETTING ODS ESCAPECHAR= AND USING ESCAPECHAR FUNCTIONS

The ability to superscript and subscript numbers or strings or formulas is a very common report requirement. Another requirement is the ability to indicate a current and final page number on every page of a document. This page numbering scheme is known as "Page X of Y" page numbering. ODS has an ESCAPECHAR function to insert these kinds of page numbers into RTF and PDF documents. Sadly, the paperless world of HTML documents does not use these ODS ESCAPECHAR functions for page numbering. Another function is the dagger function, which inserts the dagger (†) character into your output document. These functions are illustrated in the next example.

PROGRAM: INLINE2.SAS

```
data sup_sub;
  length myvar $200;
  myvar = "Pythagorean Theorem: a^{super 2} + b^{super 2} = c^{super 2}";
  output;
  myvar = "This is something that needs a footnote. ^{super 1}";
  output;
  myvar = "Macbeth: 'Is this a dagger I see before me?' ^{dagger}";
  output;
  myvar = "The Caffeine molecule is an alkaloid of the methylxanthine family: " ||
    "C^{sub 8}H^{sub 10}N^{sub 4}O^{sub 2}";
  output;
run;

ods html file='inline2.html' style=sasweb;
ods rtf file='inline2.rtf' notoc_data;
ods pdf file='inline2.pdf';
ods escapechar='^';

proc print data=sup_sub;
  title j=r 'PDF & RTF: Page ^{thispage} of ^{lastpage}';
  title2 j=c 'RTF only: ^{pageof}';
  footnote ^{super 1}If this were a real footnote, there would be something very
academic here.';
  footnote2 ^{dagger} Macbeth talked to himself a lot. This quote is from Macbeth:
Act 2, Scene 1, Lines 33-39.';
run;
ods _all_ close;
```

The output from the program INLINE2.SAS is shown in Figures 2 through 6. Because the output from HTML, RTF, and PDF differs, all three outputs are shown. However, only the footnotes from HTML and PDF are shown in order to conserve space. The RTF footnote is the same as the PDF footnote and appears at the bottom of the output document when opened in Microsoft Word.

Obs	myvar
1	Pythagorean Theorem: a ² + b ² = c ²
2	This is something that needs a footnote. ¹
3	Macbeth: 'Is this a dagger I see before me?' †

Figure 2: Partial LISTING Output from Program INLINE2.SAS

Notice in Figure 2 that the ESCAPECHAR string appears in the LISTING window output because the string was actually inserted into the data but the LISTING window cannot render these strings.

PDF & RTF: Page of

RTF only:

Obs	myvar
1	Pythagorean Theorem: $a^2 + b^2 = c^2$
2	This is something that needs a footnote. ¹
3	Macbeth: 'Is this a dagger I see before me?' †
4	The Caffeine molecule is an alkaloid of the methylxanthine family: $C_8H_{10}N_4O_2$

¹If this were a real footnote, there would be something very academic here.
 † Macbeth talked to himself a lot. This quote is from Macbeth: Act 2, Scene 1, Lines 33-39.

Figure 3: HTML Output from Program INLINE2.SAS

Notice in Figure 3 that the page numbering functions are not used at all in the HTML file.

PDF & RTF: Page 1 of 1

RTF only: 1 of 1

Obs	myvar
1	Pythagorean Theorem: $a^2 + b^2 = c^2$
2	This is something that needs a footnote. ¹
3	Macbeth: 'Is this a dagger I see before me?' †
4	The Caffeine molecule is an alkaloid of the methylxanthine family: $C_8H_{10}N_4O_2$

Figure 4: RTF Output (Shown in Print Preview Mode) from Program INLINE2.SAS

Notice in Figure 4 that all of the page numbering functions are used by the RTF destination.

PDF & RTF: Page 1 of 1

RTF only: {pageof }

Obs	myvar
1	Pythagorean Theorem: $a^2 + b^2 = c^2$
2	This is something that needs a footnote. ¹
3	Macbeth: 'Is this a dagger I see before me?' †
4	The Caffeine molecule is an alkaloid of the methylxanthine family: $C_8H_{10}N_4O_2$

Figure 5: PDF Output from Program INLINE2.SAS

Notice in Figure 5 that only the `{thispage}` and `{lastpage}` functions are used by the Printer-Family PDF destination.

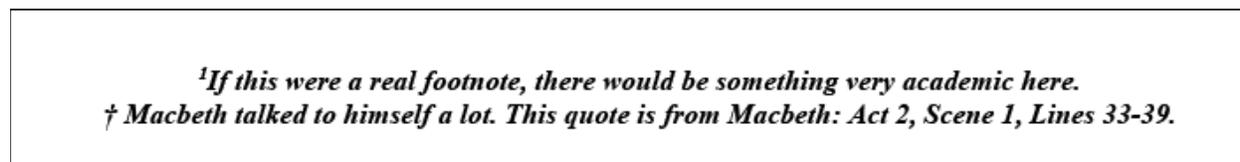


Figure 6: PDF File Footnote from Program INLINE2.SAS

As shown in Figure 6, the RTF document contains the footnote in the footer portion of the document, when opened with Microsoft Word.

The `{super}` or `{sub}` functions can be used to superscript or subscript more than one character. If, for example, you changed the above program to create another observation using this syntax:

```
myvar = "Look ^{sub UNDER} and ^{super OVER} everything.";
```

you would get the following output as shown in Figure 7:

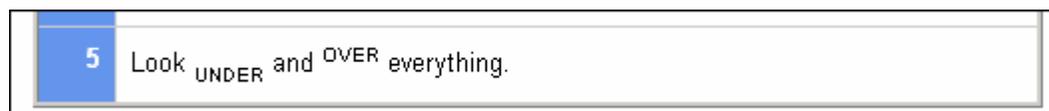


Figure 7: HTML Output from Changed Syntax Using `{super}` and `{sub}` on Whole Words

There is only one other function that is available at this time to discuss, and that is the ESCAPECHAR plus `{leaders <string>}` function. It is reserved for use with Printer-Family destinations only and its use is generally limited to style templates. The `{leaders}` function repeats the string character. For example, it would insert the row of dots or periods before a page number in a table of contents. The coverage of style templates is outside the scope of this paper; consult the ODS documentation "New ODS Printer Features for SAS 9.1" (SAS Institute Inc. 2007 (b)) for more information about the `{leaders}` function.

INSERTING LINE BREAKS AND NON-BREAKING SPACES WITH ODS ESCAPECHAR=

If you look at Figure 3, 4, or 5, you see that Observation 4 is one long string. What if you wanted to introduce a line break before the chemical formula for the caffeine molecule? In RTF control syntax, you might introduce the `\line` controls string. In HTML, you might introduce the `
` tag into the cell. Using ODS ESCAPECHAR control strings, it is possible to introduce one line break sequence or string that will be respected in all of these destinations.

Have you ever struggled with the limit of ten titles or ten footnotes? For most reports, ten titles or footnotes might suffice, but if they don't suffice, using ODS ESCAPECHAR= gives you a way around this limit. The key for both of these scenarios involves knowing that ODS ESCAPECHAR= gives you line breaking capability through the use of your chosen ESCAPECHAR + n. So, if the ESCAPECHAR is set to the tilde character (~), then ~n is one line break and ~2n is a double line feed (or carriage return for those who remember manual typewriters). Any positive integer between the ESCAPECHAR and the 'n' will result in that many line feeds or carriage returns. In the code below, ~n is a single line break; ~2n is 2 line breaks, ~3n is 3 line breaks; and ~4n is 4 line breaks. Figures 8, 9, and 10 show the output from the INLINE3.SAS program. Only the line break escape sequences are highlighted in the code:

PROGRAM: INLINE3.SAS

```

data linebr;
  length myvar $200;
  myvar ="The Caffeine molecule is an alkaloid of the methylxanthine family: ~n" ||
    " C~{sub 8}H~{sub 10}N~{sub 4}O~{sub 2}";
  output;
  myvar ="The Caffeine molecule is an alkaloid of the methylxanthine family: ~2n" ||
    " C~{sub 8}H~{sub 10}N~{sub 4}O~{sub 2}";
  output;
  myvar ="The Caffeine molecule is an alkaloid of the methylxanthine family: ~3n" ||
    " C~{sub 8}H~{sub 10}N~{sub 4}O~{sub 2}";
  output;
  myvar ="The Caffeine molecule is an alkaloid of the methylxanthine family: ~4n" ||
    " C~{sub 8}H~{sub 10}N~{sub 4}O~{sub 2}";
  output;
run;

ods listing;
ods html file='inline3.html' style=sasweb;
ods rtf file='inline3.rtf' notoc_data;
ods pdf file='inline3.pdf';
ods escapechar='~';

proc print data=linebr;
  title 'Using the ODS ESCAPECHAR for line break';
  title2 'Title 2';
  title3 'Title 3';
  title4 'Title 4';
  title5 'Title 5';
  title6 'Title 6';
  title7 'Title 7';
  title8 'Title 8';
  title9 'Title 9';
  title10 'Title 10 ~n Title 11 ~n Title 12 ~n Title 13 ~n Title 14';
run;

ods _all_ close;

```

In the last TITLE statement, the use of white space around the '~n' is for readability purposes only. The output for the LISTING window (Figure 8) shows that the ESCAPECHAR control string is not used in that destination. However, the output in the other destinations shows how the inserted line breaks look in the title area and in the table area.

Using the ODS ESCAPECHAR for line break	
	Title 2
	Title 3
	Title 4
	Title 5
	Title 6
	Title 7
	Title 8
	Title 9
	Title 10 ~n Title 11 ~n Title 12 ~n Title 13 ~n Title 14
Obs	myvar
1	The Caffeine molecule is an alkaloid of the methylxanthine family: ~nC~{sub 8}H~{sub 10}N~{sub 4}O~{sub 2}
2	The Caffeine molecule is an alkaloid of the methylxanthine family: ~2nC~{sub 8}H~{sub 10}N~{sub 4}O~{sub 2}
3	The Caffeine molecule is an alkaloid of the methylxanthine family: ~3nC~{sub 8}H~{sub 10}N~{sub 4}O~{sub 2}
4	The Caffeine molecule is an alkaloid of the methylxanthine family: ~4nC~{sub 8}H~{sub 10}N~{sub 4}O~{sub 2}

Figure 8: LISTING Destination Results from INLINE3.SAS

Using the ODS ESCAPECHAR for line break

Title 2
Title 3
Title 4
Title 5
Title 6
Title 7
Title 8
Title 9
Title 10
Title 11
Title 12
Title 13
Title 14

Obs	myvar
1	The Caffeine molecule is an alkaloid of the methylxanthine family: C ₈ H ₁₀ N ₄ O ₂
2	The Caffeine molecule is an alkaloid of the methylxanthine family: C ₈ H ₁₀ N ₄ O ₂
3	The Caffeine molecule is an alkaloid of the methylxanthine family: C ₈ H ₁₀ N ₄ O ₂
4	The Caffeine molecule is an alkaloid of the methylxanthine family: C ₈ H ₁₀ N ₄ O ₂

Figure 9: HTML Results from INLINE3.SAS

Using the ODS ESCAPECHAR for line break

Title 2
Title 3
Title 4
Title 5
Title 6
Title 7
Title 8
Title 9
Title 10
Title 11
Title 12
Title 13
Title 14

Obs	myvar
1	The Caffeine molecule is an alkaloid of the methylxanthine family: C ₈ H ₁₀ N ₄ O ₂
2	The Caffeine molecule is an alkaloid of the methylxanthine family: C ₈ H ₁₀ N ₄ O ₂
3	The Caffeine molecule is an alkaloid of the methylxanthine family: C ₈ H ₁₀ N ₄ O ₂
4	The Caffeine molecule is an alkaloid of the methylxanthine family: C ₈ H ₁₀ N ₄ O ₂

Figure 10: RTF Results from INLINE3.SAS (Shown in Print Preview Mode)

PDF results are not shown but essentially look the same as the RTF results.

Other features related to line break insertion are the ability to:

- insert special symbols, such as the copyright (©) symbol or the registered trademark (®) symbol (shown in INLINE5.SAS)
- insert strings such as NOTE, WARNING, ERROR, or FATAL
- insert a line break within a line and wrap or indent remaining text by using a "marker" ESCAPECHAR string

For more information about these features, consult ODS FAQs and Concepts (SAS Institute Inc. 2007(c)) or "Let the ODS PRINTER Statement Take Your Output into the Twenty-First Century" (Huntley 2006). In addition, the BONUS_LINEWRAP.SAS program (which demonstrates these features) is available for download with all the other programs used in this paper. See Zender (2007) in the Resources section.

INSERTING RAW TEXT USING ODS ESCAPECHAR CONTROL STRINGS

What if, for some reason, you wanted to insert control strings specific to RTF formatting or markup tags specific to HTML formatting? This is one of the reasons for using the Raw Text insertion capability of ODS ESCAPECHAR control strings. Another reason is that the ODS ESCAPECHAR control string does add bytes to the variable's length or the length of your TITLE or FOOTNOTE statement. When you insert Raw Text using the ESCAPECHARR+R method, your "raw" text string is passed directly to the selected destination—it does not get stored as part of the variable or title value, so the Raw Text string would not show in the LISTING destination and it would not add bytes to the variable value or title length.

Consider the following simple control strings for RTF. **\line** is the RTF-specific control string for a line break, while **\tab** is the RTF-specific control string for a tab character. Since these control characters do not exist in HTML, you must use HTML-specific markup tags to perform line breaks and the equivalent of tabbing. For HTML, the line break is accomplished through the use of the `
` tag. HTML does not have a TAB capability. The only thing that you can do to emulate a tab character is to insert white space between text strings by using ` `; which is the non-breaking space entity for HTML documents (note that ` ` is case sensitive and must be lowercase). In the source code the ESCAPECHAR has been changed to '#' for this example. The output is shown in Figure 11:

PROGRAM: INLINE4.SAS

```
data linebr_raw;
  length myvar $200;
  myvar = "The Caffeine molecule is an alkaloid of the methylxanthine family: " ||
    "#R/HTML'<BR/>' #R/RTF'\line' " ||
    " C#{sub 8}H#{sub 10}N#{sub 4}O#{sub 2}";
  output;
  myvar = "The Caffeine molecule is an alkaloid of the methylxanthine family: " ||
    "#R/HTML'<BR/><BR/>' #R/RTF'\line \line' " ||
    " C#{sub 8}H#{sub 10}N#{sub 4}O#{sub 2}";
  output;
  myvar = "Start of Text" ||
    "#R/HTML'&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;' #R/RTF'\tab' " ||
    "Rest of Text";
  output;
run;

options nodate nonumber linesize=200;
ods listing;
ods html file='inline4.html' style=sasweb;
ods rtf file='inline4.rtf' notoc_data;
ods escapechar='#';

proc print data=linebr_raw;
  title 'Using the ODS ESCAPECHAR and Raw Text for line break';
run;

ods _all_ close;
```


Using the ODS ESCAPECHAR and Raw Text for line break

Obs	myvar
1	The Caffeine molecule is an alkaloid of the methylxanthine family: C ₈ H ₁₀ N ₄ O ₂
2	The Caffeine molecule is an alkaloid of the methylxanthine family: C ₈ H ₁₀ N ₄ O ₂
3	Start of Text Rest of Text

Figure 13: RTF Output from INLINE4.SAS Program

MORE STYLE ATTRIBUTE EXAMPLES

In this paper, you have seen DATA step programs create data "from scratch" to use ODS ESCAPECHAR= functionality. What if your data is already in SAS data sets? The following examples, which use SASHELP data sets, illustrate the use of ODS ESCAPECHAR control strings with various SAS procedures and ODS statements.

ODS ESCAPECHAR=, ODS TEXT=, AND THE LABEL STATEMENT

The ODS TEXT= statement is one way to have an impact on ODS output by inserting text into ODS output. For example, when you run PROC MEANS on the Sales variable in SASHELP.SHOES, by default the output table has a spanning header that looks like this (Figure 14):

Analysis Variable : Sales Total Sales					
Region	N Obs	Sum	Minimum	Mean	Maximum

Figure 14: Default Spanning Header for the SALES Variable with PROC MEANS

Notice in Figure 14 that the label for the SALES variable is "Total Sales" and that it appears on the same line as the header "Analysis Variable: Sales".

What if you wanted the string "Total Sales" to be "Shoe Sales"? That's easy. You can just change the variable label. But what if you wanted the new label "Shoe Sales" to start on a new line? One way to accomplish this is to change the table template for PROC MEANS, but another way to put the label string "Shoe Sales" on a separate line is to insert the ODS ESCAPECHAR control string into the label as a line break. Consider the following program and output (Figures 15, 16, and 17) which illustrates how to insert line breaks, font_face, and font_size style attributes into the variable label. The program also inserts text between procedures using the ODS TEXT= statement. Then, the program creates three macro variables to hold some special ASCII hexadecimal codes for the registered trademark symbol, the copyright symbol, and the cent symbol. Finally, the three macro variables are used in the ODS TEXT= statement for each destination.

PROGRAM: INLINE5.SAS

```
data _null_;
  hexcode = 'AE'x;
  call symput ('hexreg',hexcode);
  hexcode1 = 'A9'x;
  call symput('hexcopy' , hexcode1);
  hexcode2 = 'A2'x;
  call symput('hexcent' , hexcode2);
run;
** Display the hex codes in the SAS log;
%put hexreg= &hexreg;
%put hexcopy= &hexcopy;
%put hexcent= &hexcent;
```

```

ods html file='inline5.html' style=sasweb;
ods rtf file='inline5.rtf' notoc_data startpage=no;
ods pdf file='inline5.pdf' startpage=no;
ods escapechar='^';
ods noptitle;

proc means data=sashelp.shoes sum min mean max;
  class Region;
  var Sales;
  label Sales='{font_style=italic}^nShoe Sales';
run;

ods pdf text="^S={just=c font_size=18pt}PDF File^nOpens with Adobe Reader^n
&hexreg &hexcopy My 2&hexcent";
ods rtf text="^S={just=c font_size=18pt}RTF File^nOpens with Microsoft Word^n
&hexreg &hexcopy My 2&hexcent";
ods html text="^S={just=c font_size=18pt}HTML File^nOpens with Microsoft Internet
Explorer^n &hexreg &hexcopy My 2&hexcent";

proc freq data=sashelp.shoes;
  tables Product /nocum;
  label Product='{font_style=italic}Shoe Types Sold';
run;

ods _all_ close;

```

The ESCAPECHAR is set to caret (^) for this program. The only difference between this program and previous examples is that we are using the ODS ESCAPECHAR control string for variable labels in the two steps instead of creating a special data set to show the variables and ODS ESCAPECHAR= effect with PROC PRINT. The screen shots that follow show how the special formatting was handled in each of the three destinations: HTML, RTF, and PDF.

Analysis Variable : Sales Shoe Sales					
Region	N Obs	Sum	Minimum	Mean	Maximum
Africa	56	2342588.00	801.0000000	41831.93	360209.00
Asia	14	460231.00	937.0000000	32873.64	149013.00
Canada	37	4255712.00	1190.00	115019.24	757798.00
Central America/Caribbean	32	3657753.00	936.0000000	114304.78	576112.00
Eastern Europe	31	2394940.00	712.0000000	77256.13	304093.00
Middle East	24	5631779.00	449.0000000	234657.46	1298717.00
Pacific	45	2296794.00	325.0000000	51039.87	373908.00
South America	54	2434783.00	1716.00	45088.57	245757.00
United States	40	5503986.00	554.0000000	137599.65	456985.00
Western Europe	62	4873000.00	736.0000000	78596.77	502636.00

HTML File
Opens with Microsoft Internet Explorer
© © My 2ç

Shoe Types Sold		
Product	Frequency	Percent
Boot	52	13.16

Figure 15: Partial HTML Output from INLINE5.SAS with the ODS TEXT Inserted and the Italicized Labels for Both Tables

Analysis Variable : Sales <i>Shoe Sales</i>					
Region	N Obs	Sum	Minimum	Mean	Maximum
Africa	56	2342588.00	801.0000000	41831.93	360209.00
Asia	14	460231.00	937.0000000	32873.64	149013.00
Canada	37	4255712.00	1190.00	115019.24	757798.00
Central America/Caribbean	32	3657753.00	936.0000000	114304.78	576112.00
Eastern Europe	31	2394940.00	712.0000000	77256.13	304093.00
Middle East	24	5631779.00	449.0000000	234657.46	1298717.00
Pacific	45	2296794.00	325.0000000	51039.87	373908.00
South America	54	2434783.00	1716.00	45088.57	245757.00
United States	40	5503986.00	554.0000000	137599.65	456985.00
Western Europe	62	4873000.00	736.0000000	78596.77	502636.00

RTF File
Opens with Microsoft Word
® © My 2¢

<i>Shoe Types Sold</i>		
Product	Frequency	Percent
Boot	52	13.16
Men's Casual	45	11.39

Figure 16: Partial RTF Output (Shown in Print Preview Mode) from INLINE5.SAS with the ODS TEXT String Inserted and the Italicized Headers for Both Tables

Analysis Variable : Sales <i>Shoe Sales</i>					
Region	N Obs	Sum	Minimum	Mean	Maximum
Africa	56	2342588.00	801.0000000	41831.93	360209.00
Asia	14	460231.00	937.0000000	32873.64	149013.00
Canada	37	4255712.00	1190.00	115019.24	757798.00
Central America/Caribbean	32	3657753.00	936.0000000	114304.78	576112.00
Eastern Europe	31	2394940.00	712.0000000	77256.13	304093.00
Middle East	24	5631779.00	449.0000000	234657.46	1298717.00
Pacific	45	2296794.00	325.0000000	51039.87	373908.00
South America	54	2434783.00	1716.00	45088.57	245757.00
United States	40	5503986.00	554.0000000	137599.65	456985.00
Western Europe	62	4873000.00	736.0000000	78596.77	502636.00

PDF File
Opens with Adobe Reader
® © My 2¢

<i>Shoe Types Sold</i>		
Product	Frequency	Percent
Boot	52	13.16
Men's Casual	45	11.39

Figure 17: Partial PDF Output from INLINE5.SAS with the ODS TEXT String Inserted and the Italicized Headers for Both Tables

In the PDF output you can see that the JUST=C attribute override was used; however in the HTML and RTF output, the JUST=C attribute was not used. The ASCII hexadecimal codes are respected in all the ODS destinations. For a more detailed example of using ASCII characters compared to using the SYMBOL font, refer to the BONUS_HEXCHAR.SAS program in the programs that are available for download. See Zender (2007) in the Resources section.

ODS ESCAPECHAR= AND INSERTING IMAGES

The PREIMAGE and POSTIMAGE style attributes provide you a way to insert images before or after text in your ODS output file. The following program uses PROC FORMAT to associate a specific image with every product in the WHERE statement. The image will be assigned to every value of the PRODUCT column when the user-defined format is used for the POSTIMAGE style attribute value. Note that there is no usage of ODS ESCAPECHAR= in the PROC FORMAT or PROC REPORT code. However, the company logo is inserted into the SAS TITLE statement using the PREIMAGE attribute with the ODS ESCAPECHAR control string. In addition, the label for the N statistic uses the ODS ESCAPECHAR control string to set the background color. Compare the way that this form of the technique is rendered in the HTML output with the way that the HEADER for the Product variable is rendered in HTML. RTF and PDF use the ODS ESCAPECHAR control string in the LABEL option differently from the way the HTML destination uses it.

Keep in mind that the location of the image to be used may need to be a fully qualified URL for HTML reports in a production system (instead of the physical path location shown in this example). The image location for HTML files has to be the location where the image will be stored when the HTML file is rendered by the browser. This is different from the image location for PDF and RTF output files. The location for those destinations must be the location where the image is stored when the PDF or RTF file is created. Both PDF and RTF destinations grab the image at file creation time and convert the image into an internal format. Therefore, it is better in these destinations to be sure that your image is sized appropriately for these destinations. For the destinations in this program, the c:\temp location is the correct location for both file rendering and file creation; however, that may not always be the case.

PROGRAM: INLINE6.SAS

```
proc format;
  value $prdimg 'Boot' = 'c:\temp\boot.jpg'
              'Women's Dress' = 'c:\temp\dressheels.jpg'
              'Slipper' = 'c:\temp\slipper.jpg';

run;

ods html file='inline6.html' style=sasweb;
ods rtf file='inline6.rtf' notoc_data;
ods pdf file='inline6.pdf';
ods escapechar='^';

proc report data=sashelp.shoes nowd
  style(column) = {vjust=b};
  title '^S={preimage="c:\temp\shoe_logo.jpg"}';
  where product in ('Boot', 'Slipper', "Women's Dress");
  column Product Sales N;
  define Product / group
    style(header)={background=white foreground=black}
    style(column)={postimage=$prdimg.};
  define Sales / sum;
  define N / '^S={background=white foreground=black}Number of Sales';
run;

ods _all_ close;
```

The output for all three destinations is shown in Figure 18.



Figure 18: Comparison of HTML, RTF, and PDF Output from INLINE6.SAS

In Figure 18, notice how the RTF document puts each image on a separate line in the cell, while PDF and HTML put the image on the same line as the PRODUCT value.

PREVIEW OF COMING ATTRACTIONS

The ODS ESCAPECHAR capability as described in this paper will continue to work, as advertised in the Resources listed below. Technically, the ODS ESCAPECHAR= capability has been experimental ever since SAS 8.2, although many people adopted its use because of the need to enhance their output with in-line formatting. However, when SAS 9.2 is released, the ESCAPECHAR syntax will follow a function model. What this means is that instead of having this:

```
var1 = "The ^S={font_weight=bold}quick ^S={ } brown fox";
```

you would have this:

```
var1 = "The ^{style [font_weight=bold]quick} brown fox";
```

to alter the appearance of the text string in the data cell. The advantages of the new syntax are immediately apparent. The "halo" effect of the style change is limited to the single string enclosed within the curly brackets. Within the square brackets, you have the attribute-value pairs that you are changing. Another advantage of the new syntax is the ability to nest ESCAPECHAR functions. If you download the zip file of this paper and programs (see Zender (2007) in the Resources section), there is an additional set of programs that shows how the programs would change for SAS 9.2. In addition, the production version of ODS ESCAPECHAR= will include other features such as the ability to nest ODS ESCAPECHAR style functions and support for UNICODE character references. For example, either of these references would insert the UNICODE α (alpha) character into your output:

```
^{unicode 03B1} or ^{unicode alpha} (if ODS ESCAPECHAR = '^');
```

CONCLUSION

The ODS ESCAPECHAR= capability enables you to apply those last final typographic tweaks to your output. Once you understand the way that the ODS ESCAPECHAR= and control string can be used, you will wonder how you ever got along without it! Changes coming down the road for SAS 9.2 will only make ODS ESCAPECHAR= even better!

GLOSSARY OF USAGE

Terms	Usage
ODS ESCAPECHAR=	ODS ESCAPECHAR statement and the functionality that using the statement makes possible. ODS ESCAPECHAR='~';
ESCAPECHAR	The ODS ESCAPECHAR character value . For example, If the statement ODS ESCAPECHAR='~', is in effect, then the ESCAPECHAR is a tilde (~).
ODS ESCAPECHAR control string	This refers to the current ESCAPECHAR that is in effect <u>plus</u> a string that will change or affect ODS output. For example: If the statement ODS ESCAPECHAR='~' is in effect, then ~S={foreground=red} is a complete ODS ESCAPECHAR control string or ~{pageof} is a complete ODS ESCAPECHAR control string

ACKNOWLEDGMENTS

The author thanks the following people for making suggestions to improve this paper and for taking the time to review it: Jane Stroupe, Michele Ensor, Linda Jolley, and Rick Bell. As always, special thanks go to Chevell Parker, Bari Lawhorn, David Kelley, Wayne Hester, Kevin Smith, Eric Gebhart, and Dan O'Connor for answering interminable questions about ODS.

RESOURCES

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

SAS Institute Inc. 2007 (a). In-Line Formatting for ODS in SAS 8.2. Available at support.sas.com/rnd/base/topics/expv8/inline82.html.

SAS Institute Inc. 2007(b). New ODS Printer Features for SAS 9.1. Available at support.sas.com/rnd/base/topics/odsprinter/new91.html.

SAS Institute Inc. 2007(c). ODS FAQs and Concepts. Available at support.sas.com/rnd/base/topics/templateFAQ/Template.html.

You can download the programs used for this paper from the SAS Web site for Technical Papers and Presentations for SAS Global Forum 2007:

Zender, Cynthia. 2007. "Funny ^Stuff~ in My Code: Using ODS ESCAPECHAR". Technical Papers and Presentations. Cary, NC:SAS Institute Inc. Available at support.sas.com/rnd/papers/.

CONTACT INFORMATION

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

Cynthia L. Zender
SAS Institute Inc.
Home Office
Las Cruces, NM 88011
Work Phone: (505) 522-3803
Fax: (505) 521-9328
E-mail: Cynthia.Zender@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.