

**064-2007**  
**Using Invisible Characters in SAS® Programming.**  
**Hidden Dragons, Episode II**

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

### **ABSTRACT**

Use of special Invisible Characters in SAS code is described which helps to remove unnecessary text, align, position and paginate information in SAS textual and graphical output. This is a sequel to our previous paper describing magic sequence Alt-255 which places an Invisible Character – Hidden Dragon into text. This article adds two more useful Invisible Characters – Linefeed and Pagefeed – to the growing family of Hidden Dragons, The Invisibles.

Practical applications of Invisible Characters Alt-255, Linefeed and Pagefeed 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, formatting text in data tips box in interactive Java Applets and pagination of multiple concatenated reports.

Also discussed are differences in displaying of the Invisible Characters 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, SAS ODS for PDF files and interactive Java Applets.

We present special symbols, Alt-255, which we named The Hidden Dragon for its Invisible, Text Hiding and Text Aligning Guardian character as well as Linefeed and Pagefeed, a pair of Crouching Tigers which silently and invisibly help for creating user desirable SAS output.

### **CREATING INVISIBLE CHARACTERS IN THE PROGRAM CODE**

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

#### **Alt-255, Hidden Dragon:**

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

**Figure1. 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'
        refnum = '' ;
run ;
```

```
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

**Figure 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

**Figure 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 'GENDER' ;
  define age / display width=5 '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 Figure 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.

#### SUPRESSING 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 Figure 4.



```

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;

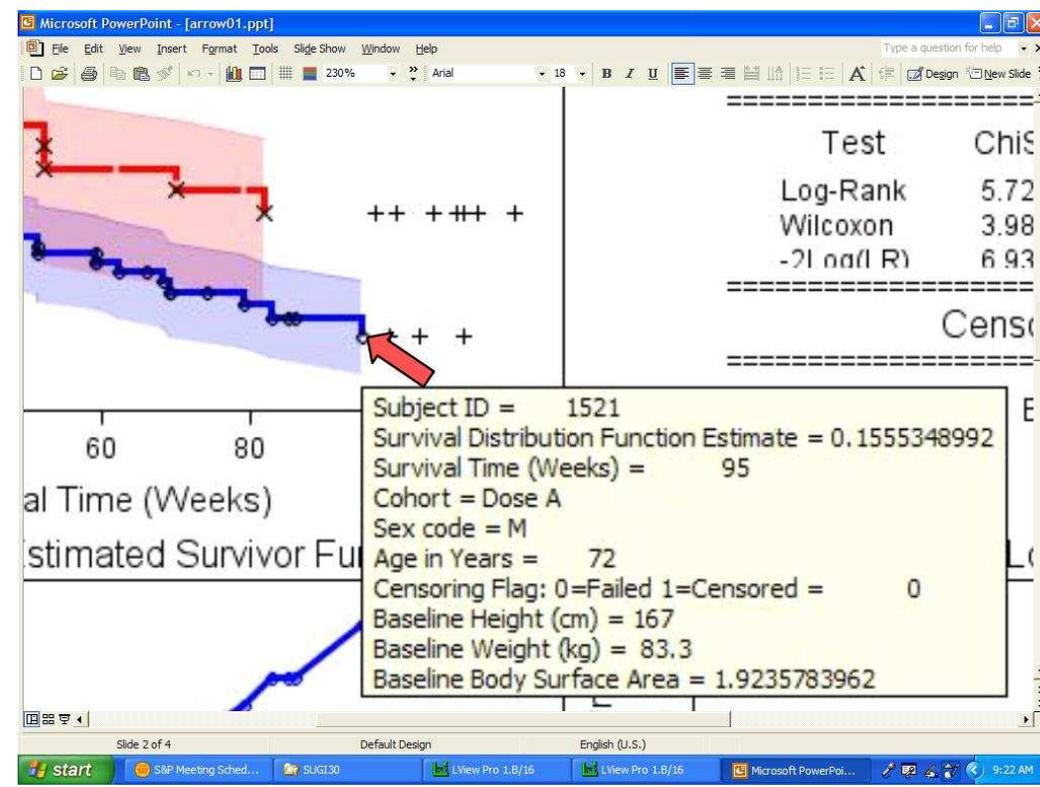
proc report data=textfile nowd
  style={frame=void rules=none};
  column v1;
  define v1 /style(header)= {background=white foreground=white font_size=1pt
  cellheight=0 pt}
  style(column)= {font_face='Courier New' font_size=12pt font_weight=bold
  cellheight= &n pt cellpadding=0pt cellspacing=0pt}; 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 Figure 4.

### USING INVISIBLE CHARACTER LINEFEED WITH SAS JAVA APPLETS

The following graph uses ODS output Tables from PROC LIFETEST as input for graphics presentation. The graphics are plots on 3 panels and statistical test and censor information are shows as text table format on one panel.



**Figure 6 Red Arrow Cursor Positioned Over a Graph Element and Tool Tips Feature with Drop Down Text Box in Interactive Java applet. The correct formatting of text required a use of Linefeed invisible character.**

The interactive hint which appears when user puts cursor over a particular point in the graph, shows additional information about the data. In order to format the text for the hint, the invisible character Linefeed was used as shown in the code below:

```
data final;
  merge pl final;
  by trt efsurv;
  length tip $1022.;
  tip='ALT="SubjID: ' || trim(left(subjid)) || '0D'x ||
    'Treatment: ' || trim(left(trtgrpt)) || '0D'x ||
    'Age: ' || trim(left(age)) || '0D'x ||
    'Sex: ' || trim(left(sexcd)) || '0D'x ||
    'Weight(kg): ' || trim(left(dmwt)) || '0D'x ||
    'Height(cm): ' || trim(left(dmht)) || '0D'x ||
    'Body surface area: ' || compress(round(dmbsa,.001)) || '0D'x || ' ' ;
run ;
```

Without using the Linefeed, the text for the hint would be a one long line. Using the Invisible Character Linefeed, we found a new use for the Java applet hint (which is supposed to be a short hint) – to show additional valuable data info, related to the data object.

### USING INVISIBLE CHARACTER PAGEFEED WHEN COMBINING MULTIPLE SAS OUTPUTS

In the following example we use the Invisible Character Pagefeed to concatenate multiple SAS output files into one long report. The Pagefeed character is used to vertically format and align the separate reports which otherwise would be mixed at the last pages of each report.

The following macro was used to add Pagefeed character between separate reports, using the function byte(12).

```

/*****
* MACRO PARAMETERS:
*
* dir: Directory where outputs reside.
*
* prmptcmd: This is the list of options to be added to the "dir"
*           statement in MS-DOS which will provide a list of
*           the desired files to standard output. These are
*           the files that will be concatenated together.
*
* catfile: Name of file to contain all of the concatenated files.
*
* pagefeed: Set to Y if a page feed is needed between files.
*****/
%macro concat2(dir=, prmptcmd=, catfile=, pagefeed=N);

filename filelist pipe "dir %cmpres(&dir)&prmptcmd";

/*****
* Delete external file with the same name as the concatenation
* file. (If it exists.)
*****/
filename tmp "%cmpres(&dir)%cmpres(&catfile)";
%let rc=%sysfunc(fdelete(tmp));
/*****
* If a page feed is requested between each output, create a
* file in the work directory that contains a single page feed.
*****/

%if %upcase(&pagefeed)=Y %then %do;
  data _null_;
    file "%sysfunc(pathname(work))\pagefeed.txt";
    pagefeed=byte(12);
    put @1 pagefeed;
  run;
%end;

```

## DISPLAYING INVISIBLE CHARACTERS IN UNIX

Everybody knows - not all Dragons and Tigers are alike. The same is true for Hidden Dragon, The Invisible Character and Crouching Tigers Linefeed and Pagefeed. Graphics based programs such as Word, Notepad, UltraEdit, Adobe Acrobat, etc. in Windows display these characters as invisible. Other non graphics programs may show it in a strange "true" appearance. We provide here examples of displaying the Hidden Dragons symbols 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 7. 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 8. Appearance of the Invisible Character in UNIX editor VI.**

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

## CONCLUSION

The techniques using Invisible Character Alt-255, The Hidden Dragon and Linefeed and Pagefeed, The Crouching Tigers, described in this paper to help in enhancing SAS output. Practical code examples are provided which explain how to suppress unnecessary text in PROC REPORT output, align and preserve original layout when using SAS ODS output in PDF format, formatting text in Java applets hints and concatenated multiple reports. It is shown that the Invisible Characters can be displayed differently in Windows and UNIX programs.

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

## ACKNOWLEDGEMENTS

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

## CONTACT INFORMATION

Shi-Tao-Yeh, Ph.D.  
 GlaxoSmithKline  
 2301 Renaissance Blvd  
 RN0420, P.O.Box 61540  
 King of Prussia, PA 19406-2772  
 Tel: (610)787-3856  
 Fax: (610)787-7074  
 e-mails: [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.