

Paper 055-31

## The Invisible Character Alt-255 – Hidden Dragon: Hiding And Aligning Text in SAS® Output

Eugene Tsykalov, Shi-Tao Yeh  
GlaxoSmithKline, King of Prussia, PA

### ABSTRACT

Use of a special Invisible Character in SAS code is described to remove sometimes unnecessary text and help in text aligning and positioning in SAS output. The magic sequence of keys Alt-255 typed at numeric keypad places an Invisible Character symbol into text. This character looks like a blank space in the program code and SAS output but is processed and printed by many programs as a valid text character.

Practical applications of Invisible Character Alt-255 – the Hidden Dragon which eats out undesirable text and guards SAS output text aligning are presented: removing unnecessary column names from Proc Report output, suppressing the name of a Pageby variable in page headers, aligning text in titles, footnotes and columns, preserving the original text layout in SAS ODS output in PDF format.

Also discussed are differences in displaying of the Invisible Character in Windows and UNIX based programs.

### INTRODUCTION

SAS has many powerful tools available to programmers which help to produce complicated and professional looking output reports. Good examples of such tools are Proc Report and SAS ODS. These tools have many options to customize output layout, content and take care of text aligning and positioning.

In some cases though SAS reporting tools may put unnecessary text into output or result in misaligned text due to internal algorithms such as processing of leading and trailing blanks and text justifying.

In this article we describe a simple and elegant approach which solves these problems and enhances the output of SAS reporting tools, such as Proc Report and SAS ODS for PDF files. We present a special symbol, Alt-255, which we named The Hidden Dragon for its Invisible, Text Hiding and Text Aligning Guardian character.

### CREATING INVISIBLE CHARACTER IN THE PROGRAM CODE

To create an Invisible Character in the program the following magic sequence should be applied:

1. Put cursor at a position in the text of a program where Invisible Character is to be placed.
2. Activate numeric keypad of the keyboard by clicking the key "NumLock" (keyboards usually have a light indicator for "NumLock" key which goes on at this point).
3. Press and hold "Alt" key and while holding it, type digit keys 255 at the numeric keypad.
4. Release the "Alt" key.
5. After releasing the "Alt" key a blank character is inserted into a text and cursor moves to the next position.
6. Click again the "NumLock" key to switch off the numeric keypad (light indicator goes off).

It is important to use a numeric keypad for typing the magic number 255. The magic will not work if using the regular digits keys. The inserted blank character looks exactly as a blank space. That is due to the way Windows based programs interpret this character – as an empty graphical symbol. Actually, using this Alt-N technique it is possible to insert in the text any letter and many graphical symbols. For example, combination Alt-65 inserts letter "A" and Alt-241 inserts symbol ±. In general, codes above 127 belong to extended ASCII codes and represent graphical symbols.

The Alt-255 is of special interest because it appears to be invisible, yet it is processed and printed as a valid text character by many Windows based programs such as Editors Word, Notepad UltraEdit, Adobe Acrobat and others. It should be noted that different operating systems can interpret and present the Alt-255 symbol in a different way. We will discuss it later how Alt-255 symbol is presented by non graphical text Editor VI, and text utilities "more" and "less" in UNIX.

## USING INVISIBLE CHARACTER WITH PROC REPORT

### SUPRESSING COLUMN NAMES

For the presentation purposes we will use the following simplified dataset TEST:


```
Obs  subjid  age  sex  trtgrp  refnum
1    0011   21  Female  1      1
2    0017   23  Male    2      2
3    0019   32  Male    1      3
4    0024   29  Female  2      4
5    0025   25  Male    2      5
```

To print this dataset we assign labels and use PROC REPORT:

```
data test ;
  set test ;
  label subjid = 'SUBJECT'
        sex    = 'GENDER'
        age    = 'AGE'
        trtgrp = 'TREATMENT'
        refnum = 'REF NUMBER' ;
run ;

proc report data=test headline headskip split='^' ;
  column refnum subjid sex age trtgrp ;
  define refnum / order width=7 left ;
  define subjid / order width=8 center ;
  define sex    / order width=10 ;
  define age    / display width=5 ;
  define trtgrp / display center spacing = 5 width=10 ;
run ;
```

which produces the following output:




REF	SUBJECT	GENDER	AGE	TREATMENT
1	0011	Female	21	1
2	0017	Male	23	2
3	0019	Male	32	1
4	0024	Female	29	2
5	0025	Male	25	2

**FIG 1. Output of Proc Report with variable labels used for column names**

If we want to suppress printing a label for the variable REFNUM we may assign an empty label to this variable:

```
data test ;
  set test ;
  label subjid = 'SUBJECT'
        sex    = 'GENDER'
        age    = 'AGE'
        trtgrp = 'TREATMENT'
        refnum = ' ' ;
run ;
```

In this case SAS will put a name of the variable to PROC REPORT output:




refnum	SUBJECT	GENDER	AGE	TREATMENT
1	0011	Female	21	1
2	0017	Male	23	2
3	0019	Male	32	1
4	0024	Female	29	2
5	0025	Male	25	2

**FIG 2. Output of Proc Report with variable name used as a column name**

Now we replace a blank character with Invisible Character Alt-255 in the label for variable REFNUM:

```
data test ;
  set test ;
  label subjid = 'SUBJECT'
        sex    = 'GENDER'
        age    = 'AGE'
        trtgrp = 'TREATMENT'
        refnum = ' ' ; /* Alt-255 */
run ;
```

This code looks exactly as previous one for label but this time it has a Hidden Dragon which devours the label for variable REFNUM in the SAS output of PROC REPORT:




	SUBJECT	GENDER	AGE	TREATMENT
1	0011	Female	21	1
2	0017	Male	23	2
3	0019	Male	32	1
4	0024	Female	29	2
5	0025	Male	25	2

**FIG 3. Output of Proc Report with column name suppressed by Hidden Dragon**

For comparison we provide here two other methods of removing column names in PROC REPORT:

#### Classic way:

```
proc report data=test headline headskip split='^' ;
  column refnum subjid sex age trtgrp ;
  define refnum / order width=7 left ' ' ;
  define subjid / order width=8 center 'SUBJECT' ;
  define sex    / order width=10 center 'GENDER' ;
  define age    / display width=5 center 'AGE' ;
  define trtgrp / display center spacing = 5 width=10 'TREATMENT' ;
run ;
```



In this case column names were assigned within PROC REPORT and blank space was used to suppress the column name for variable REFNUM.

### Extravagant way:

```
data test ;
  set test ;
  label subjid = 'SUBJECT'
        sex     = 'GENDER'
        age     = 'AGE'
        trtgrp  = 'TREATMENT'
        refnum  = '^' ;
run ;

proc report data=test headline headskip split='^' ;
  column refnum subjid sex age trtgrp ;
  define refnum / order width=7 left ;
  define subjid / order width=8 center ;
  define sex    / order width=10 ;
  define age    / display width=5 ;
  define trtgrp / display center spacing = 5 width=10 ;
run ;
```

In this code we assigned a '^' character as a label for variable REFNUM and then used this character as a split character in PROC REPORT. Well, there was nothing to split for SAS so it did the trick of suppressing the column name in the output.

Both the Classic and Extravagant ways produce the same result as shown in FIG 3, effectively suppressing the column name in the PROC REPORT output.

What is the advantage of our Esoteric way of using Hidden Dragon in SAS output?

We came to this method when we used a third party Reporting Toolkit and were stuck when trying to remove unnecessary column names from the output. The Toolkit used forced logic to assign column name as a label or variable name and thus the Classic and Extravagant ways were useless. Hidden Dragon worked like a charm.

Similar story happened with an older, very useful validated macro which somehow was removing from the output records with one particular character variable missing values. It turned out later that it was purposely designed this way. But we needed to report all records and did not have time to change the code and revalidate it. Well, we replaced missing character values with Invisible Character and it worked. In other words, Hidden Dragon may be very handy when using existing code which cannot be changed to make it work when suppressing unnecessary text in the SAS output or replacing missing characters with valid invisible text.

### SUPPRESSING THE NAME OF A PAGEBY VARIABLE

The following code illustrates the way to suppress a name of Pageby variable SEX in the PROC REPORT output.

```
data test ;
  set test ;
  label sex = '' ; /* Alt-255 */
run ;
proc sort data = test ;
  by sex ;
run ;
proc report data=test headline headskip split='^' ;
  column refnum subjid sex age trtgrp ;
  by sex ;
  define refnum / order width=7 left ;
  define subjid / order width=8 center ;
  define sex    / order width=10 ;
  define age    / display width=5 ;
  define trtgrp / display center spacing = 5 width=10 ;
run ;
```

It works the same way as in previous examples using Invisible Character as a label.

## USING INVISIBLE CHARACTER WITH SAS ODS OUTPUT

For archiving and write protected viewing purposes it could be desirable to convert existing SAS text reports to ODS destination PDF files. The example of clinical report with complicated text aligning is shown at FIG 4.

Page 1 of 5

```

Protocol: 999999
Population: All Treated Subjects
Table 99.1
Summary of Adverse Events By Toxicity Grade During Pre-Therapy Period
----- Dose = Dose 1 N = 67 -----
MedDRA System Organ Class Text    Grade 1    Grade 2    Toxicity
Preferred Term                      Grade 3    Grade 4    Grade UK
-----
TOTAL                               14 ( 21%)  6 (  9%)  1 (  1%)  0      0
Cardiac disorders
Atrial fibrillation                 0          0          1 (  1%)  0      0
1 (  1%)
Gastrointestinal disorders
Abdominal pain NOS                  1 (  1%)  1 (  1%)  0          0      0
Constipation                        1 (  1%)  0          0          0      0
Diarrhoea NOS                       1 (  1%)  0          0          0      0
Nausea                              2 (  3%)  1 (  1%)  0          0      0
General disorders and
administration site conditions
Asthenia                            1 (  1%)  1 (  1%)  0          0      0
Fatigue                             1 (  1%)  3 (  4%)  0          0      0
Malaise                             1 (  1%)  0          0          0      0
Oedema NOS                          0          1 (  1%)  0          0      0
Oedema peripheral                   1 (  1%)  0          0          0      0
Pyrexia                             1 (  1%)  0          0          0      0

USER ID: /product/group2/999999/code/tae04.sas 13DEC03 14:06
  
```

FIG 4. Example of Clinical Report with complicated text aligning design.

The challenge of this task is that SAS has its own internal rules of processing leading and trailing blanks when creating ODS output. The SAS system has three options to justify the character field: 1) center 2) left and 3) right justification. None of these options preserves the original layout when reading records from an original text file and writing them into ODS output file. The text misalignment during this process results in a “collapsed file” as shown at FIG 5.

Page 1 of 5

```

Protocol: 999999
Population: All Treated Subjects
Table 99.1
Summary of Adverse Events By Toxicity Grade During Pre-Therapy Period
----- Dose = Dose 1 N = 67 -----
Toxicity
MedDRA System Organ Class Text    Grade 1    Grade 2    Grade 3    Grade 4    Grade UK
Preferred Term
-----
TOTAL                               14 ( 21%)  6 (  9%)  1 (  1%)  0      0
Cardiac disorders
Atrial fibrillation                 0          0          1 (  1%)  0      0
1 (  1%)
Gastrointestinal disorders
Abdominal pain NOS                  1 (  1%)  1 (  1%)  0          0      0
Constipation                        1 (  1%)  0          0          0      0
Diarrhoea NOS                       1 (  1%)  0          0          0      0
Nausea                              2 (  3%)  1 (  1%)  0          0      0
General disorders and
administration site conditions
Asthenia                            1 (  1%)  1 (  1%)  0          0      0
Fatigue                             1 (  1%)  3 (  4%)  0          0      0
Malaise                             1 (  1%)  0          0          0      0
Oedema NOS                          0          1 (  1%)  0          0      0
Oedema peripheral                   1 (  1%)  0          0          0      0
Pyrexia                             1 (  1%)  0          0          0      0

USER ID: /product/group2/999999/code/tae04.sas 13DEC03 14:06
  
```

FIG 5. Example of “Collapsed File” with misaligned text after conversion to ODS PDF format.

We provide here the utility program which preserves the original text layout in SAS ODS output using the charm and power of Invisible Character Alt-255.

The program consists of several steps:

- 1) read the existing SAS text output file and save it as temporary ASCII file
- 2) read this temporary file with each observation as a character field with the full length of the output file
- 3) insert the Invisible Character Alt-255 at the first position of each line (see the arrow mark)
- 4) set up the ODS destination file margins
- 5) use the ODS facility to read in the temporary file and write out the ODS destination file

```
%macro convtext (tfilein=,dtype=,dfile=);
```

```
filename readin "&tfilein";
```

```
options ls=100 ps=60;
```

```
data _null_;
```

```
infile readin end=last missover;
```

```
file 'temp.dat';
```

```
input v1 $100. ;
```

```
put @1 ' ' @2 _infile_ @97 ' ' ; /* Alt-255 */
```

```
output;
```

```
run;
```



```
filename readina 'temp.dat';
```

```
data textfile;
```

```
infile readina ;
```

```
input @1 v1 $100. ;
```

```
output;
```

```
run;
```

```
proc template;
```

```
define style Styles.Fdamargn;
```

```
parent = styles.rtf;
```

```
replace Body from Document
```

```
"def margins to meet FDA Guidelines" /
```

```
pagebreakhtml = html('PageBreakLine')
```

```
leftmargin = 1in
```

```
rightmargin = 1in
```

```
topmargin = 1.05in
```

```
bottommargin = 1.09in;
```

```
style table from table /
```

```
cellpadding = 0pt;
```

```
end;
```

```

%global n ;
%if &dtype =rtf %then %do;
    %let n=12;
%end;
%else %if &dtype =pdf %then %do;
    %let n=12.5;
%end;

options nodate orientation=landscape;
ods &dtype file="&dfile" style=Styles.Fdamargn;
title;

proc report data=textfile nowd
    style={frame=void rules=none};
    column v1;
    define v1 /style(header)=
        {background=white foreground=white font_size=1pt cellheight=0 pt}
    style(column)=
        {font_face='Courier New' font_size=12pt font_weight=bold
        cellheight= &n pt cellpadding=0pt cellspacing=0pt};

run;
quit;

ods &dtype close;
%mend convtext;
run;

%convtext(tfilein=tae04.lst, dtype=pdf, dfile=tae04.pdf);
run;

```

The resulting output of this program is PDF file shown at FIG 4.

## DISPLAYING INVISIBLE CHARACTER IN UNIX

Everybody knows - not all Dragons are alike. The same is true for Hidden Dragon, The Invisible Character. Graphics based programs such as Word, Notepad, UltraEdit, Adobe Acrobat, etc. in Windows display the Alt-255 character as invisible. Other non graphics programs may show it in a strange "true" appearance. We provide here two examples of displaying the Hidden Dragon symbol in UNIX with never dying editor VI and UNIX text utilities "more" and "less".

?	SUBJECT	GENDER	AGE	TREATMENT
1	0011	Female	21	1
2	0017	Male	23	2
3	0019	Male	32	1
4	0024	Female	29	2
5	0025	Male	25	2

FIG 6. Appearance of the Invisible Character in UNIX text utilities “more” and “less”.

```
data _null_;
  infile readin end=last missover;
  file 'temp.dat';
  input v1 $100. ;
  put @1 '\040' @2 _infile_ @97 ' ' ; /* Alt-255 */
  output;
run;
```

FIG 7. Appearance of the Invisible Character in UNIX editor VI.

It is strikingly clear that non graphics programs in UNIX do not get along with Alt-255 symbols and show them as some visible, possibly Operating System specific characters.

## CONCLUSION

The techniques using Invisible Character Alt-255 - named by authors The Hidden Dragon - described in this paper help in enhancing SAS output. Practical code examples are provided which explain how to suppress unnecessary text in PROC REPORT output and align and preserve original layout when using SAS ODS output in PDF format. It is shown that the Invisible Character Alt-255 can be displayed differently in Windows and UNIX programs.

As with any Dragons, caution and appropriate use is advised.

## AKNOWLEDGEMENTS

Authors thank our colleague Chauthi Nguyen for pointing out to “split character” method and useful discussions.

## CONTACT INFORMATION

Eugene Tsykalov, Ph.D., Shi-Tao-Yeh, Ph.D.  
 GlaxoSmithKline  
 2301 Renaissance Blvd  
 RN0420, P.O.Box 61540  
 King of Prussia, PA 19406-2772  
 Tel: (610)787-3855  
 Fax: (610)787-7074  
 e-mails: [Eugene.2.tsykalov@gsk.com](mailto:Eugene.2.tsykalov@gsk.com), [Shi-Tao.Yeh@gsk.com](mailto:Shi-Tao.Yeh@gsk.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 registered trademarks or trademarks of their respective companies.