Pbmtext User Manual

pbmtext

Updated: 14 April 2005
Table Of Contents

NAME

pbmtext - render text into a PBM image

SYNOPSIS

pbmtext [**-font** *fontfile*] [**-builtin** *fontname*] [**-space** *pixels*] [**-lspace** *pixels*] [**-nomargins**] [**-width** *pixels*] [*text*]

Minimum unique abbreviation of option is acceptable. You may use double hyphens instead of single hyphen to denote options. You may use white space in place of the equals sign to separate an option name from its value.

DESCRIPTION

This program is part of Netpbm.

pbmtext takes the specified text, either a single line from the command line or multiple lines from standard input, and renders it into a PBM graphical image.

In the image, each line of input is a line of output. Formatting characters such as newline have no effect on the formatting; like any unprintable character, they turn into spaces.

The image is just wide enough for the longest line of text, plus margins, and just high enough to contain the lines of text, plus margins.

The left and right margins are twice the width of the widest character in the font; the top and bottom margins are the height of the tallest character in the font. But if the text is only one line, all the margins are half of this. You can use the **-nomargins** option to eliminate the margins.

pbmtextps does the same thing as **pbmtext**, but uses Ghostscript to generate the characters, which means it's a lot more sophisticated and you can use Postscript fonts. But it also means you have to have Ghostscript installed and it isn't as fast.

OPTIONS

-font

-builtin-builtin selects a font among those built into Netpbm. **-font** selects a font that you supply yourself either as an X Window System BDF (Bitmap Distribution Format) file or as a PBM file in a special form.

The default is the built in font "bdf."

"bdf" is Times-Roman 15 pixels high. (That's about 14 point type printed at 75 dpi).

"fixed" is a built in fixed with font.

To create a font as a PBM file (to use with the **-font** option), do this: In your window system of choice, display the following text in the desired (fixed-width) font:

```
M ",/^_[‘jpqy| M
/  !"#$%&’()*+ /
< ,-./01234567 <
> 89:;<=>?@ABC >
@ DEFGHIJKLMNO @
_ PQRSTUVWXYZ[ _
{ ]^_‘abcdefg {
} hijklmnopqrs }
~ tuvwxyz{|}~  ~
M ",/^_[‘jpqy| M
```

Do a screen grab or window dump of that text, using for instance **xwd**, **xgrabsc**, or **screendump**. Convert the result into a pbm file. If necessary, use **pamcut** to remove everything except the text. Finally, run it through **pnmcrop**. to make sure the edges are right up against the text. **pbmtext** can figure out the sizes and spacings from that.

-space *pixels* Add *pixels* pixels of space between characters. This is in addition to whatever space surrounding characters is built into the font, which is usually enough to produce a reasonable string of text.

pixels may be fractional, in which case the number of pixels added varies so as to achieve the specified average. For example **-space=1.5** causes half the spaces to be 1 pixel and half to be 2 pixels.

pixels may be negative to crowd text together, but the author has not put much thought or testing into how this works in every possible case, so it might cause disastrous results.

-lspace *pixels* Add *pixels* pixels of space between lines. This is in addition to whatever space above and below characters is built into the font, which is usually enough to produce a reasonable line spacing.

pixels must be a whole number.

pixels may be negative to crowd lines together, but the author has not put much thought or testing into how this works in every possible case, so it might cause disastrous results.

-nomargins By default, **pbmtext** adds margins all around the image as described above. This option causes **pbmtext** not to add any margins.

Note that there may still be space beyond the edges of the type because a character itself may include space at its edges. To eliminate all surrounding background, so the type touches all four edges of the image, use **pnmcrop**.

-width *pixels*This specifies how much horizontal space the text is supposed to fit into.

If the input is one line, **pbmtext** breaks it into multiple lines as needed to fit the specified width. It breaks it between characters, but does not pay attention to white space; it may break in the middle of a word and a line may begin or end with white space.

If the input is multiple lines, **pbmtext** assumes you already have line breaks where they make sense, and **pbmtext** simply truncates each line as needed to fit the specified width.

USAGE

Often, you want to place text over another image. One way to do this is with **ppmlabel**. **ppmlabel** does not give you the font options that **pbmtext** does, though.

Another way is to use **pbmtext** to create an image containing the text, then use **pamcomp** to overlay the text image onto your base image. To make only the text (and not the entire rectangle containing it) cover the base image, you will need to give **pamcomp** a mask, via its **-alpha** option. You can just use the text image itself as the mask, as long as you also specify the **-invert** option to **pamcomp**.

If you want to overlay colored text instead of black, just use **ppmchange** to change all black pixels to the color of your choice before overlaying the text image. But still use the original black and white image for the alpha mask.

If you want the text at an angle, use **pnmrotate** on the text image (and alpha mask) before overlaying.

SEE ALSO

pbmtextps, **pamcut**, **pnmcrop**, **pamcomp**, **ppmchange**, **pnmrotate**, **ppmlabel**, **pstopnm**, **pbm**

AUTHOR

Copyright (C) 1993 by Jef Poskanzer and George Phillips

Table Of Contents

NAME
SYNOPSIS
DESCRIPTION
OPTIONS
USAGE
SEE ALSO
AUTHOR