Notes on the PostScript Text Formatter

by
Geoffrey Cooper

[Copyright (C) 1989, Geoffrey H. Cooper, All Rights Reserved.]

The “pstext” program is a text formatter that translates an input format into a PostScript(tm) program. The formatter is not significantly better in its capabilities than other formatters, and it is only targetted to PostScript. It is interesting in that it is implemented as a novel synergism between C and PostScript. The text formatter is implemented as a C front end that performs little besides parsing functions, and a PostScript macro package that implements all text formatting decisions. The C code is about 425 lines of code and the PostScript side is about 525 lines of PostScript; I believe you will be surprised by the amount of functionality that squeezes into that small a space.

The command line syntax is as follows:

pstext [-n | -t | -h -l] filename

Flags:

n — don’t prefix standard header.

t — define /TimeStamp as the current date and time.

h — put header on first page automatically (can be done in master)

l — turn off ligature processing

Normal Processing consists of three modes: text, verbatim, and passthru. In text mode, text is divided into words — strings of non-space/newline characters. Each word is converted to a PostScript syntax that the standard macro package formats and fills into paragraphs.

Within the text, octal numbers may be represented as \777 (three octal digits preceded by a backslash. The backslash character must be doubled to be put in the text. The sequence \fC where C is one of "RIBPFSL" causes the current font to change to Roman, Italic, Bold, Pbold+italic, Fixed width, Symbol or Last (the last font set is resumed). This is done by transforming \fC into "/C f" which has the appropriate interpretation in the standard macro package.

The text is also parsed for the following ligatures, listed in order of priority. The last is used to separate a ligature of higher priority, like this: fill this:

```
fi  -> \256
fl  -> \257
--  -> \320
‘‘  -> \252
’’  -> \272
,,  -> \271 (lower double close quotes)
... -> \274
<<  -> \253
```

```
>>  -> \273
\/ -> nothing
```

Note that any of these is also available by simply inserting the octal value in line. E.g., "\256ill" and "fill" result in the same PostScript.

A sequence of the form \(…) causes the contents of the parentheses to be inserted as passthru PostScript in line in context. It is entirely equivalent to the line:

’…

(see below), but may appear "prettier" to some. This construct may not be split across two lines, and the contents may not include parentheses.

Any line beginning with a period (.) or apostrophe (’) is processed as a command or a passthru. A period causes a side effect of a "break", where the text being filled is broken. An apostrophe has no such side effect.

A command line is processed as a "passthru", meaning that the contents of the line are simply emitted to the PostScript output after the first character (and an optional ’+’) is stripped off. If the second character in the line is the plus sign (+), verbatim mode is entered. In verbatim mode, each line is packaged with parentheses with "show NL", so that no special processing takes place. To achieve other effects, use passthru lines. \fC font switching is still active in this mode, as are \(…) passthru’s.

Verbatim mode is exited as it is entered, except that a minus sign is used instead of a plus sign. Plus signs may not currently be nested.

1. The Macro Package

The macro package determines the actual formatting of text. The following macros are used:

1.1. Variables

<<name>>	<<default>>	<<meaning>>
Size	10	text size
Leading	12	
Justify	true	adjust right margins
SpaceMax	10	max space before letterse
TopMargin	10 inch - Leading	
BottomMargin	1 inch	
LeftMargin	1 inch	
LineWidth	6.5 inch	
PageNum	1 (initial)	number that "H" uses
curfam	dict	contains values of /C...

1.2. Macros

```
inch                    - { 72 mul }
```

```
fontfamily F                - Change font families

  fontfamily = /Times, /Helvetica, /Courier, /Palatino, /Lucida,
  /LucidaSans, /LucidaSansTypewriter = /LucidaMono, /NewCenturySchlbk
  = /NewCenturySchoolbook = /Century, /AvantGarde, /LubalinGraph, /Bookman,
  /Garamond, /Optima, /Symbol

  For Symbol, R= Symbol, I= ZapfChancery, B= ZapfDingbats, P= Times-Italic.

[/P /B /I /R /F /S /L] f    - change face

number /FontSize            - Change size of fonts

(string)Center              - center string on the current line

NL                          - force newline.

HNL                         - half newline

numberoflines QNL           - quantified newline.

NP                          - new page

BK                          - break

(string)W                   - add a word to the current line.

number Tab                  - execute tab stop based on tabs every
                              "number" points.

number TabTo                - move to particular horiz. position

number Indent               - begin indenting by argument (may be nested)
Undent                      - undo last indent

number Squeeze              - indent both left and right margins
 ...
Unsqueeze

BeginEnum                   - enumerated lists (may be nested)
Enum
 ...
Enum
 ...*
```

```
EndEnum

Item                        - bulleted paragraph at current indent.
                              Bullet is at last indent, text at
                              this one.

(string)Descr               - Like item, but use "string" as bullet.

(string)Sect                - Begin a new section, "string" is heading.

SectPush                    - push section numbering level
SectPop                     - inverse of above.

H                           - standard header procedure (may be redefined).
```

The strings /{Header,Footer}{Left,Right,Center} define the header to be generated. A tilde is transformed into the current page number. The strings may be zero length. The default is to center the page number in em-dashes in the header. If HeaderRight is zero length and TimeStamp is defined, the HeaderRight is set to be TimeStamp (see the -t option, above). To just inhibit this behavior , set HeaderRight to be a space character.

1.3. Other variables

```
<<var name>>                <<default>>
PageNumberOnFirstPage       false
DuplexHeader                false
HeaderSize                  <inital_value_of_Size>-2  (header text's size
HeaderFontName              Times-Roman
HeaderLeftMargin            72
HeaderLineWidth             6.5 inch
HeaderYPos                  10.5 inch
FooterYPos                  0.5 inch
```