Abstract

These days, with the proliferation of graphical user interfaces, producing presentation-quality reports is becoming a required feature for any software tool. This paper will illustrate a technique for creating this type of report from SAS procedures by using the SAS macro language to insert text formatting codes into TITLE and FOOTNOTE statements. The code shown in the paper assumes a PCL™ 5-compatible printer such as the Hewlett-Packard LaserJet™ 4, but the method of inserting printer control codes should be the same no matter which printer is used. A technique for using FSFORMS to control the output on the printer will also be shown, as well as drawing attractive boxes around tabular output in both the OS/2® and Windows™ environments.

Overview

Historically, SAS has been an excellent product for analysing data and displaying that data in either columnar or tabular formats, but has been rather weak at "prettying-up" the reports. The reason for this is simple - SAS has no easy way to handle proportional fonts, which are required to do presentation-quality reports.

The main reason for the problem is the way SAS deals with the printer - whether you're using a SAS procedure or DATA step, the output controls are all based on the character position within the line. This works well if the font is a fixed-spaced one, where the inter-character spacing is the same whatever the character, but fails when using a proportional font, where different characters have different widths. Compare these two lines:

The quick brown fox jumped.
The quick brown fox jumped.

Proportional vs. Fixed Fonts

Notice that the first line, which is in a proportional font, is not as long as the second one, which is in a fixed-spaced font. Thus, it's usually possible to get more information into the same space if a proportional font is used.

Another example illustrating the problem that SAS users encounter when using proportional fonts is columnar output:

<table>
<thead>
<tr>
<th>XXX</th>
<th>YYY</th>
<th>&lt;10 spaces between.</th>
</tr>
</thead>
<tbody>
<tr>
<td>III</td>
<td>BBB</td>
<td>&lt;proportional font</td>
</tr>
</tbody>
</table>

XXX | YYY | <10 spaces here.
III | BBB | <fixed font
TTT | CCC |

The problem with columns

This example shows that using a proportional font makes it very difficult to line up columns by inserting blanks between them. This, however, is the way SAS does it, and thus there is a basic incompatibility between SAS and proportional fonts. To position data properly when using a proportional font, the position must be expressed in an absolute unit of measurement, usually inches.

In producing this document, much information was obtained from two Hewlett-Packard publications, PCL 5 Printer Language Technical Reference Manual[1], and PCL 5 Comparison Guide[2].

Font Basics

Some terms and definitions you’ll need to understand when working with fonts are.

TYPEFACE

A generic name for symbols (i.e., characters) having common design features. There are two main typeface families, serif and sans-serif. The difference between the two is the presence or absence of "tails" on the edges of the letters:

T - this is a serif font
T - this is a sans-serif font

Serif and sans-serif fonts

SYMBOL SET

A unique ordering of the characters in a font. Each symbol set is defined with a set of applications in mind. Thus, a symbol set may have box-drawing characters, or it may have foreign-language characters, or both. Symbol sets within ASCII and EBCDIC platforms can contain a maximum of 256 different characters.

STROKE WEIGHT

Describes the thickness of the strokes that make up characters. Bold is an example of a stroke weight.

STYLE

The angularity of the strokes of the characters with respect to the X-Axis. Italic is an example of a style.

PITCH

Describes the number of characters printed per horizontal inch. It is applicable only to fixed-spaced fonts.

POINT

A unit of measurement that equals \( \frac{1}{72} \) inch. Font height is measured in points, and horizontal movement within a line is measured in decipoints (\( \frac{1}{720} \) inch) on a PCL 5-compatible laser printer.

Laser Printer Basics

To be able effectively to take advantage of the capabilities of a laser printer, you need to understand how to control its functions. This entails some "bits-and-bytes" knowledge of the commands available, and how they interact with each other.

Before using the actual commands themselves, however, understanding the way the laser printer selects a font to be used is necessary. To select a font for printing, the printer needs to know the following information: symbol set, spacing, height, pitch, style, stroke weight, typeface, and orientation. Each of these characteristics has a default value that takes effect when the printer is initialized, either by powering it on, or by a software command.

Proportional Fonts with SAS® Software - Yes You Can!

John Laing, SAS Institute (Canada) Inc.  
<canw1@tor.sas.com>
The HP4 stores font information in a font select table. Whenever a font control command is received by the printer, this table is updated. When the time comes to print the text, the printer searches all available fonts and selects the one that most closely matches the select table characteristics.

The printer will also prioritize these characteristics as follows:

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbol Set</td>
<td>High</td>
</tr>
<tr>
<td>Spacing</td>
<td></td>
</tr>
<tr>
<td>Pitch</td>
<td></td>
</tr>
<tr>
<td>Height</td>
<td></td>
</tr>
<tr>
<td>Style</td>
<td></td>
</tr>
<tr>
<td>Stroke Weight</td>
<td></td>
</tr>
<tr>
<td>Typeface Family</td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td></td>
</tr>
<tr>
<td>Orientation</td>
<td>Low</td>
</tr>
</tbody>
</table>

Font selection priority

For the purposes of this paper, we will assume that the font is an internal one (Location), and will accept the default resolution.

When the printer selects a font, it first compares the highest-priority characteristic in the font select table to the corresponding characteristic for the available fonts. If only one font matches, it is selected. If multiple fonts match, the process is repeated for the next highest priority, and so on, until only one font remains. This one is then selected.

**Printer Commands**

To control the laser printer, you will need to know how to build a printer control command. The technique shown in this paper uses SAS macro variables to store these commands. Syntactically, all printer control commands start with the ESCAPE character, which is the 27th character in the ASCII collating sequence (when illustrating a printer command, the ESC character will be represented as a \( \text{ESC} \)). Thus, we can code a small DATA step to assign this character to a macro variable &esc:

```sas
data _null_
  call symputCesc', byte(27);  
run;
```

The printer commands commonly used will be described in the following section.

**RESET**

This is one of the simplest printer commands, but one that should be used at the beginning of each report. This command will change all the printer settings to their power-on status.

The syntax of the RESET command is

\[ \text{ESC} \]

To store this command in a SAS macro variable, use the following statement:

```sas
%let reset=&esc;
```

You need to ensure that the period (.) follows the &esc macro variable name, to separate it from the letter E following.

**SYMBOL SET**

Use this command to select the symbol set you wish to use. There are several available symbol sets, each of which should be illustrated in the Users Manual for your particular printer.

For the HP4, the default is Roman-8, which is inadequate if you wish to print box-drawing characters. I suggest using the PC-8 character set, which matches the character set found on PCs. Note that Windows users will need to set SAS up slightly differently to use box-drawing characters properly - more on this later! To create the command to use the PC-8 symbol set, use the following SAS statement:

```sas
%let pc8 = &esc.(10U; 
```

**SPACING**

This command controls whether your font is proportional or fixed. To create a macro containing the command to set fixed spacing, use this SAS statement:

```sas
%let fixed=&esc.(SP; 
```

For proportional spacing, use this statement:

```sas
%let prop=&esc.(SP; 
```

**PITCH**

This command controls the horizontal spacing for fixed fonts. It will be ignored when a proportional font is selected. To set a macro variable for this command, use this statement:

```sas
%let pitch12=&esc.(12H; 
```

The above command sets the pitch to 12. To set a different pitch (16.6 for example), simply change the 12 to the desired number:

```sas
%let pitch16=&esc.(16.6H; 
```

**HEIGHT**

This command controls the height of the font in points. It is ignored when a fixed font is selected. To set a macro variable for this command, use

```sas
%let h12=&esc.(12V; 
```

As with the PITCH command, the number can be changed to give a different height:

```sas
%let h16=&esc.(16V; 
```

**STYLE**

This command controls the angularity of the characters, as well as their structure. To set a macro variable which will turn on italics, use

```sas
%let italic=&esc.(I; 
```

To reset the printing back to normal characters, use

```sas
%let normal=&esc.(N; 
```

Styles are specified by changing the number. Available styles include these:

<table>
<thead>
<tr>
<th>Number</th>
<th>Style</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>upright</td>
</tr>
<tr>
<td>1</td>
<td>italic</td>
</tr>
<tr>
<td>4</td>
<td>condensed</td>
</tr>
<tr>
<td>5</td>
<td>condensed italic</td>
</tr>
<tr>
<td>8</td>
<td>compressed (extra condensed)</td>
</tr>
<tr>
<td>24</td>
<td>expanded</td>
</tr>
<tr>
<td>32</td>
<td>outline</td>
</tr>
<tr>
<td>64</td>
<td>inline</td>
</tr>
<tr>
<td>128</td>
<td>shadowed</td>
</tr>
<tr>
<td>160</td>
<td>outline shadowed</td>
</tr>
</tbody>
</table>

**STROKE WEIGHT**

This command controls the thickness of the characters. It is
usually used for emphasis. To set a SAS macro variable to turn
on bold printing, use

```sas
%let boldon=&esc.(3B);
```
To turn off bold printing, and resume normal stroke weight, use

```sas
%let boldoff=&esc.(0B);
```

Stroke weights fall within the range -7 to +7, with -7 being the
thinnest, and +7 the thickest.

**TYPEFACE**

This command controls the "look" of the font. To set a SAS
macro variable that will select the CG Times typeface, use

```sas
%let cgtimes=&esc.(4101T);
```

In the Hewlett-Packard publication PCL 5 Comparison
Guide[2], there is a complete listing of typeface families.

**Putting the Commands Together**

To use all these codes effectively, they can all be put together into
a file that can be included in your SAS program. PMACS.SAS is
an example of such a file. The first step in this file is a small
DATA step to create the macro variable &esc. The rest of the file
consists of %LET statements to assign the appropriate commands
to macro variables. Additional lines can be added to meet your
own reporting needs.

```sas
/* printer control strings for HP */
/*------------------------------------*/
data _null_;
call symput('esc',byte(27));un;
/* reset printer */
%let preset = &esc.(SIP ;
%let rur.;
%let ht8 = &esc.(s8V ;
%let ht10 = &esc.(s10V ;
%let pitch10 = &esc.(s10H ;
%let fixed = &esc.(sOP ;
%let pitch16 = &esc.(s16H ;
%let fontnorm = &esc.(s10H ;
%letUnivers = &esc.(s4148T ;
%let typeface = Letter Gothic /
%let gothic = &esc.(s4101T ;
%let univers = &esc.(s4148T ;
%let pitch10 = &esc.(s10H ;
%let stroke weight to normal */
%let boldoff = &esc.(0B ;
```

The PMACS.SAS include file

---

The **Horizontal Spacing Problem**

As noted previously, positioning data on the page must be done by
specifying a horizontal measurement expressed in inches, and not
in columns or character position. PCL 5 provides a command to
accomplish this:

```sas
<< #H
```

The desired position is substituted for the #, but the units must be
given in decipoints. Measurement for this distance should be made
from the left edge of the printable area on the page, which
normally starts .25" from the edge of the page. So, if measuring
from the edge of the page, make sure to deduct the .25".

To convert from inches to decipoints, the measurement must be
multiplied by 720. The SAS macro HPOS.SAS will create a
horizontal positioning command:

```sas
1 %macro hpos(pos);
2 %local L P INT DP INT DP posint decipt ;
3 %if &P " a %then %do;
4 %let L = %length(&pos);
5 %if &P = 0 %then
6 %let int = 41;
7 %let DP = 0;
8 %end;
9 %else %do;
10 %let DP = %eval{(%length{&pos}-41)};
11 %let int = %eval{(%length{&pos}-41)};
12 %end;
13 %let posint=%substr{&pos,1,41};
14 %if &DP > 0 %then
15 %let posint = %posint%substr{&pos,%eval{&DP+1}};
16 %end;
17 %let I = %eval{(%length{&posint})};
18 %let int = %eval{(%length{&posint})-DP};
19 %let decipt = %substr{&posint,1,41};
20 %if &DP > 0 %then
21 %let decipt = %substr{&posint,%eval{&DP+1}};
22 %let decipt = 2; %end;
23 %let decipt = &decipt..%substr{&I,1,41+1},&DP; %end;
24 %end;
25 %let decipt = &decipt;
26 %let symstr=%catal{&decipt,k} %end;
27 %end;
```

The HPOS.SAS macro.

**Notes:**

Line 1: the macro takes one parameter, the desired horizontal
position, in inches.

Lines 4-16: Since the SAS macro function %EVAL works only
with integer values, the pos parameter must be converted to an
integer. This section of code first calculates the number of decimal
places, and then puts the integer and fractional pieces back
atgether as one number(%point).

Line 17: Multiplies the calculated number by 720.

Lines 18-24: The resulting horizontal position is valid to two
decimal places. This code puts the number back together, keeping
up to two decimal positions.

Line 25: Removes any leading and/or trailing blanks.

Line 26: The returned printer command. Note that since the
command contains an &amp; the %NRSTR function must be used to
prevent attempted macro resolution.

**Using the Commands**

PFTEST.SAS is a small piece of code that illustrates how to set up
a report to use proportional fonts. An important thing to
remember is that these proportional fonts can only be used in
TITLES and in the DATA step. Printing from any SAS procedure is beyond the user's direct control, so the best that can be done is to use a "nice-looking" fixed font, such as Letter Gothic. The attached sample shows the output from PTEST.SAS.

PTEST.SAS

Notes:
1. The first two lines %INCLUDE the two files mentioned above. This will set up the macro variables containing the printer control codes, and allow access to the HPOS macro.

2. The OPTIONS statement sets up the environment. The LINESIZE= option needs to be set to a value higher than the normal 80 because we are adding the extra printer control codes to the line. This would throw off the centring that SAS normally does, so NOCENTER is used.

Getting the Output to the Printer

Before sending the document to the printer, we need to address some operating environment issues.

OS/2

In the OS/2 environment, we need to ensure that the printer will use the proper font and character set. To accomplish this, we need to do two things:

1) Bypass the OS/2 spooler, by either submitting

   OPTIONS sysprint='LPT1:';

which can be done in both the 6.08 and 6.10 releases, or by using the following methods in the SAS Display Manager System (release 6.08 only - there doesn't seem to be a way to do this in 6.10 display manager):

   Use the Printer Setup dialog:

2) Define and use a form with the FSFORMS command:

   We need to define a form so that the printer will use the PC8 character set. Not much sense in setting up the OS/2 environment properly if the printer is set differently!

   This can also reduce the code needed in the TITLE statement. In fact, a form is necessary before the LINESIZE= option can be fully implemented (the default form has a LINESIZE of 80). I have created a form I call LJ8511P to handle this. This form can be recreated by using these settings:

   Printer selection: Choose HP LaserJet(+)
   Text Body and Margin Information
   Characters per line: 130
   Lines on First Page: 58
   Lines on Following: 58
   Carriage Control Information
   Generate Carriage Control Information: No
   Print Control Information
   PRINT INIT
      -E
   PAGE 1
      -(10U
      -(s0P
      -(s12H
      -(s4102T
   PRINT TERM
      -E

   On initialization and termination, the printer is reset. Before page 1, the following characteristics are set: PC8 symbol set, fixed spacing, 12 characters per inch, Letter Gothic typeface.

   Assuming SASUSER.PROFILE is the location of the FORM entry we just created, we can use this options statement:

   OPTIONS forms=sasuser.profile.lj8511p.form;

   In release 6.08, an alternative to the OPTIONS statement is to use display manager. The FORM is set during the PRINT command:

1085
In release 6.10, the OPTIONS statement is necessary. The Printer Setup dialog is used only to indicate whether a form will be used:

Microsoft® Windows

Because Windows uses a different character set (ANSI) than either DOS or OS/2, printing box-drawing characters requires additional effort. Box-drawing characters are not part of the ANSI character set, so we need to use an alternate character set before they can be either seen on the screen, or printed. Fortunately, SAS provides a way to switch. For both release 6.08 AND release 6.10, there will be a set of lines in CONFIG.SAS looking like this:

```
/* This is the OEM character set */
/* -WINCHARSET OEM. */
/* FORMCHAR [-----]+[-----]-[/]<*/

/* This is the ANSI character set */
/* WINCHARSET ANSI */
/* FORMCHAR [-----]+[-----]-[/]<*/
```

Default CONFIG.SAS for Windows

These lines need to be changed to:

```
/* This is the OEM character set */
/* WINCHARSET OEM. */
/* FORMCHAR [-----]+[-----]-[/]<*/

/* This is the ANSI character set */
/* WINCHARSET ANSI */
/* FORMCHAR [-----]+[-----]-[/]<*/
```

Modified CONFIG.SAS for Windows

Once this modification is done, and the SAS System is restarted, the easiest way to test the change is to run a PROC TABULATE. This should result in a table having a box with solid lines.

With the OEM character set in effect, we now need to ensure that Windows doesn't get in the way when we send the output to the printer. As with OS/2, we need to do two things:

1) **Bypass the Windows Print Manager**, by either submitting

```
OPTIONS sysprint='LPT1:';
```

which can be done in both the 6.08 and 6.10 releases, or by using the following methods in display manager (release 6.08 only - there doesn't seem to be a way to do this in 6.10 display manager):

Use the **Printer Setup** dialog:

```
The Sysprint printer name: should be set to blank:
```

2) **Define and use a form with the FSFORMS command**:

This is done in the same fashion as in the OS/2 environment (see above).

**Conclusion**

Although it isn't a trivial process, proportional fonts, the key to presentation-quality reporting, can be used with SAS procedure output. By making use of the SAS macro facility, printer control codes can be inserted in the TITLE and FOOTNOTE statements, allowing complete tailoring of these report items. By specifying an appropriate fixed font such as Letter Gothic as the last code in the last TITLE statement, the body of the report can be made quite attractive as well. An additional consideration for presentation-quality reports is the use of attractive boxes for tabular output. The use of FSFORMS and the OEM character set in the Windows environment will allow these boxes to be created.
Trademarks

SAS is a registered trademark of SAS Institute, Inc. in the USA and other countries. IBM and OS/2 are registered trademarks or trademarks of International Business Machines Corporation. ® indicates USA registration.

PCL and LaserJet are registered trademarks of Hewlett-Packard Corporation.

Windows is a registered trademark of Microsoft Corporation.

References


## SAS System Options

<table>
<thead>
<tr>
<th>OPTNAME</th>
<th>SETTING</th>
<th>OPTDESC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATCH</td>
<td>NOBATCH</td>
<td>Use batch mode default system options</td>
</tr>
<tr>
<td>BUFNO</td>
<td>1</td>
<td>Number of buffers to use for each SAS data set</td>
</tr>
<tr>
<td>BUFSIZE</td>
<td>0</td>
<td>Size of internal I/O buffer</td>
</tr>
<tr>
<td>BYERR</td>
<td>BYERR</td>
<td>Null by list/stmt on <em>NULL</em> input data sets an error?</td>
</tr>
<tr>
<td>BYLINE</td>
<td>BYLINE</td>
<td>Print the by-line for at the beginning of each by-group?</td>
</tr>
<tr>
<td>CAPS</td>
<td>NOCAPS</td>
<td>Translate quoted strings and titles to upper case?</td>
</tr>
<tr>
<td>CARDIMAGE</td>
<td>NOCARDIMAGE</td>
<td>SAS source lines are processed as 80-byte cards</td>
</tr>
<tr>
<td>CATCACHE</td>
<td>0</td>
<td>Requested number of cached SAS catalogs</td>
</tr>
<tr>
<td>CBFIND</td>
<td>0</td>
<td>Number of buffers to use for each SAS catalog</td>
</tr>
<tr>
<td>CENTER</td>
<td>NOCENTER</td>
<td>Center SAS output?</td>
</tr>
<tr>
<td>CHARCODE</td>
<td>NOCHARCODE</td>
<td>Represent special symbols with alternate characters?</td>
</tr>
<tr>
<td>CLEANUP</td>
<td>CLEANUP</td>
<td>Try to recover from an out-of-resource condition?</td>
</tr>
<tr>
<td>CHEMAC</td>
<td>NOCHEMAC</td>
<td>Command line commands scanned for implicit invocation of 'cmd' macro?</td>
</tr>
<tr>
<td>COMPRESS</td>
<td>NO</td>
<td>Compress observations in created datasets?</td>
</tr>
<tr>
<td>CPUID</td>
<td>CPUID</td>
<td>Print CPU information at the top of log?</td>
</tr>
<tr>
<td>DATE</td>
<td>NODATE</td>
<td>Date printed in title?</td>
</tr>
<tr>
<td>DBCS</td>
<td>NODBCICS</td>
<td>Support Japanese, Chinese, Korean and other languages that use double-byte chara</td>
</tr>
<tr>
<td>DBCSLANG</td>
<td>PCIBM</td>
<td>Double byte character set language</td>
</tr>
<tr>
<td>DBCSTYPE</td>
<td>ENGLISH</td>
<td>Double byte character set collating sequence</td>
</tr>
<tr>
<td>DETAILS</td>
<td>NODetailS</td>
<td>Obtain and display additional information with directory lists?</td>
</tr>
<tr>
<td>DEVICE</td>
<td></td>
<td>Graphics device name</td>
</tr>
<tr>
<td>DFLANG</td>
<td>ENGLISH</td>
<td>Selects language for EUROPE date/time formats and informats.</td>
</tr>
<tr>
<td>DKRCOND</td>
<td>ERROR</td>
<td>Action for DROP/KEEP/RENAME error conditions on input data sets.</td>
</tr>
<tr>
<td>DKRCOND</td>
<td>WARN</td>
<td>Action for DROP/KEEP/RENAME error conditions on output data sets.</td>
</tr>
<tr>
<td>DMR</td>
<td>NODMR</td>
<td>Invoke a remote version of Display Manager for use with SAS/CONNECT?</td>
</tr>
<tr>
<td>DMS</td>
<td>DMS</td>
<td>Invoke the Display Manager?</td>
</tr>
<tr>
<td>DMSBATCH</td>
<td>NODMSBATCH</td>
<td>Use batch mode for DMS?</td>
</tr>
<tr>
<td>DSMFOR</td>
<td>DSMFOR</td>
<td>Treat data set not found as an error or set ds to <em>NULL</em>?</td>
</tr>
<tr>
<td>DSMAUTO</td>
<td>NODSMAUTO</td>
<td>Echo AUTOEXEC input to log?</td>
</tr>
<tr>
<td>ENGINE</td>
<td>V610</td>
<td>Default access method for SAS libraries</td>
</tr>
<tr>
<td>ERRABEND</td>
<td>NOERRABEND</td>
<td>Abend on error conditions?</td>
</tr>
<tr>
<td>ERRCHECK</td>
<td>NORMAL</td>
<td>Level of special error processing to be performed.</td>
</tr>
<tr>
<td>ERRS</td>
<td>20</td>
<td>Maximum number of observations with error messages</td>
</tr>
<tr>
<td>FIRSTOBS</td>
<td>1</td>
<td>First observation of each data set to be processed</td>
</tr>
<tr>
<td>FMTEERR</td>
<td>FMTEERR</td>
<td>Treat missing format or informat as an error?</td>
</tr>
<tr>
<td>FORMCHAR</td>
<td>1-51nH</td>
<td>[+-*/ -=*]</td>
</tr>
<tr>
<td>FORMDLM</td>
<td>130</td>
<td>Delimiter character for page separation</td>
</tr>
<tr>
<td>FORMS</td>
<td>SASUSER.PROFILE.LUB51P.FORM</td>
<td>Default form for PRINT command.</td>
</tr>
<tr>
<td>GWINOOW</td>
<td>GWINOOW</td>
<td>Graphics output put in a DMS window?</td>
</tr>
<tr>
<td>GWINOOW</td>
<td>NOGWINOOW</td>
<td>SAS statements scanned for implicit invocation of a 'STMT' macro?</td>
</tr>
<tr>
<td>IMPLMAC</td>
<td>NORMIMAC</td>
<td>Initial display manager command(s).</td>
</tr>
<tr>
<td>INITCMD</td>
<td></td>
<td>Initial statement(s) executed before SYSIN</td>
</tr>
<tr>
<td>INITSTMT</td>
<td></td>
<td>Missing value for invalid data</td>
</tr>
<tr>
<td>INVALIDDATA</td>
<td></td>
<td>Allow procedures to use variable labels?</td>
</tr>
<tr>
<td>LABEL</td>
<td>LABEL</td>
<td>Line size for printed output</td>
</tr>
<tr>
<td>LINESIZE</td>
<td>130</td>
<td>Perform macro processing?</td>
</tr>
<tr>
<td>MACRO</td>
<td>MACRO</td>
<td></td>
</tr>
</tbody>
</table>

Date: 01FEB95

Time: 08:24