Paper 215-2010

The Great Escape(char)

Louise S. Hadden, Abt Associates Inc., Cambridge, MA

ABSTRACT

SAS® provides programmers with many ways to enhance ODS output in addition to the use of both SAS-supplied and user-written ODS styles. Inline formatting of titles, footnotes, text fields, and table cells, and formatting of data using user-written formats and within the DATA step can be accomplished by using ODS ESCAPECHAR. A quick overview of syntax for some of the many possibilities for enhancing ODS output with ODS ESCAPECHAR currently available in SAS® 8.2 and SAS® 9.1.3 will be presented, including underlines, pre- and post-images, special functions, line feeds, and super- and subscripts. Some new features available in SAS® 9.2 will be discussed.

INTRODUCTION

The use of ODS ESCAPECHAR allows the SAS® programmer to "pass through" formatting commands and functions into text strings (such as titles, footnotes, and ODS text fields) and variable values (through in-line style commands, PROC FORMAT, and the data step.) Some formatting commands available include (but are not limited to!): superscript, subscript, changes within text strings, bold, italic, underline, font color and font face, page x of y, line feeds, images, and destination-specific elements (such as RTF formatting commands.) Examples of all the techniques described above will be very briefly demonstrated, for multiple output destinations. For more specific treatment of ODS ESCAPECHAR, please read papers (and/or attend presentations!) by Cynthia Zender of SAS®, Scott Huntley of SAS®, Wayne Hester of SAS®, Ray Pass, Lauren Haworth, and Leroy Bessler among others.

As the name implies, ODS ESCAPECHAR is designed to be used in output designated for the Output Delivery System. The default SAS® listing destination is not able to translate the formatting commands. Below follows a snippet of a SAS listing from PROC FREQ of values that include formatting commands.

```
column1                                          Frequency     Percent
-----------------------------------------------------------------------
**The probability is 99% or more that your             1       25.00
*The probability is 95% or more that your              1       25.00
^{style [fontsize=6pt] ^{dagger}^{dagger}}The          1       25.00
^{style [fontsize=6pt] ^{dagger}}The                   1       25.00
```

In order to use these special formatting commands, the SAS® programmer must designate a character or characters which specify that formatting commands follow. Generally, this character is one that would not ordinarily be found in SAS® data or procedural commands. Examples of commonly used "ESCAPECHARs" are "^", "~", and "#". Mainframe users will want to be particularly careful in specifying ESCAPECHARs: for example, the "^" is problematic with some mainframe operating systems and software. The defaults, *ESC* (version 9) and '03'x (version 8.2), contain multiple keystrokes and are used less frequently. It is also possible to specify longer ESCAPECHARs.

The ODS ESCAPECHAR statement should be placed inside the ODS call(s), and the ESCAPECHAR specified must be the same as in your data and/or style commands. The standard usage of ODS ESCAPECHAR is:

ESCAPECHAR={attribute=value};

However, special formatting commands can be inserted in variables using PROC FORMAT or in the data step. The variable column1 shown both above and below has formatting commands resulting from a data step command.

The same data shown in the listing above was output to an ODS destination.

```
ODS HTML FILE='freq.xls' PATH=odsout STYLE=styles.journal;
ODS ESCAPECHAR='^';
PROC FREQ DATA=footer;
    TABLES column1;
RUN;
ODS HTML CLOSE;
```

column1	Frequency	Percent
****The probability is 99% or more that your facility's rate is greater than the state average.**	1	25
***The probability is 95% or more that your facility's rate is greater than the state average.**	1	25
††The probability is 99% or more that your facility's rate is greater than the national average.	1	25
†The probability is 95% or more that your facility's rate is greater than the national average.	1	25

For the remainder of this paper, the ESCAPECHARs used will be the caret (^).

```
ODS ESCAPECHAR='^';
```

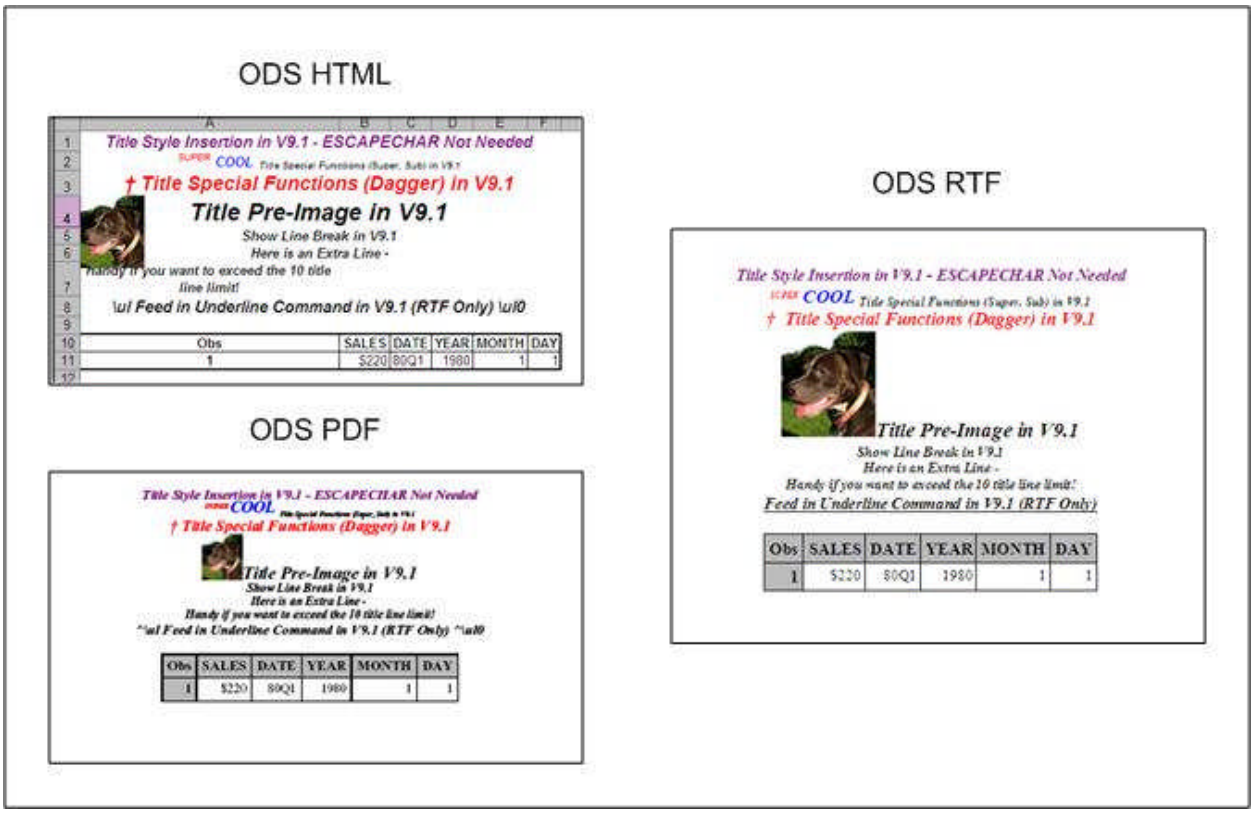
USING ODS ESCAPECHAR IN TITLES AND FOOTNOTES

ODS ESCAPECHAR is most commonly used in titles, footnotes and ODS text fields. The syntax for using ODS ESCAPECHAR in titles and footnotes is fairly consistent across destinations or tagsets, with some important exceptions which will be explored below. Style attributes CAN be inserted into titles and footnotes with ODS ESCAPECHAR, but it is not necessary unless you are "nesting" various attributes, one or more of which require the use of ODS ESCAPECHAR. In addition to style attributes, text "functions" such as superscripts, subscripts, and special characters, pre- and post-images, line breaks and non-breaking lines, and RTF or HTML control strings may be inserted into titles and footnotes using ODS ESCAPECHAR. Finally, special functions such as {pageof} (RTF only) and {thispage}/{lastpage} (RTF and PDF only) can be inserted into titles and footnotes using ODS ESCAPECHAR.

The code below demonstrates using ODS ESCAPECHAR and special functions such as superscript, subscript and dagger; the insertion of pre-images; the insertion of line breaks; and feeding an RTF control string (underline) into titles. Identical syntax can be used for footnotes. Screenshots of the different destination outputs follow below.

```
ODS PDF FILE='escapechar_title.pdf';
ODS RTF FILE='escapechar_title.doc' PATH=odsout STYLE=styles.rtf BODYTITLE;
ODS HTML FILE='escapechar_title.xls' PATH=odsout STYLE=styles.journal;
ODS ESCAPECHAR='^';

PROC PRINT DATA=sashelp.retail (OBS=1);
TITLE1 HEIGHT=3 JUSTIFY=center italic COLOR=purple 'Title Style Insertion in V9.1 -
ESCAPECHAR Not Needed';
TITLE2 HEIGHT=2 COLOR=red '^{super SUPER}' COLOR=blue 'COOL ' COLOR=black
 '^{sub Title Special Functions (Super, Sub) in V9.1}';
TITLE3 HEIGHT=4 COLOR=red '^{dagger} Title Special Functions (Dagger) in V9.1';
TITLE4 HEIGHT=5 '^S={preimage="bonnietiny.jpg"}' "Title Pre-Image in V9.1";
TITLE5 HEIGHT=2 'Show Line Break in V9.1 ^nHere is an Extra Line –
 ^nHandy if you want to exceed the 10 title line limit!';
```

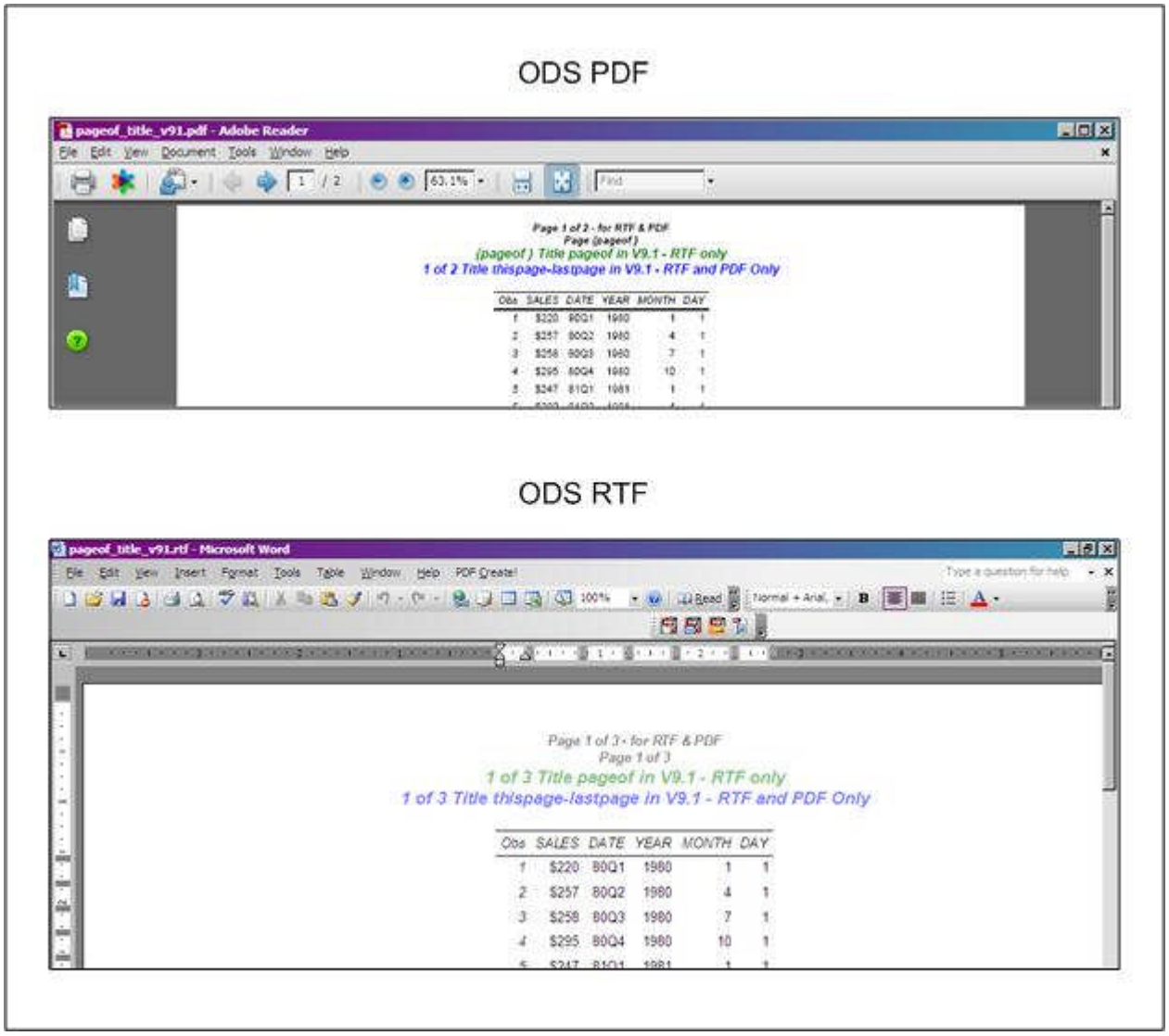
```
TITLE6 HEIGHT=3 '^\ul Feed in Underline Command in V9.1 (RTF Only) ^\ul0';
RUN;

ODS _ALL_ CLOSE;
```



The code below demonstrates using ODS ESCAPECHAR and the special Printer destination specific {pageof} (RTF only) and {thispage}/{lastpage} (RTF and PDF only) to insert page numbering into titles. Identical syntax can be used for footnotes. Screenshots of the different destination outputs follow below. *NOTE: I have become accustomed to sending RTF output to the .doc extension out of sheer laziness (when "saving as", one doesn't have to change the extension!) These particular functions will NOT work if you use the .doc extension (I found this out the hard way!)*

```
ODS PDF FILE='pageof_title_v91.pdf' STYLE=styles.journal;
ODS RTF FILE='pageof_title_v91.rtf' PATH=odsout STYLE=styles.journal NOTOC_DATA;
ODS ESCAPECHAR='^';

PROC PRINT DATA=sashelp.retail (OBS=100);
TITLE1 'Page ^{thispage} of ^{lastpage} - for RTF & PDF';
TITLE2 'Page ^{pageof}';
TITLE3 HEIGHT=4 COLOR=green '^{pageof} Title pageof in V9.1 - RTF only';
TITLE4 HEIGHT=4 COLOR=blue '^{thispage} of ^{lastpage} Title thispage-lastpage in
V9.1 - RTF and PDF Only';
RUN;

ODS _ALL_ CLOSE;
```

Results in:



USING ODS ESCAPECHAR IN ODS TEXT FIELDS (PRINTER DESTINATIONS ONLY)

The ODS text command within ODS statements in “Printer” destinations such as RTF and PDF allows the user to insert lines of text after titles and procedures at will – or without running any procedures at all!

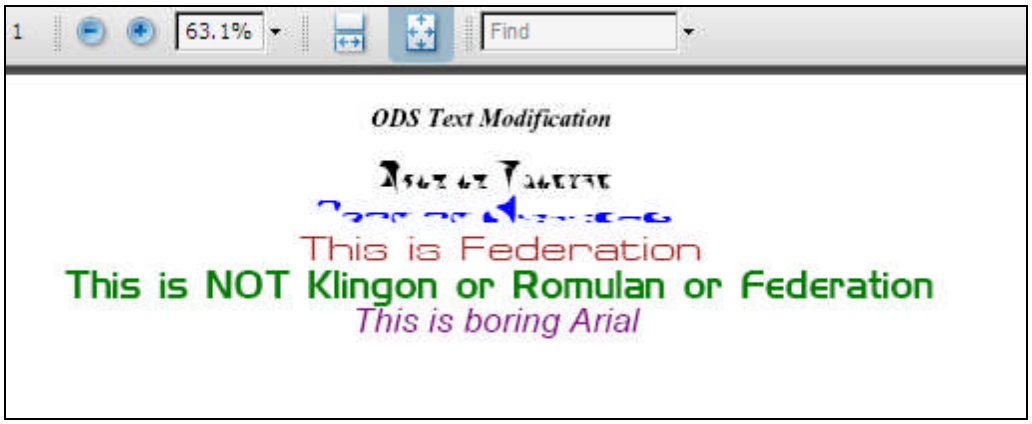
```
ODS RTF TEXT='{\animtext1 This is an example of inserting a text field.}';
```

Results in:

```
This is an example of inserting a text field.
```

```
ODS PDF FILE='format_ODS_text.pdf';
ODS ESCAPECHAR='^';
TITLE 'ODS Text Modification';
ODS PDF TEXT = '^S={FONT_SIZE=18pt JUST=c FONT_FACE=Klingon} This is Klingon';
ODS PDF TEXT = '^S={FONT_SIZE=18pt FOREGROUND=blue JUST=c FONT_FACE=Romulan}
 This is Romulan';
ODS PDF TEXT = '^S={FONT_SIZE=18pt FONT_WEIGHT=bold FOREGROUND=brown JUST=c
 FONT_FACE=Federation} This is Federation';
ODS PDF TEXT = '^S={FONT_SIZE=18pt FOREGROUND=green JUST=c FONT_FACE=trek}
 This is NOT Klingon or Romulan or Federation';
ODS PDF TEXT = '^S={FONT_SIZE=18pt FOREGROUND=purple FONT_STYLE=italic JUST=c
 FONT_FACE=arial} This is boring Arial';

ODS PDF CLOSE;
```

Results in:



USING ODS ESCAPECHAR IN PROCEDURAL OUTPUT

The syntax for using ODS ESCAPECHAR in procedural output varies by both procedure and destination. Formatting commands can also be made creating variables containing ODS ESCAPECHAR commands in the data step or through PROC FORMAT. In-line formatting commands are most commonly used in the reporting procedures (PROC PRINT , PROC REPORT and PROC TABULATE); however, by passing ODS ESCAPECHAR commands through in the data step or in PROC FORMAT, text “functions” such as superscripts, subscripts, and special characters, line breaks and non-breaking lines, and RTF or HTML control strings may be used in the ODS output of other procedures as well, as shown in the first example using PROC FREQ above. Selected samples will be shown below.

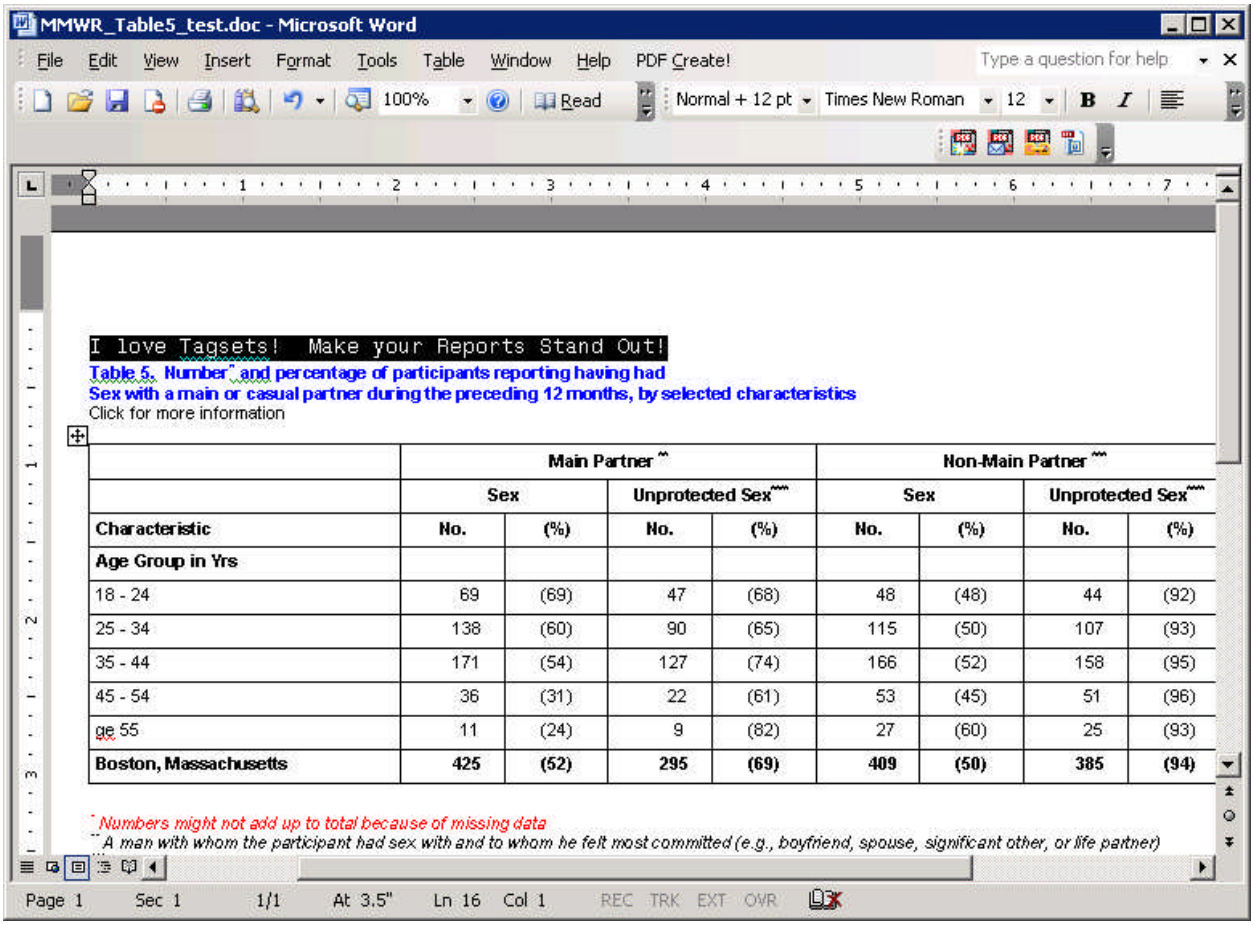
SUPERSCRIPT - RTF

The code below demonstrates using ODS ESCAPECHAR and the superscript function in procedural output. Subscripts work in a similar way. A screenshot demonstrating the use of the superscript function in RTF output follows below.

```
OPTIONS ps=55 ls=175 errorabend missing=' ' orientation=landscape nocenter nodate
nonumber;

ODS RTF FILE='MMWR_Table5_test.doc' PATH=odsout STYLE=styles.nmep BODYTITLE;
ODS ESCAPECHAR = "^";

PROC REPORT nowd DATA=table5 SPACING=8
    style(report)=[CELLPADDING=3pt VJUST=b]
    style(lines)=[JUST=left FONT_FACE=Arial]
    style(header)=[JUST=center FONT_FACE=Arial FONT_WEIGHT=bold FONT_SIZE=8pt]
```

```
    ls=176;
    COLUMNS fmtvar ('Main Partner ^{super **} ' ( ('Sex' count1 colper1)
                    ('Unprotected Sex^{super****}' count2 colper2) ) )
                   ('Non-Main Partner ^{super***}' ( ('Sex' count3 colper3)
                    ('Unprotected Sex^{super****}' count4 colper4) ) )
            count boldit;
  DEFINE boldit / display ' ' NOPRINT;
  DEFINE fmtvar / style(COLUMN)={JUST=l FONT_FACE=Arial
          FONT_SIZE=8pt CELLWIDTH=300 ASIS=on}
                    style(HEADER)={JUST=l FONT_FACE=Arial FONT_WEIGHT=bold
          FONT_SIZE=8pt CELLWIDTH=300 ASIS=on};
    . . .
```

Results in:

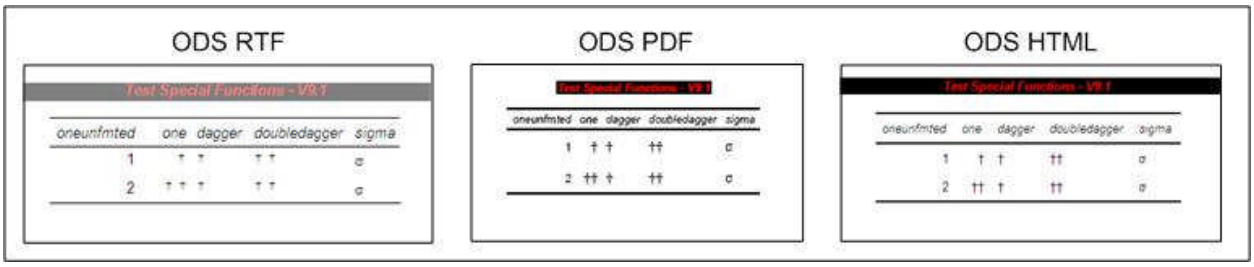


SPECIAL FUNCTIONS

The code below demonstrates using ODS ESCAPECHAR and some special functions (dagger and output) in procedural output. Both data step and PROC FORMAT functionality are used. Screenshots of the different destination outputs follow below. *NOTE: as in the special page numbering functions described above, HTML output should have an HTML extension, and RTF output should have an RTF extension, for these special functions to work reliably.*

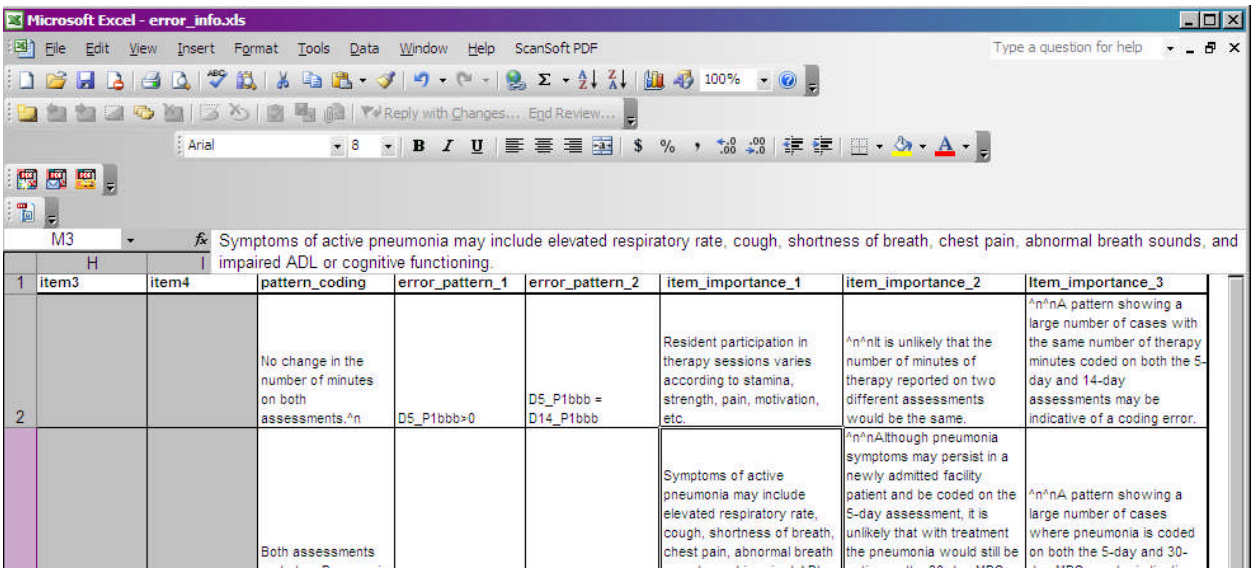
```
PROC FORMAT;
    VALUE one 1='^{dagger}'
              2='^{dagger}^{dagger}';
```

```
RUN;

DATA one;
    one=1;
    dagger='^{dagger}';
    doubledagger='^{dagger}^{dagger}';
    sigma='^{sigma}';
RUN;

DATA two;
    one=2;
    dagger='^{dagger}';
    doubledagger='^{dagger}^{dagger}';
    sigma='^{sigma}';
RUN;

DATA printit;
    SET one two;
    oneunfmted=one;
RUN;

ODS PDF FILE='dagger_v91.pdf' STYLE=styles.journal;
ODS RTF FILE='dagger_v91.rtf' PATH=odsout STYLE=styles.journal;
ODS HTML FILE='dagger_v91.html' PATH=odsout STYLE=styles.journal;

ODS ESCAPECHAR='^';

PROC PRINT DATA=printit noobs;
    VAR oneunfmted one dagger doubledagger sigma;
FORMAT one one.;
TITLE1 COLOR=red BCOLOR=black 'Test Special Functions - V9.1';
RUN;

ODS _ALL_ CLOSE;
```



LINE BREAKS AND OTHER SPECIAL CODES

Below follows a screenshot of an Excel spreadsheet with 3 fields on item importance. To produce 15,000+ reports on nursing home facilities, this spreadsheet is imported into SAS® and the three items concatenated into a single field, which is then used to generate a item-level page for each facility and item. You will notice ^n^n in the second 2 fields on item importance. This generates 2 line breaks (essentially starts a new paragraph) within the variable.




```
DATA printimp;
    LENGTH importance $ 1032;
    SET itemtemp;
    intro="Importance of "||TRIM(item_name)||" as an MDS Accuracy Trigger:";
    importance=CATX('  ',item_importance_1,item_importance_2,item_importance_3);
RUN;

DATA _null_;
   SET itemtemp;
   CALL SYMPUT("callitem",TRIM(item_name)||' ('||TRIM(PROPCASE(item_short))||')');
   CALL SYMPUT("calldef",item_def);
RUN;

DATA _null_;
   SET printimp;
   CALL SYMPUT("intro",intro);
RUN;

. . .

ODS PROCLABEL="Importance";
ODS ESCAPECHAR='^';

PROC REPORT nowd DATA=printimp spacing=8
    style(report)=[cellpadding=3pt vJUST=b]
    style(lines)=[JUST=left FONT_FACE=Arial]
    style(header)=[JUST=left FONT_FACE=Arial FONT_WEIGHT=bold FONT_STYLE=italic
FONT_SIZE=10pt]
    ls=176 contents="Importance";
    COLUMNS ("&intro" importance);
    DEFINE importance / ' ' style(COLUMN)={JUST=l FONT_FACE=Arial FONT_STYLE=italic
           FONT_SIZE=8pt CELLWIDTH=1000}
                    ;
title1 ;
```

The code snippets shown above result in PDF pages that include the following:

Number of 90-90 day MDS Paired Assessments	31	38,598	1,207,549
Number of Paired Assessments coded on Item(s)	31	38,582	1,207,193
Number of Residents with Possible Error Pattern	0	1,035	66,813
Average Possible Error Rate	0.0%	2.7%	5.5%
Possible Error Rate Percentile of Facility	NA	16	11

**The probability is 95% or more that your facility's rate is greater than the state average.*

***The probability is 99% or more that your facility's rate is greater than the state average.*

†The probability is 95% or more that your facility's rate is greater than the national average.

‡The probability is 99% or more that your facility's rate is greater than the national average.

Potential MDS Error Pairs

Possible error rate in facility did not exceed parameters: no resident records were selected.

Importance of Pain Frequency as an MDS Accuracy Trigger:

Adequate pain management should decrease the frequency of pain.

After 90 days of pain management, residents should not be reporting or showing evidence of pain on a daily basis.

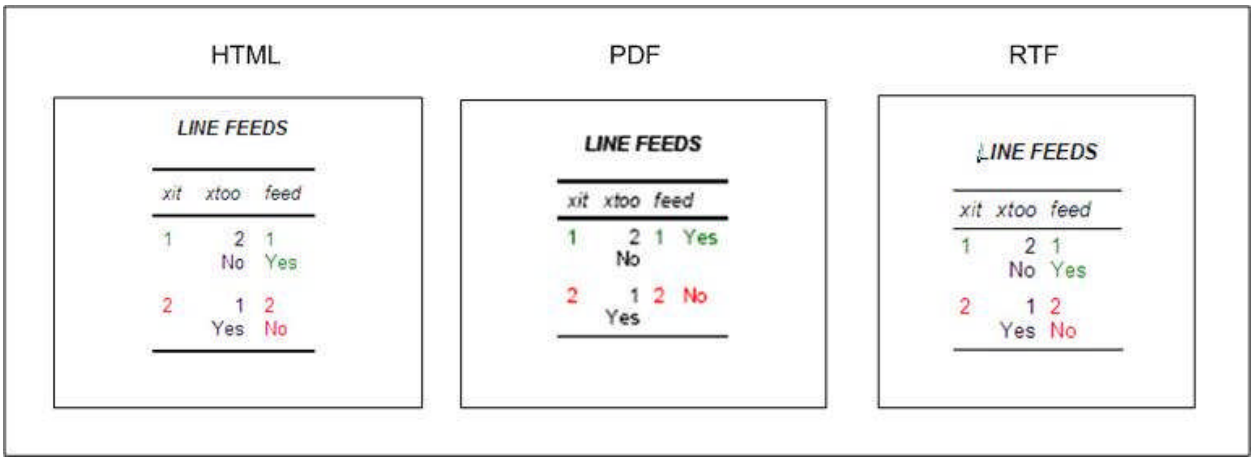
A pattern showing a large number of cases of daily pain on two consecutive 90-day assessments may be indicative of a coding error.

HTML AND RTF CONTROL STRINGS

We have seen above that we can insert line feeds with ODS ESCAPECHAR, into all three destination "families." There may be other similar "events" that we would like to have in our output, such as tabs. These events are often destination-specific. In addition, we may want to "stack up" or nest these events and style attributes. ODS ESCAPECHAR R allows this functionality. Note that insertions done with ODS ESCAPECHAR R are non-printing in list output while insertions done with ESCAPECHAR N do print in list output. This may affect the LENGTH of character variables including ESCAPECHAR R sequences in data sets and data set size – they will be shorter. On the other hand, you have to know your control strings pretty well! The code and output samples below show how to insert line feeds both using ESCAPECHAR N and ESCAPECHAR R, and how to add style attributes.

```
PROC FORMAT;
     VALUE xit 1="Green"
               2="Red";
     VALUE xtoo 1="1 ^n Yes"  /* ODS ESCAPECHAR N */
                2="2 ^n No";
     VALUE $ fit "1"-<"2"="Green"
                 "2"-<"3"="Red";
RUN;

DATA one;
    LENGTH feed $ 132;
    xit=1;
    xtoo=2;
    feed=cat("1 ","^R/HTML'<BR/>' ^R/RTF'\line'"," Yes");  /* ODS ESCAPECHAR R */
RUN;

DATA two;
    LENGTH feed $ 132;
    xit=2;
    xtoo=1;
    feed=cat("2 ","^R/HTML'<BR/>' ^R/RTF'\line'"," No");  /* ODS ESCAPECHAR R */
RUN;
```

```
    /* Note that the string defining the variable FEED includes a line feed
       for both HTML and RTF */
    /* The variable FEED will not print out as expected in PDF */

DATA printit;
    SET one two;
RUN;

ODS PDF FILE='style_format_v91.pdf' STYLE=styles.journal;
ODS RTF FILE='style_format_v91.rtf' PATH=odsout STYLE=styles.journal BODYTITLE;
ODS HTML FILE='style_format_v91.html' PATH=odsout STYLE=styles.journal;
ODS ESCAPECHAR='^';

PROC PRINT DATA=printit LABEL NOOBS;
    VAR xit / STYLE={JUST=l FOREGROUND=xit.};
    VAR xtoo;
    VAR feed / STYLE={JUST=l FOREGROUND=$fit.};
FORMAT xtoo xtoo. ;
TITLE1 'Line Feeds';
RUN;

ODS _ALL_ CLOSE;
```



As can be seen, the variable FEED is shown both color-coded and with a line feed in the HTML and RTF destinations. The HTML and RTF control strings embedded in the FEED variable do not print in the PDF destination.

FORMATTING CONTROL OF SPANNING HEADERS IN PROC REPORT

In order to format spanning headers, ODS ESCAPECHAR must be used. I discovered this technique with excellent advice from Cynthia Zender of SAS® when I was trying to color-coordinate the headers with the alternating colored columns for a set of reports. The snippet of the program below shows how to do this. Note that while you can format the background color of actual columns in the define statements without using ODS ESCAPECHAR, formatting the background colors of spanning headers does require ODS ESCAPECHAR.

```
ODS RTF FILE='testing_x_race.doc' PATH=odsout STYLE=styles.nmep BODYTITLE;

ODS ESCAPECHAR='^';

PROC REPORT nowd DATA=page1 spacing=8
    style(report)=[cellpadding=3pt vJUST=b]
    style(header)=[JUST=center FONT_FACE=Arial FONT_WEIGHT=bold FONT_SIZE=8pt]
    style(lines)=[JUST=left FONT_FACE=Arial] ls=132;
    COLUMNS fmtvar ('^S={BACKGROUND=lightblue}IDU Cycle 1 - Overall'
```

```
                  count_msm1 colper_msm1)
                ('IDU Cycle 1 - White, Non-Hispanic' count_idu1 colper_idu1)
              ('^S={BACKGROUND=lightblue}IDU Cycle 1 - Black, Non-Hispanic'
                count_het1 colper_het1)
              ('IDU Cycle 1 - Hispanic' count_whbs1 colper_whbs1)
              ('^S={BACKGROUND=lightblue}IDU Cycle 1 - Other Race'
                count_whbs2 colper_whbs2) boldit;
DEFINE boldit / display ' ' NOPRINT;
DEFINE fmtvar / style(COLUMN)={JUST=l FONT_FACE=Arial
        FONT_SIZE=8pt CELLWIDTH=300 ASIS=on}
                  style(HEADER)={JUST=l FONT_FACE=Arial FONT_WEIGHT=bold
        FONT_SIZE=8pt CELLWIDTH=300 ASIS=on};
DEFINE count_msm1 / style(COLUMN)={JUST=c FONT_FACE=Arial BACKGROUND=lightblue
        FONT_SIZE=8pt CELLWIDTH=90 ASIS=on}
                  style(HEADER)={JUST=c FONT_FACE=Arial FONT_WEIGHT=bold
                  BACKGROUND=lightblue FONT_SIZE=8pt CELLWIDTH=90 ASIS=on};
DEFINE colper_msm1 / style(COLUMN)={JUST=c FONT_FACE=Arial BACKGROUND=lightblue
        FONT_SIZE=8pt CELLWIDTH=90 ASIS=on}
                  style(HEADER)={JUST=c FONT_FACE=Arial FONT_WEIGHT=bold
                  BACKGROUND=lightblue FONT_SIZE=8pt CELLWIDTH=90 ASIS=on};
```

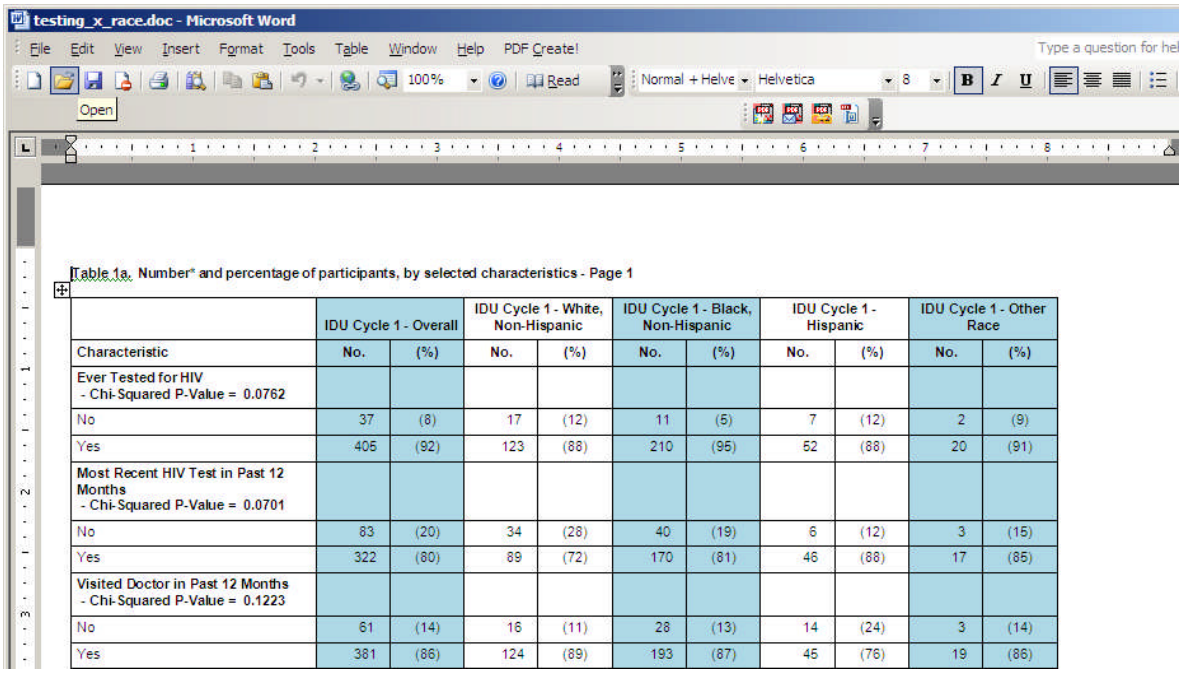


Table 1a. Number* and percentage of participants, by selected characteristics - Page 1

	IDU Cycle 1 - Overall		IDU Cycle 1 - White, Non-Hispanic		IDU Cycle 1 - Black, Non-Hispanic		IDU Cycle 1 - Hispanic		IDU Cycle 1 - Other Race	
Characteristic	No.	(%)	No.	(%)	No.	(%)	No.	(%)	No.	(%)
Ever Tested for HIV - Chi-Squared P-Value = 0.0762										
No	37	(8)	17	(12)	11	(5)	7	(12)	2	(9)
Yes	405	(92)	123	(88)	210	(95)	52	(88)	20	(91)
Most Recent HIV Test in Past 12 Months - Chi-Squared P-Value = 0.0701										
No	83	(20)	34	(28)	40	(19)	6	(12)	3	(15)
Yes	322	(80)	89	(72)	170	(81)	46	(88)	17	(85)
Visited Doctor in Past 12 Months - Chi-Squared P-Value = 0.1223										
No	61	(14)	16	(11)	28	(13)	14	(24)	3	(14)
Yes	381	(86)	124	(89)	193	(87)	45	(76)	19	(86)

ODS ESCAPECHAR ENHANCEMENTS IN VERSION 9.2

SAS® developers have added many enhancements to the use of ODS ESCAPECHAR in 9.2. Among these are the addition of the ability to nest inline style attributes, the ability to use Unicode characters, new style functions such as nbspace, nbline and text decoration, and new fonts. Examples of selected enhancements follow below.

NESTING OF INLINE STYLE ATTRIBUTES

SAS 9.2 allows you to specify several style attributes for a text string without having to reset the previously used style. The basic syntax is:

^{styles <style-element-name><[style-attribute-specification(s)]> formatted text}

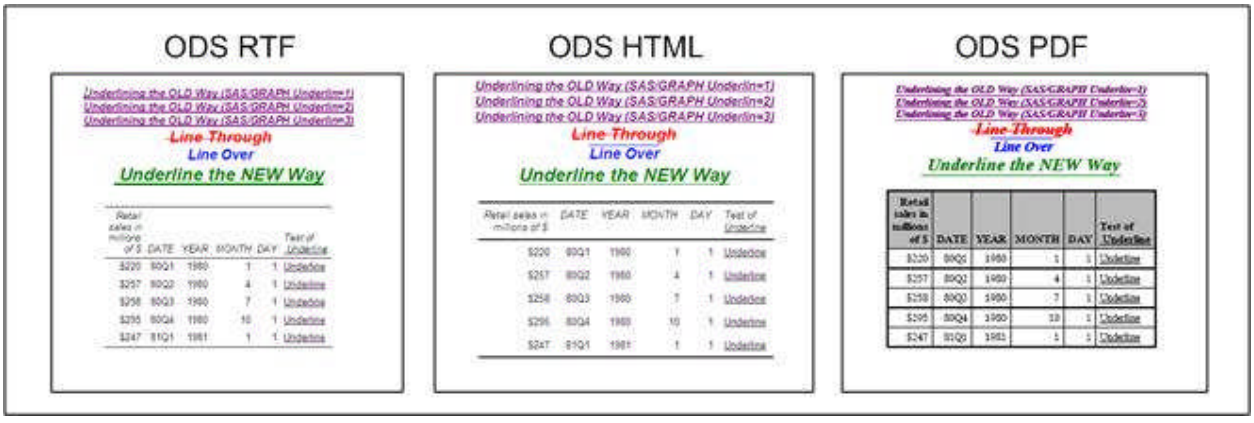
This syntax can be used in titles, footnotes, formats, and the data step, as shown above in examples not specific to version 9.2. Some new style elements that can be modified using SAS version 9.2 include the borders of cells.

NEW INLINE STYLE FUNCTIONS

New inline style functions in SAS 9.2 include nbspace and nbline, which insert non-breaking blank space(s) and blank line(s) respectively when specified. Also included in 9.2 is the ability to write your own style functions (which require the use of ODS ESCAPECHAR) but that's a subject for another paper!

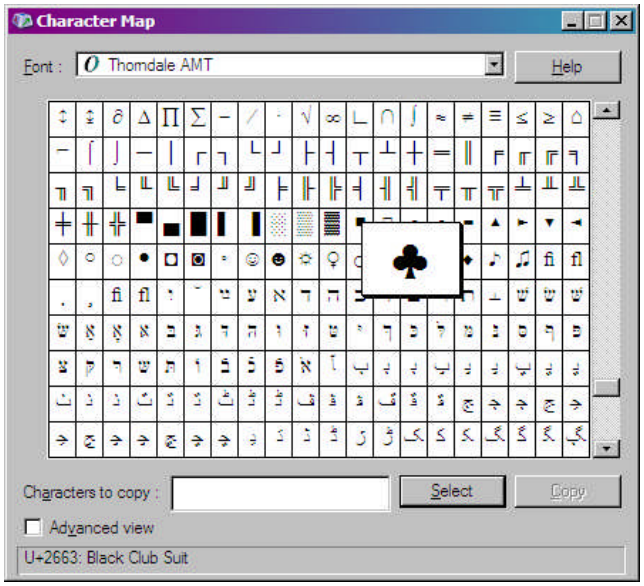
Another new style function is TextDecoration. It incorporates such items as underlines, overlines and strike-throughs. The code and samples below demonstrate underlining both using TextDecoration and using the SAS/GRAPH "UNDERLIN" syntax (available in earlier versions), in titles, labels and variables. Overlines and strike-throughs work in a similar fashion (TextDecoration only!)

```
DATA printit;
    SET sashelp.retail (OBS=5);
    underline_test='^S={TEXTDECORATION=underline}Underline';
LABEL underline_test='Test of^n ^S={TEXTDECORATION=underline} Underline';
RUN;

ODS PDF FILE='text_decoration_v92.pdf' NOTOC;
ODS RTF FILE='text_decoration_v92.doc' PATH=odsout STYLE=styles.journal BODYTITLE;
ODS HTML FILE='text_decoration_v92.html' PATH=odsout STYLE=styles.journal;
ODS ESCAPECHAR='^';

PROC PRINT DATA=printit LABEL NOOBS;

TITLE1 HEIGHT=3 JUSTIFY=center italic COLOR=purple UNDERLIN=1
 'Underlining the OLD Way (SAS/GRAPH UNDERLIN=1)';
TITLE2 HEIGHT=3 JUSTIFY=center italic COLOR=purple UNDERLIN=2
 'Underlining the OLD Way (SAS/GRAPH UNDERLIN=2)';
TITLE3 HEIGHT=3 JUSTIFY=center italic COLOR=purple UNDERLIN=3
 'Underlining the OLD Way (SAS/GRAPH UNDERLIN=3)';
TITLE4 '^S={TEXTDECORATION=line_through FONT_SIZE=16pt COLOR=red}
 Line Through';
TITLE5 '^S={TEXTDECORATION=overline COLOR=blue FONT_SIZE=14pt}
 Line Over';
TITLE6 '^S={TEXTDECORATION=underline COLOR=green FONT_SIZE=18pt}
 Underline the NEW Way';
RUN;

ODS _ALL_ CLOSE;
```

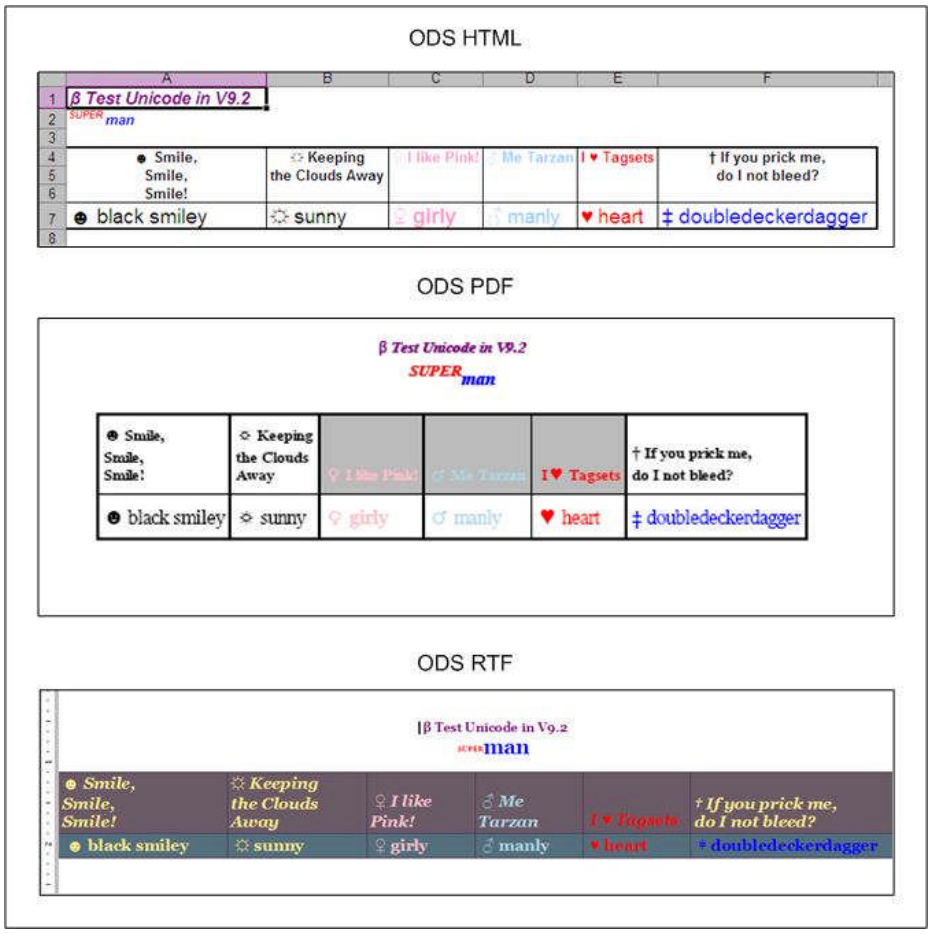


UNICODE CHARACTERS

The following program creates a data set with a number of variables containing Unicode symbols using the Unicode style function, which is new in V9.2. These variables are also labeled with Unicode symbols and the output is titled with Unicode symbols. As can be seen in the graphic below, these Unicode symbols are translated by ODS into graphic characters. This allows the addition of useful characters such as Greek letters, etc. without changing fonts. The syntax is ^{Unicode xxxx} where ^ is the specified ODS ESCAPECHAR and xxxx is the 4-digit Unicode value. To look up Unicode values on a system running Windows XP, open Start -> Programs -> Accessories -> System Tools -> Character Map. For example, I chose Thorndale AMT, which is used by SAS® V9.2. When a character on the screen is selected and/or hovered over, the Unicode VALUE is shown on the bottom of the screen (in this case the Black Club Suit is Unicode VALUE 2663.)




```
DATA escapechar_v92;
    smiley='^{style [fontsize=14pt] ^{unicode 263B} black smiley}';
    sunny='^{style [fontsize=14pt] ^{unicode 263C} sunny}';
    girly='^{style [fontsize=14pt COLOR=pink fontweight=bold] ^{unicode 2640} girly}';
    manly='^{style [fontsize=14pt COLOR=lightblue] ^{unicode 2642} manly}';
    heart='^{style [COLOR=red fontsize=14pt] ^{unicode 2665} heart}';
    doubledagger='^{style [fontsize=14pt COLOR=blue]
                  ^{unicode 2021} doubledeckerdagger}';
    LABEL smiley='^{unicode 263b} Smile,^n Smile,^n Smile!'
          sunny='^{unicode 263c} Keeping ^nthe Clouds Away'
          girly='^{style [COLOR=pink] ^{unicode 2640} I like Pink!}'
          manly='^{style [COLOR=lightblue] ^{unicode 2642} Me Tarzan}'
          heart='^{style [COLOR=red] I ^{unicode 2665} Tagsets}'
          doubledagger='^{unicode 2020} If you prick me,^n do I not bleed?';
RUN;

ODS PDF FILE='unicode.pdf';
ODS RTF FILE='unicode.doc' PATH=odsout STYLE=styles.astronomy BODYTITLE;
ODS HTML FILE='unicode.xls' PATH=odsout STYLE=styles.journal;
ODS ESCAPECHAR='^';
PROC PRINT DATA=escapechar_v92 LABEL NOOBS;
    VAR smiley sunny girly manly heart doubledagger;
TITLE1 HEIGHT=3 '^{style [JUST=c COLOR=purple] ^{unicode beta} Test Unicode in V9.2}';
TITLE2 HEIGHT=2 COLOR=red '^{super SUPER}' COLOR=blue 'man';
RUN;

ODS _ALL_ CLOSE;
```



REAL WORLD SGF 2010

My exploration of ODS ESCAPECHAR has facilitated the production of some great reports over the years. Many of the examples shown above are not what you would expect to see in everyday SAS reporting, but serve to show what could be done. The example below is an edited screenshot of the front page of a sample PDF (one of over 15,000 3 page, data driven reports for nursing home facilities) generated monthly for a "real world" project. Techniques involving ODS ESCAPECHAR used in these reports include: insertion of a government agency logo (changed to my company's logo here), insertion of line feeds (^n), insertion of inline style commands (underline, font color, italic, font size) and Unicode characters (the stars).

Code snippets used to perform all these techniques are shown below.

Logo and Line Feeds (ODS TEXT):

```
ODS listing CLOSE;

OPTIONS topmargin=.5in leftmargin=1in rightmargin=1in papersize=letter;

ODS PDF file =
".\facilityPDFs\A5set1##a50000&fnum.##&facnode.5star_report8_&filedate..pdf"
STYLE=styles.minimal NOTOC AUTHOR='Abt Associates Inc.' BOOKMARKLIST=hide
ANCHOR='Table' COMPRESS=9;

ODS ESCAPECHAR='^';
ODS PDF STARTPAGE=no;
```

```
ODS PROCLABEL '5 Star Ratings';

ODS TEXT='^S={preimage="abtlogosmall.tif"}';
ODS TEXT="^S={FONT_SIZE=14pt FONT_WEIGHT=bold JUST=c FONT_FACE=arial}Nursing Home
Compare";
ODS TEXT="^S={FONT_SIZE=14pt FONT_WEIGHT=bold JUST=c FONT_FACE=arial}5-Star Ratings
of Nursing Homes";
ODS TEXT='^n ';
ODS TEXT="^n^S={FONT_SIZE=12pt JUST=l FONT_FACE=arial}Nursing Home Rating Report";
ODS TEXT="^S={FONT_SIZE=12pt JUST=l FONT_FACE=arial}Incorporating data reported
through &rptdt.";
ODS TEXT='^n ';

ODS PDF STARTPAGE=no;

PROC REPORT NOWD DATA=rptline . . .
```

NOTE: Make sure to use LENGTH statements in a data step so that the variables are long enough to accommodate the extra formatting information. Use of the CATT function when concatenating strings will remove extra blanks at the end of input strings.

Inline Styles (italic, font size):

```
LENGTH blurb blurb0 blurb1 blurb2 blurb3 blurb4 blurb5 blurb6 $ 8128 . . .;

sent0='^{style [fontweight=bold fontSTYLE=italic fontsize=12pt COLOR=blue]
IMPORTANT HELPLINE NOTE:  Live staff are available this month to answer calls.';
    sent1='  Helpline hours of operation are 9:00 AM to 5:00 PM Eastern Standard
Time,';
    sent2='  Monday through Friday, July 13th through August 3rd.};';
    blurb0=catt(sent0,sent1,sent2);
    . . .

blurb1=catt(sentence0,sentence1,sentence1a,sentence1b,sentence1c,sentence1d,sentence
1e,sentence2);
```

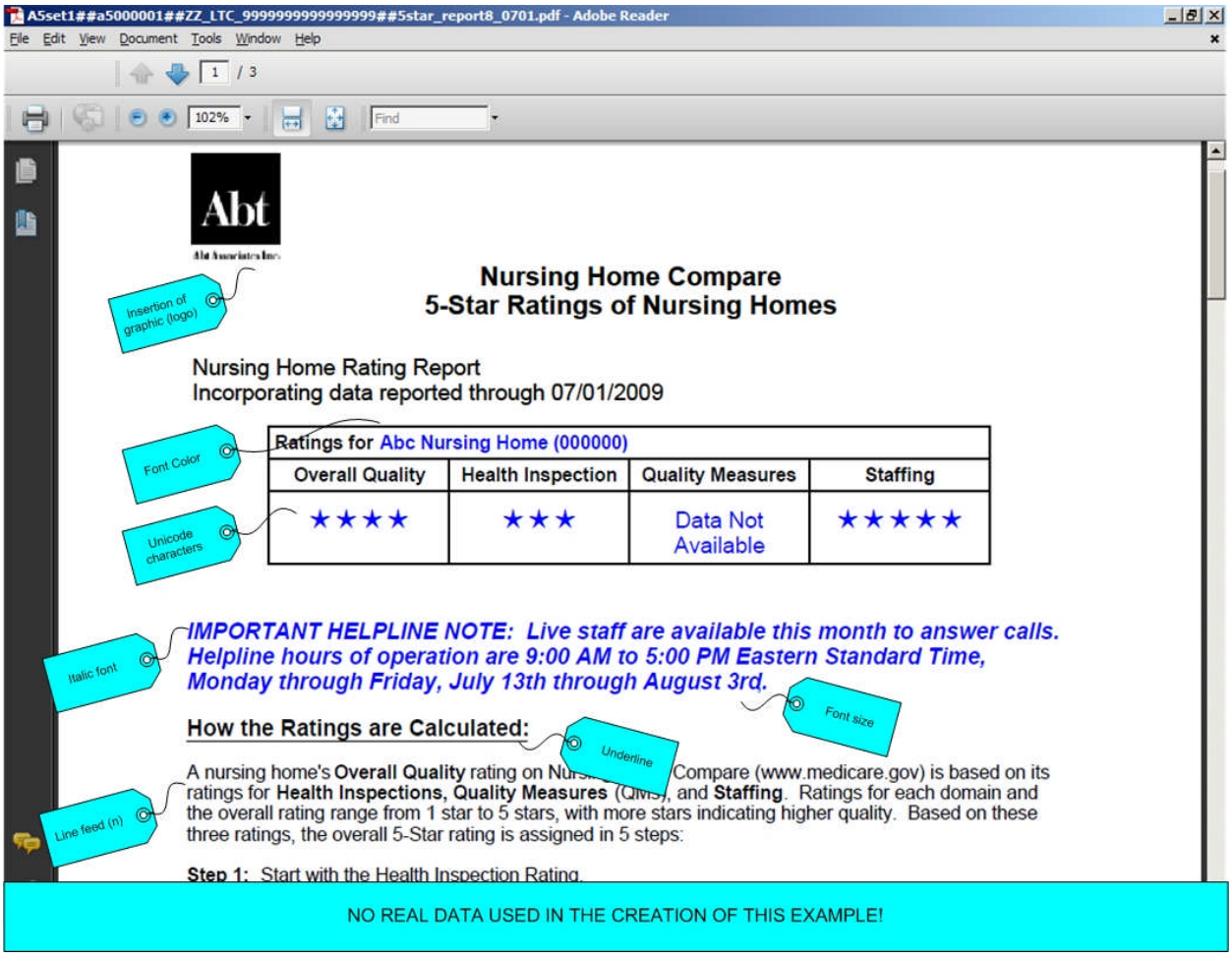
Inline Styles (bold, underline, font size) and Line feeds (Data Step):

```
    sentence0='^n^n^{style [fontweight=bold TEXTDECORATION=underline
fontsize=12pt]How the Ratings are Calculated:}';
    sentence1="^n^nA nursing home's";
    sentence1a='^{style [fontweight=bold] Overall Quality}';
    sentence1b=' rating on Nursing Home Compare (www.medicare.gov) is based on its
ratings for';
    sentence1c='^{style [fontweight=bold] Health Inspections, Quality Measures}';
    sentence1d=' (QMs), and';
    sentence1e='^{style [fontweight=bold] Staffing}.';
    . . .

blurb1=catt(sentence0,sentence1,sentence1a,sentence1b,sentence1c,sentence1d,sentence
1e,sentence2);
```

Unicode characters: [Accomplished via PROC FORMAT]

```
PROC FORMAT;

VALUE $starfmt '10'='^{Unicode 2605}'
               '20'='^{Unicode 2605}^{Unicode 2605}'
               '30'='^{Unicode 2605}^{Unicode 2605}^{Unicode 2605}'
               '40'='^{Unicode 2605}^{Unicode 2605}^{Unicode 2605}^{Unicode 2605}'
               '50'='^{Unicode 2605}^{Unicode 2605}^{Unicode 2605}
```

```
^{Unicode 2605}^{Unicode 2605}'
              '70'='Too New to Rate'
              '90'='Data Not Available'
;

RUN;

DATA rptline;
    LENGTH col1 col2 col3 col4 col5 col1a col2a col3a col4a col5a $ 200;
    SET in.provider_rating&filedate (where=(provnum="&provnum"));
    col1a=put(overall_rating,$starfmt.);
    col2a=put(survey_rating,$starfmt.);
    col3a=put(quality_rating,$starfmt.);
    col4a=put(staffing_rating,$starfmt.);
    col5a=put(rn_staffing_rating,$starfmt.);

    IF overall_rating in('10','20','30','40','50') THEN
        col1=col1a;
    IF overall_rating in('70','90') THEN
        col1=catt('^{style [fontsize=12pt]',col1a,'}');
```



CONCLUSION

SAS® provides us with many tools to customize ODS output. ODS ESCAPECHAR allows SAS® programmers greater control and flexibility in producing "camera ready" output from SAS® procedures, by enhancing titles and footnotes, ODS text fields, and procedural output.

REFERENCES

SAS® Online Documentation PC SAS V9.1
SAS® Online Documentation PC SAS V9.2

http://support.sas.com

Haworth, Lauren. 1999. *PROC TABULATEBy Example*. Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

SUPPORT.SAS.COM – the samples, FAQs and human beings behind the scene are the greatest!

Scott Huntley for his inspirational talk at SUGI 31 on the future of reporting in 9.2, and his patient, helpful advice on the use of ODS PDF.

Eric Gebhart, Vince DelGobbo and Chevell Parker for introducing me to in-line styles (or style overrides as Vince prefers) in HTML-based tagsets.

Sandy McNeill, Lauren Haworth and Ray Pass for introducing me to the use of in-line styles in PROC TABULATEand PROC REPORT .

Cynthia Zender for her wise counsel on the use of in-line style commands and ODS ESCAPECHAR.

Wayne Hester for his generous help with my problems with the RTF destination, and for coming up with really cool additions to the RTF toolset.

My colleague Christianna Williams, with whom I built the "real world" example pdfs.

I couldn't have done it without them! Please read their papers listed in "Recommended Reading" below, and/or attend any of their presentations.

RECOMMENDED READING

DelGobbo, Vince. "A Beginner's Guide to Incorporating SAS® Output in Microsoft® Office Applications." Proceedings of the Twenty-Eighth Annual SAS® Users Group International Conference. April 2003.

DelGobbo, Vince. "Creating *AND* Importing Multi-Sheet Excel Workbooks the Easy Way with SAS®." Proceedings of the Thirty-First Annual SAS® Users Group International Conference. March 2006.

Gebhart, Eric. "The Beginners Guide to ODS MARKUP: Don't Panic." Proceedings of the Thirty-First Annual SAS® Users Group International Conference. March 2006.

Gebhart, Eric. "Inline Formatting with ODS Markup." Proceedings of SAS® Global Forum 2009 Conference. March 2009.

Hester, Wayne. "Teaching your RTF Tagset to Do Clever Tricks." Proceedings of the Thirty-First Annual SAS® Users Group International Conference. March 2006.

Huntley, Scott. "Let the ODS Printer Statement Take Your Output into the Twenty-First Century." Proceedings of the Thirty-First Annual SAS® Users Group International Conference. March 2006.

Huntley, Scott. "How to Add a Little Spice to your PDF Output." Proceedings of SAS Global Forum 2008 Conference. March 2008.

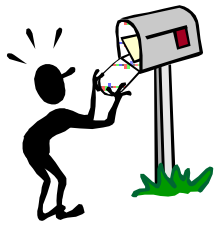
Parker, Chevell. "Generating Custom Excel Spreadsheets using ODS." Proceedings of the Twenty-Eighth Annual SAS® Users Group International Conference. April 2003.

Pass, Ray and McNeill, Sandy. "PROC REPORT : Doin' It in Style!" Proceedings of the Twenty-Eighth Annual SAS® Users Group International Conference. April 2003.

Pass, Ray and McNeill, Sandy. "PROC TABULATE: Doin' It in Style!" Proceedings of the Twenty-Ninth Annual SAS® Users Group International Conference. May 2004.

Zender, Cynthia. "Funny ^Stuff~ in My Code: Using ODS ESCAPECHAR." Proceedings of SAS Global Forum 2007 Conference. April 2007.

CONTACT INFORMATION



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

Louise Hadden
Abt Associates Inc.
55 Wheeler St.
Cambridge, MA 02138
(617) 349-2385 (work)
(617) 349-2675 (fax)
Email: louise_hadden@abtassoc.com

Sample code is available from the author upon request.