NAME

   psposter – create simple one-page colored posters in PostScript

SYNOPSIS

   psposter [+**author**] [+**bd** *color*] [+**bg** *color*] [+**bm** *dimension*] [+**bs** *factor*] [+**bw** *dimension*]
           [+**c** *nnn*] [+**copyright**] [+**d**] [+**e** *encoding*] [+**fg** *color*] [+**fn** *fontname*] [+**h** *dimension*]
           [+**help**] [+**l**] [+**mw** *dimension*] [+**p** *dimension*] [+**paper** *form*] [+**r** *angle*] [+**s** *factor*]
           [+**source**] [+**table** *{ 8/o | 8/h | 16/o | 16/h }*] [+**version**] [+**w** *dimension*]
           [+**x** *dimension*] [+**y** *dimension*] [**'?'**] [++]
           [word(s) or <*infile*] >*Encapsulated-POSTSCRIPT-file*

DESCRIPTION

   psposter creates an Encapsulated POSTSCRIPT file containing a one-page poster. It may be included as a figure in any document processing system that supports inclusion of POSTSCRIPT figures.

   The poster text is typeset inside a colored box of specified width and height, with an empty surrounding margin, a colored border, and an inner border margin.

   The text area has a width of *(boxwidth - 2*marginwidth - 2*borderwidth - 2*bordermargin)*, and a height of *(boxheight - 2*marginwidth - 2*borderwidth - 2*bordermargin)*.

   When landscape orientation is chosen, poster text will be rotated 90 degrees counterclockwise, so that it runs from bottom to top. The width and height of the poster box are *not* affected by this.

   The poster text is assumed to be encoded in Adobe Standard Encoding. The ASCII encoding used by most operating systems and file systems occupies the first 128 characters of this encoding.

   Because the characters in positions 129 . . . 255 in Adobe Standard Encoding do not correspond to more commonly-encountered encodings, **psposter** allows you to select alternate encodings, such as ISO-8859-1, also known as Latin-1.

   The ISO-8859-1 character set handles languages of Western Europe (Danish, Dutch, English, Faeroese, Finnish, Flemish, French, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish, and Swedish). Many other languages can be written in this character set, including Hawaiian, Indonesian, and Swahili.

   Use the +**e** *encoding* and +**table** options described below to produce a font table from which the octal values of any character in a particular encoding can be easily determined.

   If your keyboard or text editor do not support entry of data in extended character sets, you can represent such characters by a backslash followed by the three octal digits defining their position in the character set. For example, \347 represents the character 'c-cedilla' in ISO-8859-1 encoding. If you supply such data on the command line, you must remember to quote it (on UNIX, with apostrophes) to protect the backslash from shell interpretation.

OPTIONS

   Most options require values, which are supplied as the *next* command-line argument.

   Options are processed in order, so later ones can override earlier ones.

   The current implementation (in awk) requires options to begin with a '+', rather than the conventional '−', so as to distinguish them from **nawk**(1) options.

   Color values can be specified by red/green/blue fractional decimal values in 0 . . 1 separated by characters other than digits or period, optionally prefixed by *rgbi:*, or by X Window System named colors, or by X Window System hexadecimal values. The latter take either of two general forms: *#<red><green><blue>* or *rgb:<red>/<green>/<blue>* where the angle-bracketed colors are hexadecimal strings of characters from *0123456789abcdef* or *0123456789ABCDEF* of the form *h* (4 bits), *hh* (8 bits), *hhh* (12 bits), and *hhhh* (16 bits). The number of hexadecimal digits specified determines the divisor used to reduce the integer value to the range 0 . . 1: *f=15* (4 bits), *ff=255* (8 bits), *fff=4096* (12 bits), and *ffff=65535* (16 bits), For example, *green*, *#0f0*, *#00ff00*, *#000fff00*, *#0000ffff0000*, *rgb:0/f/0*, *rgb:0/ff/0*, *rgb:0/fff/0*, *rgb:0/ffff/0*, *rgbi:0/1/0*, and *0/1/0* all represent green, and *gray50*, *grey50*, *#888*, *#808080*, *#800800800*, *#800080008000*, *rgb:8/8/8*, *rgb:80/80/80*, *rgb:800/800/800*, *rgb:8000/8000/8000*, *rgbi:0.5/0.5/0.5*, and *0.5:0.5:0.5* all represent 50% gray.

Other X Window System color space prefixes (CIEXYZ, CIEuvY, CIExyY, CIELab, CIELuv, and TekHVC) are *not* recognized.

No gamma correction is applied to any color value.

Letter case and spaces in color names is *not* significant, although names with spaces will require shell quoting or backslashing.

Dimensions consist of a (possibly signed) number followed by a TEX dimension unit: *bp* (big points), *cc* (ciceros), *cm* (centimeters), *dd* (didot points), *in* (inches), *mm* (millimeters), *pc* (picas), *pt* (points), or *sp* (scaled points). E.g., 72bp, 2.54cm, 67.542dd, 1in, 25.4mm, 6.023pc, 72.27pt, and 4 736 286.72sp all specify a dimension of one inch. POSTSCRIPT uses *big points* as its default units, and its documentation refers to them as points.

Poster text is given either as separate arguments on the command line, or as separate lines of a file supplied on standard input. Multiple-word arguments must be quoted or their spaces escaped to be properly recognized.

+author	Display an author credit on *stderr* and quit with a success return code (on UNIX, 0).
+bd *color*	Define border color. *Default: 0.5:0.5:0.5 (50% gray).*
+bg *color*	Define background color. *Default: 0.9:0.9:0.9 (10% gray).*
+bm *dimension*	Define border margin, which is additional space between the text frame and the colored border. *Default: 0bp.*
+bs *factor*	Set the baseline skip to *factor* times the font pointsize. This value defines the interline spacing, or *leading*, as it is called in typography. *Default: 1.2.*
+bw *dimension*	Define border width. *Default: 10bp.*
+c *nnn*	Define number of copies to print. *Default: 1.*
+copyright	Display copyright information on *stderr* and quit with a success return code (on UNIX, 0).
+d	Display the font size on a second output page. This option is useful for making a series of posters with the same font size: specify **+d** for the poster with the longest text lines, then use the output font size in an explicit **+p** option for all of the other posters. This round-about way is required because automatic text sizing happens inside the POSTSCRIPT output device, rather than in **psposter**.
+e *encoding*	Specify a character-set encoding for the font.

An encoding is specified by either an absolute or relative file name, or the base name of a file (with *.enc* suffixed). In the latter case, the file is searched for in the *AFM-PATH* search path.

POSTSCRIPT fonts contain definitions of one or more characters, each with a symbolic name. Those names are then assembled into an *encoding vector* that is used to map an input character number to a particular font character.

Each POSTSCRIPT font contains a default encoding vector, in most cases, one conforming to Adobe Standard Encoding, which contains the standard ASCII characters in the lower 128 slots, and then assorted accented letters and special symbols in the upper 128.

It is not necessary for all defined characters to be present in the encoding vector, and indeed, some POSTSCRIPT fonts contain more than 256 characters, which means that only a subset of at most 256 characters can be used from the font at any given time.

The default encoding can be determined either by searching a POSTSCRIPT font (ASCII or binary) (*.pfa* or *.pfb*) file for lines defining the encoding, often

```
/Encoding StandardEncoding def
```

or by examining the corresponding Adobe font metric (*.afm*) file for character entries like this:

```
EncodingScheme AdobeStandardEncoding
...
C 251 ; WX 662 ; N germandbls  ; B 29 -12 647 742 ;
C  -1 ; WX 402 ; N onesuperior ; B 71 272 324 680 ;
```

The second item on each such character line is either the character's position in the font's default encoding vector, or -1 if it has no assigned default position.

The encoding file specified by the **+e** option may contain blank or empty lines, optional comments from percent to end of line, and an encoding vector that looks something like this:

```
/ISOLatin1Encoding [
  /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
  ...
  /.notdef /.notdef /space /exclam /quotedbl /numbersign
  ...
  /question /at /A /B /C /D /E /F /G /H /I /J /K /L /M
  /ucircumflex /udieresis /yacute /thorn /ydieresis
] def
```

There must be exactly 256 entries in the encoding vector; slots that have no assigned character are filled with the special name `/.notdef`.

The name of the encoding vector is arbitrary, but is usually chosen to name the encoding, then given a suffix of *-Encoding*.

Encoding files in this format are widely used in TEXware.

psposter assumes Adobe Standard Encoding by default, but is installed with at least these additional encodings: *ASCII*, *CP1250*, *DOS437*, *DOS850*, *Expert*, *HPRoman*, *IBM-CP437*, *IBM-CP850*, *IBMPC*, *ISO-8859-1* (== *Latin-1*), *ISO-8859-2* (== *Latin-2*), *ISO-8859-3* (== *Latin-3*), *ISO-8859-4* (== *Latin-4*), *ISO-8859-5*, *ISO-8859-6*, *ISO-8859-7*, *ISO-8859-8*, *ISO-8859-9* (== *Latin-5*), *ISO-8859-10* (== *Latin-6*), *ISO-8859-11*, *ISO-8859-12*, *ISO-8859-13* (== *Latin-7*), *ISO-8859-14* (== *Latin-8*), *ISO-8859-15*, *KOI8*, *MS-CP1250*, *MacExpert*, *Macintosh*, *NextStep*, *PDF-Doc*, *Standard*, *Symbol*, *Windows*, and *ZapfDingbats*.

Each encoding is defined in a separate file of the same basename, suffixed with *.enc*, and stored in the shared directory where other **psposter** files are located (see the variable *SHRLIBDIR* in the **psposter** script; it appears to have the value @shrlibdir@ at your site).

Few fonts have the necessary character repertoire to support all of these encodings, so you may need to print a font table (with this **+e** *encoding* option and the **+table** option described below) to find out what characters are missing. POSTSCRIPT printers will silently replace undefined characters in a font encoding vector by the character *.notdef*, which, in Adobe fonts at least, functions like a space. Some POSTSCRIPT previewers, however, may give you helpful diagnostics about missing characters, like these from **ghostscript**(1):

```
    Substituting .notdef for Aogonek
    Substituting .notdef for Lcaron
    ...
```

Default: Standard.

+fg *color*   Define foreground color. *Default: 0:0:0 (black).*

+**fn** *fontname*   Select the POSTSCRIPT font. The name can be either the long name of a Type 1 POSTSCRIPT font, such as *NewCenturySchlbk-BoldItalic*, or the basename of a POSTSCRIPT font in ASCII (*.pfa*), or binary (*.pfb*) format, such as *NewCenBI*. **psposter** will also handle compressed variants of these files from **compress**(1) and **gzip**(1) (*.pfa.Z*, *.pfb.Z*, *.pfa.gz*, and *.pfb.gz*), decompressing them on-the-fly. *Default: Times-Bold.*

Most POSTSCRIPT printers contain a number of resident fonts, but only a small number of downloadable Type 1 POSTSCRIPT fonts are available without a font license, notably Adobe Courier and Utopia, Bitstream Charter and Courier, and URW Antiqua, Grotesk, and Nimbus. These three vendors have generously made these fonts available for free distribution, but retain the copyrights. More than 2400 fonts are listed in the *afmfonts.map* file (see the **ENVIRONMENT VARIABLES** section below), but only a very small number are likely to be available at a given site.

On a UNIX system, you can list the downloadable fonts available at your site by
`grep /FontName /usr/local/share/lib/afm/*.pfa`,
assuming that
`/usr/local/share/lib/afm`
is where they reside.

For a list of printer resident fonts, consult your printer documentation, or print the small POSTSCRIPT file, *fontlist.ps*, that is distributed with **psposter**. For convenience, that file is reproduced here; you can adjust *PointSize*, *ColumnWidth*, and the default display font (*Courier*) if you wish:

```
%!
% fontlist.ps
% Send this file to a PostScript device to list
% the names of all of the resident fonts.
% [13-Dec-1994]
/inch { 72 mul } def
/PointSize 10 def
/ColumnWidth 3 inch def
/Courier findfont PointSize scalefont setfont
/n 0 def
/newcolumn
{
    /x x ColumnWidth add def
    x y moveto
    8 inch x gt { showpage newpage } if
} def
/newpage
{
    /x 1 inch def
    /y 10 inch def
    x y moveto
} def
/nstr 10 string def
/scratch 256 string def
newpage
(*)
{
    /n n 1 add def
    gsave
        n nstr cvs show
        (: ) show
```

```
                              show
                        grestore
                        0 PointSize -1.2 mul rmoveto
                        currentpoint exch pop 1 inch lt
                        {
                              newcolumn
                        } if
                  }
                  scratch /Font resourceforall
                  showpage
```

+**h** *dimension*   Define poster height. *Default: paper height.*

+**help**   Display a help message on *stderr* and quit with a success return code (on UNIX, 0).

+**l**   Use landscape (instead of portrait) orientation for text. Text is rotated 90 degrees counterclockwise, so that it runs from bottom to top. This is a shorthand for the rotation option +**r 90** (see below). Only text, *not* the poster box, is affected by this option.

+**mw** *dimension*   Define margin width. Most POSTSCRIPT hardcopy output devices are incapable of imaging closer than about 5mm to the edge of the display surface, so a non-zero margin width is usually desirable. *Default: 1in (or 25mm for ISO paper sizes).*

+**p** *dimension*   Set font pointsize. Normally, the size is chosen to make the poster text just fill the text frame, either horizontally or vertically. You can use this option to force a particular font size, e.g., to produce a series of posters with different text, all at the same size.

                  Any poster text outside the text frame is clipped, and not displayed. *Default: automatically set.*

+**paper** *form*   Set paper size. See **lptops**(1) for a list of supported sizes.

                  In addition to the standard sizes, any particular size can be requested by giving it in the form *widthxheight*, e.g., 8.5inx11in or 210mmx297mm.

                  Selecting a landscape-format (width greater than height) paper size does *not* affect poster text orientation.

                  If a paper size is too small to contain the default or user-specified page margin, border frame, border margin, and poster text, **psposter** issues a "tight-fit" warning, and eliminates the margins and frame in order from outside to inside to try to fit the poster text in the paper size.

                  If there is still insufficient space for poster text, then **psposter** terminates execution immediately with a failure exit code (on UNIX, 1).

                  Default: installation dependent: usually A (in the USA) and A4 (elsewhere).

+**r** *angle*   Set the text rotation angle, which must be a multiple of 90 degrees. It is measured counterclockwise from the horizontal axis. The +**l** landscape option is equivalent to +**r 90**. *Default: 0.*

+**s** *factor*   Define poster scale factor. This option can be used to resize a poster without having to change any other command-line options. *Default: 1.*

+**source**   Display software source archive information on *stderr* and quit with a success return code (on UNIX, 0).

+**table** *{ 8/o | 8/h | 16/o | 16/h }*
                  When this option is given, command-line poster words are ignored, and standard input is not read. Instead, **psposter** makes a table of all of the characters in the poster font. The option value defines the number of columns (8 or 16) and the label number base (octal or hexadecimal). The delimiter character '/' can be any punctuation

character. With 16-column tables, you may also want to specify the landscape option **+l**.

The table title includes the font size and name, and the font encoding.

The title, and the row/column labels, are always set in black Times-BoldItalic text.

Unless an explicit **+p** *dimension* option is provided, the font size is automatically adjusted to an integer value that maximizes table size, while remaining within the poster text area.

Default: no font table.

+version   Display the program version number on *stderr* and quit with a success return code (on UNIX, 0).

+w *dimension*   Define poster width. *Default: paper width.*

+x *dimension*   Define horizontal offset. A positive value moves the poster right on the output surface, and is *unaffected* by any poster scale factor, or by the choice of rotated text. *Default: 0bp.*

+y *dimension*   Define vertical offset. A positive value moves the poster up on the output surface, and is *unaffected* by any poster scale factor, or by the choice of rotated text. *Default: 0bp.*

?   Display a help message on *stderr* and quit with a success return code (on UNIX, 0).

++   No more options: remaining words are for poster text. This option is only needed if any of the poster words begin with a '+'.

EXAMPLES

Print a full-size poster with an announcement:
```
psposter Chemistry 101 Examination 1:00pm--3:00pm | lpr
```

Make a door sign in landscape mode:
```
psposter +l +w 10cm +h 25cm 'Prof. Jane Smith' >doorsign.eps
```

Print the same sign, in color:
```
psposter +l +w 10cm +h 25cm +bd blue +bg skyblue \
         +fg navyblue 'Prof. Jane Smith' | lpr
```
If a POSTSCRIPT printer is incapable of printing in color, it will simulate color by shades of gray.

Make a sale sign:
```
psposter +bg yellow <<END-OF-DATA >yardsale.eps
Terrific
Yard Sale
Today
123 Elm St.
END-OF-DATA
```

Make a seminar announcement:
```
psposter +bg orange <<EOF >seminar.eps
Biology Colloquium
Friday 23-Sep-1994
3pm--4pm
LSB 234

Dr. Judith Brown
Sex Life of Gnats
EOF
```

Make a Danish road sign (the octal escape sequences select the letters AE and o-slash; see Appendix E of the *POSTSCRIPT Language Reference*, third edition, Addison-Wesley, 1999, ISBN 0-201-37922-8):
```
psposter +bg yellow +fg black '\341resk\371bing' 5km >vejskilt.eps
```

Make another road sign in a downloaded font, supplying one leading and trailing blank on each line:

```
        psposter +bg green +fg white +fn URWGroteskT-Bold +r -90 \
                ' Green River ' ' 106 miles ' >roadsign.eps
```

(The stretch of highway I-70 in Utah from Salina to Green River is claimed to be the longest in the US Interstate Highway System without services.)

Make a poster showing the dingbat font layout:

```
        psposter +fn ZapfDingbats +table 8/o >dingbats.eps
```

ENVIRONMENT VARIABLES

AFMPATH   Search path for POSTSCRIPT font mapping file, (*afmfonts.map*), and for downloadable fonts in POSTSCRIPT ASCII (*.pfa*) and binary (*.pfb*) formats. The font mapping file connects long POSTSCRIPT font names with (usually shorter) file names on the host computer.

On UNIX, the search path is a colon-separated list of directories. On other operating systems, the search path follows the conventions of those systems.

RGBPATH   Search path for the X Window System color definition file, *rgb.txt*.

FILES

afmfonts.map   POSTSCRIPT font mapping file. It is distributed, together with some freely-distributable Type 1 fonts in *.pfa* and *.pfb* formats, and numerous Adobe Font Metric (*.afm*) files, with **lptops**(1). **psposter** looks for this file, and downloadable fonts, in the *AFMPATH* search path.

Typical lines in *afmfonts.map* look like this:

```
% This line is a comment.
% This font name is too long for some
% operating systems, so we store it under a
% shorter name.  The font files will be
% HelvNaBI.afm, containing the character
% metrics (assorted dimensions), and
% HelvNaBI.pfa (ASCII) or HelvNaBI.pfb
% (binary), containing encrypted POSTSCRIPT
% definitions of the character shapes.
Helvetica-Narrow-BoldOblique = HelvNaBI
```

rgb.txt   Color mapping file distributed with the X Window System. **psposter** looks for this file in the *RGBPATH* search path.

Typical lines in *rgb.txt* look like this:

```
000 000 000        black
000 000 255        blue
000 255 000        green
000 255 255        cyan
255 000 000        red
255 000 255        magenta
255 255 000        yellow
255 255 255        white
250 235 215        antique white
250 235 215        AntiqueWhite
```

There is no comment syntax.

The first three numbers define red, green, and blue intensities on a scale of $0 \ldots 255$, and the remaining text on each line is the color name, which, in the case of compound color names, appears in two forms, one in lower case with blank separators, and one without blanks, in mixed case.

SEE ALSO

> **compress**(1), **dxpsview**(1), **ghostscript**(1), **ghostview**(1), **gs**(1), **gzip**(1), **lptops**(1), **pageview**(1), **rgb**(1), **showps**(1), **tex**(1).

AUTHOR

> Nelson H. F. Beebe, Ph.D.
> Center for Scientific Computing
> University of Utah
> Department of Mathematics, 322 INSCC
> 155 S 1400 E RM 233
> Salt Lake City, UT 84112-0090
> Tel: +1 801 581 5254
> FAX: +1 801 585 1640, +1 801 581 4148
> Email: beebe@math.utah.edu, beebe@acm.org, beebe@ieee.org
> URL: http://www.math.utah.edu/~beebe

COPYRIGHT

AVAILABILITY

> Internet source distributions of **psposter** are available at the World-Wide Web Uniform Resource Locator addresses
>
> ```
> ftp://ftp.math.utah.edu/pub/misc/psposter-0.11.jar
> ftp://ftp.math.utah.edu/pub/misc/psposter-0.11.tar.gz
> ```

```
ftp://ftp.math.utah.edu/pub/misc/psposter-0.11.zip
ftp://ftp.math.utah.edu/pub/misc/psposter-0.11.zoo
http://www.math.utah.edu/pub/misc/psposter-0.11.jar
http://www.math.utah.edu/pub/misc/psposter-0.11.tar.gz
http://www.math.utah.edu/pub/misc/psposter-0.11.zip
http://www.math.utah.edu/pub/misc/psposter-0.11.zoo
```