General Methods to Use Special Characters

Dennis Gianneschi, Amgen Inc., Thousand Oaks, CA

ABSTRACT

This paper presents three general methods to use special characters in SAS® procedure output as well as one RTF specific method. The methods combine features found in the Windows® Character Map Utility, ODS, RTF, and the SAS macro language. Key factors determining the effectiveness of a method are identified. Implementation details such as syntax differences in ODS RTF markup and software bugs in SAS versions 8.2 and 9.1 are discussed. One method uses a macro to encode the special character into a macro variable. Input parameters are unicode and macro variable name. The unicode is the 4-digit code found in the Character Map Utility. The macro variable is then placed in the SAS statement that targets an area of the output: title, footnote, or a column heading. Depending on character chosen, the global cannot be resolved in data displayed in the row, a bug in SAS prior to Version 9.2 deletes from display characters that follow. The following is an example macro call, title statement usage, log message, and output file.

SAS program code:

```
%special_char(unicode=00AE,name=registered_trademark);
title "Title with Registered Trademark &registered_trademark";
proc print;run;
```

Program execution log:

```
Note: special_char: registered_trademark = ->|®|<-
```

Output file display:

```
Title with Registered Trademark ®
```

INTRODUCTION

A special character is defined in the broadest sense as a character that is not seen on a standard computer keyboard. A special character can either display something or control the display of the characters that follow. Display examples are: a custom graphic A, the registered trademark sign ®, and a nonbreaking blank. Control examples are: new line, new page, and new font. A special character has the following processing life cycle:

- Coded into a program with a program editor.
- Copied or translated into an output file during program execution.
- Performs the intended function when the output file is viewed.

Several configuration factors determine choice of method to use a special character:

- Computing platform used, default font, and fonts available when building the program, executing the program, and displaying the output file.
- Editor used to build the program.
- Version of SAS.
- Area of output targeted: title, column heading, footnote, or data displayed as rows in the output.

All methods will be effective given the following specific set of configuration of factors. The SAS /Windows enhanced editor is used to build the program and code the special character where "True Type Courier" is the default font. The program executes on a Unix computing platform, using proc report and ODS to produce an RTF file, where "Arial" is the default font. The RTF output file is displayed in WORD® for Windows where "Arial" is the default font. The target areas for special character display are title, column heading, and footnote.

OVERVIEW OF METHODS

The following methods are presented and referred to by method number.

1. Select, copy, and paste
2. Input the 4-digit unicode into the %special_char macro
3. Use ODS text mark up commands
   a. Pure ODS
   b. RTF Based

The methods are ordered from least to most complex in data processing and user knowledge. Methods #1 and #2 are limited to displaying a single special character within the default character set. Method #3b is limited to the RTF output destination.

WINDOWS CHARACTER MAP UTILITY

This utility is provided free as part of the Windows operating system. The utility is used in all the methods and displays all characters in a particular font library along with their 4-digit unicode. To run the utility, click on the Windows **Start** button and make the following menu selections: **Programs, Accessories, System Tools, Character Map.** The utility comes up showing the default font of "Arial". Controls allow selection of font and browsing characters within font. Click on a specific character to select it and display the 4-digit unicode at the lower left. Click on the copy button to copy the selected character to the Windows clipboard.

METHOD #1. – SELECT, COPY, AND PASTE

From the Windows Character Map Utility mentioned above, select and copy the special character to the clipboard. Changing focus to your Windows editor, paste the special character into your program code as in the following example using ™ : SAS program code: `title 'Title with trademark symbol ™';` SAS executes, moving the special character to the title target area of the output file: `Title with trademark symbol ™`

Method #1 may require a Windows based editor like SAS/Windows enhanced editor or TextPad®. These editors support certain characters but not others. Immediate feedback is provided on which characters are supported. Unsupported characters are translated to a question mark (?) during the paste operation. Other method #1 advantages are fewest keystrokes and simplicity in the program execution phase of the processing life cycle, where the special character is simply moved from program to output file with no encoding or translation.

METHOD #2. – USE %SPECIAL_CHAR MACRO

Method #2 does not require that the editor support the special character of interest and works at execution time to encode the special character into a global macro variable. The global macro reference is then placed in the SAS statements where it is needed similar to Method #1. The %special_char macro has the following usage steps:

FIND THE 4-DIGIT UNICODE

a. Bring up the Windows Character Map Utility.
b. Select the font, usually the default font when viewing the output file.
c. Click on the character.
d. The 4-digit unicode displays at the bottom left of the window after the "U+". For example, click on the non breaking blank and "U+00A0" appears at the bottom left. Ignore the "U+" prefix, use "00A0" as input to the macro below.

USE %SPECIAL_CHAR MACRO

Copy and paste this SAS macro code into your program:

```
%macro special_char(unicode=, name=);
   %global &name;
   data _null_;
        A=input("&unicode."x,$UCS2B4.);
        call symput("&name.",trim(left(A)));
        stop;
   run;
   %put Note: special_char: &name = ->|&&&name.|<- ;
%mend;
```

Call the macro to encode the special character, inputting the unicode and a global macro variable name:

```
%special_char(unicode=00A0,name=non_breaking_blank);
%special_char(unicode=00AE,name=registered_trademark);
%special_char(unicode=00B1,name=plus_minus);
%special_char(unicode=2122,name=trademark);
```

Refer to the macro variable in the SAS statement that will target a certain area of output to display the special character:

```
title "Title with trademark symbol &trademark";
```

Check SAS log to confirm successful naming and encoding:

```
Note: special_char: non_breaking_blank = ->| |<-
Note: special_char: registered_trademark = ->|®|<-
Note: special_char: plus_minus = ->|±|<-
Note: special_char: trademark = ->|™|<-
```

Partial display of the SAS output file:

```
Title with trademark symbol ™
```

The effectiveness of method #2 depends on which unicodes the $UCS2B4. SAS supplied format supports. If a unicode is unsupported, the following message will be displayed and the global macro will be set to blank.
`NOTE: Invalid argument to function INPUT at line 1 column 25. A= _ERROR_=1 _N_=1`
Certain symbol characters such as ±,°,µ encoded using #2 will cause text display truncation when used in data displayed as rows in the body of an output. This is due to a bug in SAS versions prior to 9.2. Target areas such as titles, column labels, and footnotes do not hit this truncation bug.

METHOD #3. – USE ODS TEXT MARKUP COMMANDS

ODS text markup commands start with the escape character that the user defines:
ods escapechar = "^"; There are two types of ODS text markup commands:

a. Pure ODS commands are referenced by SAS ODS documentation and are supported all output destinations such as HTML, RTF, and PDF.
b. ODS RTF based commands are referenced mainly by Microsoft RTF documentation and are supported only in the RTF output destination.

METHOD #3.A. – PURE ODS

Use pure ODS commands to display a special character from a font other than the default. For example to print the ≥ symbol, use the following pure ODS text markup command: `^S={font_face=symbol}³^S={}`
When the output file is later viewed by the user, the appropriate ODS destination commands execute to switch the font to "Symbol", select the unicode for ≥ and then switch back to the default font. The unicode displays a superscripted 3 in the default font "Arial", but displays a ≥ when the font is switched to "Symbol". The following table provides examples of embedding pure ODS text mark up commands for displaying math symbols. The macro variable is used to hide the complexity of the ODS markup command when it is referenced in downstream parts of the program.

Method #3a – Pure ODS Commands [1]
Math Special Characters

SAS Code:	`%let GE = %nrbquote(^S={font_face=symbol}³^S={});`
Usage:	Footnote1 "P-value is &GE. 0.05 and two-sided.";
Display:	P-value is ≥ 0.05 and two-sided
SAS Code:	`%let LE = %nrbquote(^S={font_face=symbol}£^S={});`
Usage:	Footnote1 "P-value is &LE. 0.001 and two-sided.";
Display:	P-value is ≤ 0.001 and two-sided

[1] Do not use in data to display as rows in a table, data truncation may occur in SAS Versions prior to 9.2.

Method #3a works correctly for titles, footnotes, and column headings but hits the same display truncation bug as method #2 when the target area is data displayed in the row. The bug shows when displaying a "Symbol" font character that start with '0x' in the unicode. ODS apparently interprets the '0x' as an end of character string when displaying data in rows of an output. All data after the '0x' does not display even though it exists in the data. SAS technical support confirmed the bug and has fixed it in 9.2. The bug is not limited to the "Symbol" font and is character dependent.

Method #3a behavior appears to be ODS destination dependent. When testing under SAS 9.1.3. SP3, I got different results between RTF and HTML destinations using #3a methods in the column heading target area. The HTML destination blanked out columns from display but the RTF destination displayed all columns. The log message for one of the columns that did not display in the HTML table was:

```
WARNING: Data too long for column "ae ^S={font_face=symbol}Ô^S={} is
      ^S={font_face=symbol}³^S={} 0.05  body"; truncated to 80 characters to fit
```

The RTF column appeared correctly:

ae ™ is ≥ 0.05 body
BODY P-value™ is ≥ 0.05 unadjusted and ≤ 0.001 adjusted.

METHOD #3.B. – RTF BASED

RTF based markup commands can only be used only in the RTF output destination and differ in syntax between SAS Version 8 and 9, but not in functionality. All method #3b SAS coding examples will be clearly titled as Version 8 or 9 specific. SAS Institute provides a web page on RTF based escape codes for SAS Version 8: **Link.** Microsoft provides reference material for RTF, not SAS. Here is the RTF Version 1.6 website: **Link.** The following website is an additional reference for RTF codes and describes an older, backward compatible Version 1.5: **Link**. Try some feature in WORD, then save in RTF format. Use a text editor to view the RTF file to see the true RTF codes in order to get some sense of the RTF language. ODS translates the RTF based codes into true RTF codes during program execution and places the true RTF codes in the output file. The word processor reads the file and executes the RTF commands. ODS RTF based text mark up commands are needed in the following two general situations:

- Provide text formatting commands such as new line.
- Display superscripted or subscripted text.

PROVIDE TEXT FORMATTING COMMANDS SUCH AS NEW LINE

In the following title1 statement produce three title lines because of the embedded RTF based text markup commands.

SAS Code – Version 8 Specific

```
Title1 'Listing of All Subjects Experience Serious Adverse Events ^n During the
        First Phase of the Study ^n (Protocol XXX00100101)';
```

SAS Code – Version 9 Specific

```
Title1 'Listing of All Subjects Experience Serious Adverse Events ^\line During the
        First Phase of the Study ^\line (Protocol XXX00100101)';
```

The ^n in Version 8 and the ^\line in Version 9 are ODS RTF based commands that tells ODS to generate true RTF command in the output RTF file to make the word processor display three title lines.

DISPLAY SUPERSCRIPTED OR SUBSCRIPTED TEXT

The following example shows SAS code that embeds an ODS RTF based command in a SAS variable label with superscripted "a".

SAS Code – Version 8 Specific

```
data efficacy; set in201.efficacy;
     label trt_out = 'Treatment Outcome^{super a}';
run;
proc print label;run;
```

SAS Code – Version 9 Specific

```
data efficacy; set in201.efficacy;
     label trt_out = 'Treatment Outcome^\super a ^\nosupersub ';
run;
proc print label;run;
```

TRANSLATING A TRUE RTF CODE BACK TO A ODS RTF BASED CODE

SAS Version 8 Specific

The RTF based commands used in ODS after the escape character are based on true RTF commands. There is a three step process that will sometimes work in transforming a true RTF code into a SAS Version 8 ODS RTF based code. 1) Find the true RTF command. 2) Prefix the RTF command by the escape character (^). 3) Remove the backslash. Some native RTF command can be translated back to ODS RTF codes but other codes don't follow the rules. For example, ^{line} does not translate into {\line}, ^n is the ODS RTF code.

True RTF Code	SAS Version 8 ODS RTF Code
{\sub a}	^{sub a}
{\super a}	^{super a}
{\line}	^n

SAS Version 9 Specific

Version 9 support for RTF has more of a direct pass through nature but seems to use a different set of RTF commands that Version 8 did. The following steps will sometimes work in transforming a true RTF code into a SAS Version 9 ODS RTF based code. 1)Find the RTF command. 2) Prefix the RTF command by the escape character (^). For example, the following true RTF codes can be translated back to their corresponding SAS Version 9 ODS RTF code.

True RTF Code	SAS Version 9 ODS RTF Code
\super a \nosupersub	^\super a ^\nosupersub
\sub a \nosupersub	^\sub a ^\nosupersub
\line	^\line
\par	^\par

METHOD #3.B. – RTF BASED – EXAMPLES

Method #3b – RTF Based

SAS Version 8 Specific

ODS RTF Code	Display	Description / True RTF Code
A^{super a} A^{super 789}	A^{a} A^{789}	All letters, numbers, as well as other characters are available as a single or multiple superscript or subscripts. Version 8 seems to generate curly braced true RTF commands like {\super a}.
A^{sub a} A^{sub 789}	A_{a} A_{789}	
C^{sub max} (0-8hrs)	C_{max} (0-8hrs)	Maximum drug concentration during the first 8 hours after drug administration. {\sub max} (0-8hrs)
t^{sub ½} (XXX001)	$t_{1/2}$ (XXX001)	Half-Life of XXX001. {\sub ½} (XXX001).
^n		A text layout soft return line feed. {\line}

SAS Version 9 Specific

ODS RTF Code	Display	Description / True RTF Code
A^\super a ^\nosupersub A^\super 789 ^\nosupersub	A^{a} A^{789}	Version 9 seems to generate RTF commands without the curly brace like \super a \nosupersub..
A^\sub a ^\nosupersub A^\sub 789 ^\nosupersub	A_{a} A_{789}	
C^\sub max^\nosupersub (0-8hrs)	C_{max} (0-8hrs)	Maximum drug concentration during the first 8 hours after drug administration
T^\sub ½^\nosupersub (XXX001)	$t_{1/2}$ (XXX001)	Half-Life of XXX001. \sub ½ \nosupersub
^\line		Text layout soft return line feed. \line
^\par		Text layout hard return line feed. \par

Mixing Pure ODS and RTF Based [1]
Area Under the Concentration by Time Curve from 0 to Theoretical Infinity

SAS Version 8 Specific	
SAS Code:	`AUC^{sub (0-}^S={font_face=symbol}^{sub ¥}^S={}^{sub )}(XXX001)`
Display:	$AUC_{(0-\infty)}$ (XXX001)
SAS Version 9 Specific	
SAS Code:	`AUC^\sub (0-^S={font_face=symbol}¥^S={})^\nosupersub (XXX001)`
Display:	$AUC_{(0-\infty)}$ (XXX001)

[1] Do not use in data to display as rows in a table, data truncation may occur in SAS Versions prior to 9.2.

CONCLUSIONS

All methods make use of the Windows Character Map Utility. Each method has limitations based on configuration factors present during program development, execution, and output file viewing. Method #1 should be the first choice because of the following advantages: immediate feedback, works in all target areas, fewest keystrokes, and simplest processing. Method #2 is useful if no Windows editor is available and involves extra program development keystrokes to code the %special_char macro call and macro variable references. Method #2 has two processing steps during program execution: encode and then move the special character from the program file to the output file. Method #3a provides the ability to switch fonts but may not be reliable across ODS output destinations. Method #3b provides an RTF pass through capability in Version 9 for superscripting, subscripting, and text layout but requires RTF knowledge. In SAS Versions prior to 9.2, methods #2 and #3a are limited to a title, footnote, or column heading, due to a text display truncation bug. Not every character seen in the Windows Character Map Utility can be displayed in an output, no matter what method is employed, for example, Arial unicode 2206, the Greek capital letter delta, ∆ due to any one of a number of configuration factors.

REFERENCES

SAS Version 9 Help System

SAS Version 8 Online Help

SAS Institute Technical Support

http://support.sas.com/rnd/base/topics/expv8/inline82.html
SAS Institute's web page on RTF based escape codes for SAS Version 8.

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnrtfspec/html/rtfspec.asp Microsoft's reference material for RTF.

http://www.biblioscape.com/rtf15_spec.htm Additional reference for RTF codes.

TRADEMARKS

CONTACT INFORMATION

dgiannes@amgen.com
Dennis Gianneschi
Mail Stop 24-2-C
One Amgen Center Drive
Thousand Oaks CA 91320-1799