NAME

pkfix–helper – preprocess dvips–produced PostScript documents before passing them to pkfix

SYNOPSIS

pkfix-helper   [−−**help**]   [−−**verbose**]   [−−**force**=*name=fontspec*]   [−−**ps**=*filename.ps*]
[−−**tex**=*filename.tex*]   [−−**cache**=*filename*]   [−−**include**=*fontspec*]   [−−**exclude**=*regexp*]
[−−**keep**=*fontspec*] [−−**quiet**] [−−**no−repeats**] [−−**spp**=*number*] [*input.ps* [*output.ps*]]

DESCRIPTION

Motivation

PostScript documents created with old versions of **dvips** almost invariably utilize bitmapped (PostScript Type 3) fonts. The problem with bitmapped fonts is that they target a specific device resolution; a PostScript file produced using 300 DPI fonts will look grainy on a 600 DPI printer. Even worse, *all* bitmapped fonts look grainy when zoomed in on screen. The solution is to use vector (PostScript Type 1) fonts, which are resolution-independent and appear crisp at any size or scale.

While it is no longer difficult to configure **dvips** to use vector fonts, it is not always possible to rerun **dvips** on an old *.dvi* file. The *.dvi* file and document source may have been lost; or, the source may no longer compile because packages it depends upon may no longer be available.

Heiko Oberdiek's **pkfix** script replaces bitmapped fonts in **dvips**–produced PostScript files with the corresponding vector fonts. It works by parsing the PostScript comments with which **dvips** surrounds bitmapped-font definitions. For example, a font definition beginning with the comment `%DVIPSBitmapFont: Fi cmss10 11 28` and ending with a matching `%EndDVIPSBitmapFont` is known to define font `Fi` as `cmss10` (Computer Modern Sans Serif at a design size of 10 points) scaled to `11` points. Only the `28` characters actually used by the document are defined. **pkfix** then replaces the font definition with one that defines `Fi` using the same set of characters but taken from the *cmss10.pfb* vector font file.

Unfortunately, **pkfix** works only with versions of **dvips** newer than v5.58 (ca. 1996). Naturally, the older a PostScript document, the less likely its sources still exist and can still be recompiled. Older versions of **dvips** lack `%DVIPSBitmapFont` comments and various other PostScript comments on which **pkfix** relies. Without PostScript comments to guide it, **pkfix** is unable to determine which vector fonts correspond with which bitmapped fonts.

Overview

The **pkfix-helper** script is a preprocessor for **pkfix** that attempts to determine the association between each document-font name (e.g., `Fi`) in a PostScript file and the original font (e.g., `cmss10`) and fonts size (e.g., `11` points). It then fabricates the PostScript comments that **pkfix** expects to see so that **pkfix** can do its job.

pkfix-helper works by comparing every document font against every *.tfm* font file it knows about (assuming that each such font has a corresponding *.pfb* vector version) and selecting the best matching *.tfm* file for every document font. **pkfix-helper** has access only to the widths of characters and only to those characters actually used in the document. Also, the program recognizes only a limited set of the most popular *.tfm* files and scaling factors. Consequently, the comparison is imperfect and **pkfix-helper** may attribute an incorrect font to a given name. Fonts comprising only one or two characters actually used in a document are particularly problematic for **pkfix-helper** because many fonts may be near-enough matches to fool the problem.

pkfix-helper is designed so that a user can guide the font-selection process by manually designating matching fonts. With a modicum of diligence and patience a user can correct any

mismatched fonts and help the program provide proper input to **pkfix**.

OPTIONS

pkfix-helper accepts on the command line the filename of a PostScript document to process (with the default being the standard input device) and the filename of a modified PostScript document to create (with the default being the standard output device). The program also accepts the following command-line options:

Frequently Used Options

−h, −−**help**

Display usage information and exit. The −−**verbose** and −−**quiet** options can be used to increase and decrease the amount of information presented.

−v, −−**verbose**

Increase the amount of status information that **pkfix-helper** displays as it runs. Additional instances of −−**verbose** on the command line further increase the program's verbosity. By default, only major operations are displayed. A single −−**verbose** additionally displays information about individual font comparisons. A second −−**verbose** additionally displays details about some of the program's internal operations.

−f *name=fontspec*, −−**force**=*name=fontspec*

Force **pkfix-helper** to associate a specific font with a given font name appearing the document. *name* is a two-character **dvips** font name such as Fa. *fontspec* is a font specification such as cmmi8 or cmsy10 @ 1.1X. An asterisk used in the name of the base font (e.g., cmti*) will automatically try all integral test font sizes from 5 to 17 points (cmti5, cmti6, ..., cmti17). An asterisk used as a scale value (e.g., cmsy10 @ *) will be replaced by the scale value that gives the best match to the original font's metrics. The −−**force** option can be specified repeatedly on the command line.

−p *filename.ps*, −−**ps**=*filename.ps*

Create a PostScript file called *filename.ps* that shows the **dvips** name and a font sample of every font used by the input document.

−t *filename.tex*, −−**tex**=*filename.tex*

Create a Plain TeX file called *filename.tex* that shows the **dvips** name and a font sample of every font that **pkfix-helper** used in the output document.

Infrequently Used Options

−C *filename*, −−**cache**=*filename*

Speed up TFM file processing by caching character metrics into file *filename*. On some systems it takes a long time to read a TFM file, spawn *tftopl* to convert it to PL format, and extract from the PL data the metrics for each character. The first time −−**cache** is specified, **pkfix-helper** proceeds as normal then writes all of the extracted character metrics to *filename*. On subsequent runs in which −−**cache**=*filename* is specified, **pkfix-helper** reads the previously extracted metrics from *filename*, going through the *tftopl*–based process only for TFM files that were not previously encountered.

−q, −−**quiet**

Instruct **pkfix-helper** to produce no output during its run except for fatal error messages.

−1, −−**no−repeats**

Prevent **pkfix-helper** from associating the same *fontspec* with more than one **dvips** font name.

−i *fontspec*, **−−include**=*fontspec*

Add *fontspec* to the list of font specifications against which **pkfix-helper** compares *every* document font. (In contrast, **−−force** designates a font specification to use only for a *specific* document font.) The **−−include** option can be specified repeatedly on the command line.

−x *regexp*, **−−exclude**=*regexp*

Remove all font specifications matching regular expression *regexp* from **pkfix-helper**'s list of known fonts. The **−−exclude** option can be specified repeatedly on the command line.

−k *fontspec*, **−−keep**=*fontspec*

Do not substitute a vector font for bitmapped font *fontspec* (`Fa`, `Fb`, etc.). This is useful when converting documents that use obscure bitmapped fonts for which there is no vector equivalent. For example, it was somewhat common in the past to include graphics such as university or corporate logos into a document by converting the bitmapped image into a single-character font and using that font in LaTeX. **−−keep** prevents such fonts from being replaced. The **−−keep** option can be specified repeatedly on the command line.

−a, **−−any−scale**

Allow any value to be used to scale a font when `"*"` is specified as the scale factor. Normally, **pkfix-helper** considers only integer multiples of 0.1 that are greater than or equal to 1.0 (i.e., *fontname*`@1X`, `@1.1X`, `@1.2X`, `@1.3X`, etc.).

−s, **−−spp**

Specify the number of font samples per page to print to the files indicated using the **−−ps** and **−−tex** options. The default value, 25, should work well in most circumstances.

DIAGNOSTICS

`Best match for` *name* `is rather poor`

The best font **pkfix-helper** found for **dvips** font name *name* has a mismatch value greater than or equal to 1.0. (The mismatch value is the sum of the squares of the difference between the character widths of a document font and a potential replacement font.) Use the **−−force** option to designate an alternative replacement font or scaling amount.

EXAMPLES

For the purpose of the following examples, assume that *oldfile.ps* is the name of a PostScript file produced by an old version of **dvips** and utilizing at least one bitmapped font. It's always worth verifying that **pkfix** can't convert the file on its own:

```
$ pkfix oldfile.ps newfile.ps
PKFIX 1.3, 2005/02/25 - Copyright (c) 2001, 2005 by Heiko Oberdiek.
==> no fonts converted
```

(Alternatively **pkfix** may issue an error message such as `!!!  Error: Parse error (@start parameters)!`.) Only when **pkfix** can't replace bitmapped fonts with vector fonts is **pkfix-helper** needed. In its simplest form, **pkfix-helper** takes the name of an input file (*oldfile.ps* in this example) and the name of an output file (*pkfix−oldfile.ps*), which will have the same contents as the input file but serve as suitable input for **pkfix**:

```
$ pkfix-helper oldfile.ps pkfix-oldfile.ps
Reading oldfile.ps ... done.
Number of Type 3 fonts encountered: 10
Bitmapped fonts are typeset at 600 DPI.
Finding character widths ... done.
Reading TFM files ... done (103 TFMs in 193 scaling variations).
Matching fonts:
    Processing Fi ... done (cmr10 @ 1X, mismatch=0.11683).
    Processing Fa ... done (cmti10 @ 1X, mismatch=0.08892).
    Processing Fb ... done (cmr8 @ 1X, mismatch=0.07133).
    Processing Ff ... done (cmbx12 @ 1.2X, mismatch=0.02948).
    Processing Fh ... done (cmtt10 @ 1X, mismatch=0.06895).
    Processing Fd ... done (cmmi10 @ 1X, mismatch=0.03966).
    Processing Fj ... done (cmbx12 @ 1X, mismatch=0.03972).
    Processing Fe ... done (cmbx10 @ 1X, mismatch=0.00762).
    Processing Fg ... done (cmsy10 @ 1X, mismatch=0.00875).
    Processing Fc ... done (cmr6 @ 1X, mismatch=0.00284).

$ pkfix pkfix-oldfile.ps newfile.ps
PKFIX 1.3, 2005/02/25 - Copyright (c) 2001, 2005 by Heiko Oberdiek.
*** Font conversion: `cmti10' -> `CMTI10'.
*** Font conversion: `cmr8' -> `CMR8'.
*** Font conversion: `cmr6' -> `CMR6'.
*** Font conversion: `cmmi10' -> `CMMI10'.
*** Font conversion: `cmbx10' -> `CMBX10'.
*** Font conversion: `cmbx12' -> `CMBX12'.
*** Font conversion: `cmsy10' -> `CMSY10'.
*** Font conversion: `cmtt10' -> `CMTT10'.
*** Font conversion: `cmr10' -> `CMR10'.
*** Font conversion: `cmbx12' -> `CMBX12'.
*** Merging font `CMBX12' (2).
==> 10 converted fonts.
==> 1 merged font.
```

Although **pkfix-helper** tries to automate as much as possible the font-detection process, some fonts will invariably be incorrectly identified. The program outputs a warning message if it *knows* a match is bad but the lack of a warning message does not necessarily indicate that **pkfix-helper** did a good job. It is therefore strongly recommended that the user produce "before" and "after" font sheets:

```
$ pkfix-helper -q oldfile.ps pkfix-oldfile.ps \
  --ps=oldfonts.ps --tex=newfonts.tex

$ tex newfonts.tex
This is TeX, Version 3.14159 (Web2C 7.4.5)
(./newfonts.tex [1] )
Output written on newfonts.dvi (1 page, 1292 bytes).
Transcript written on newfonts.log.
```

```
$ dvips newfonts.dvi -o newfonts.ps
This is dvips(k) 5.92b Copyright 2002 Radical Eye Software (www.radicale
' TeX output 2006.06.11:1636' -> newfonts.ps
<texc.pro><8r.enc><texps.pro>. <cmr6.pfb><cmsy10.pfb><cmbx10.pfb><cmbx12
<cmmi10.pfb><cmtt10.pfb><cmr8.pfb><cmti10.pfb><cmr10.pfb>[1]
```

After running the preceding commands, *oldfonts.ps* shows samples of the fonts in *oldfile.ps* and *newfonts.ps* shows samples of the replacement fonts that **pkfix-helper** used to produce *pkfix−oldfile.ps*. Print *oldfonts.ps* and *newfonts.ps* and compare them carefully for incorrect fonts and sizes.

Suppose that the choice of `cmbx12 @ 1.2X` for font `Ff` looks wrong; say the characters look taller in *oldfonts.ps* than in *newfonts.ps*. This is where the trial-and-error stage begins. Let's hypothesize that `cmb12` is a better match than `cmbx12` but we don't know how much to scale the font. Fortunately, **pkfix-helper** allows `*` to be used as a scaling factor to tell the program to automatically detect an optimal scaling factor, even if doing so means choosing a nonstandard font size:

```
$ pkfix-helper oldfile.ps pkfix-oldfile.ps --force="Ff=cmb12 @ *"
Reading oldfile.ps ... done.
Number of Type 3 fonts encountered: 10
Bitmapped fonts are typeset at 600 DPI.
Finding character widths ... done.
Reading TFM files ... failed.
pkfix-helper: Unable to process user-specified TFM file "cmb12"
```

Oops, it looks like we don't have a *cmb12.tfm* file on our system. Let's try scaling up *cmb10.tfm* instead:

```
$ pkfix-helper oldfile.ps pkfix-oldfile.ps --force="Ff=cmb10 @ *"
Reading oldfile.ps ... done.
Number of Type 3 fonts encountered: 10
Bitmapped fonts are typeset at 600 DPI.
Finding character widths ... done.
Reading TFM files ... done (103 TFMs in 193 scaling variations).
Matching fonts:
    Processing Fi ... done (cmr10 @ 1X, mismatch=0.11683).
    Processing Fa ... done (cmti10 @ 1X, mismatch=0.08892).
    Processing Fb ... done (cmr8 @ 1X, mismatch=0.07133).
    Processing Ff ... done (cmb10 @ 1.5X, mismatch=0.00035).
    Processing Fh ... done (cmtt10 @ 1X, mismatch=0.06895).
    Processing Fd ... done (cmmi10 @ 1X, mismatch=0.03966).
    Processing Fj ... done (cmbx12 @ 1X, mismatch=0.03972).
    Processing Fe ... done (cmbx10 @ 1X, mismatch=0.00762).
    Processing Fg ... done (cmsy10 @ 1X, mismatch=0.00875).
    Processing Fc ... done (cmr6 @ 1X, mismatch=0.00284).
```

The match has definitely improved, although 15 pt. is certainly an odd size for a font. Then again, many documents *do* use nonstandard sizes so this may in fact be correct. The best way to verify is once again to produce, print, and compare a pair of font samples and iterate until all of the fonts look correct. Use one instance of −−**force** for each font you want to alter.

ENVIRONMENT

pkfix-helper honors the following environment variables:

GS   The name of the Ghostscript interpreter (default: *gs*)

TFTOPL   The name of a utility for converting *.tfm* files to *.pl* files (default: *tftopl*)

BUGS

Even when **pkfix-helper** finds a perfect match (i.e., the correct font in the correct size) the mismatch value is still typically nonzero. The same error is probably what causes **pkfix-helper** sometimes to consider the wrong font as being a better match than the correct font. Suggestions for fixing these bugs are welcome.

RESTRICTIONS

pkfix-helper works only with PostScript files produced by **dvips**, not with arbitrary PostScript files. The program has not been tested with output from versions of **dvips** older than v5.490 (ca. 1992); output from older versions may or may not work. Only bitmapped fonts loaded by **dvips** can be analyzed, not bitmapped fonts loaded by embedded graphics.

pkfix-helper works by comparing character widths, not the actual glyphs. Consequently, it is misled by sets of fonts with similar character widths (at least for those characters used by a given document). As an extreme example, all Computer Modern Teletype fonts of a given design size (e.g., `cmtt10`, `cmsltt10`, and `cmitt10`) use exactly the same widths for all characters. Human assistance is generally needed to guide **pkfix-helper**'s font-matching procedures.

NOTES

Files produced using the −−**tex** option are Plain TeX files and therefore must be compiled with *tex* (or a variation such as *pdftex*, *luatex*, etc.), *not* with *latex*.

SEE ALSO

pkfix (1), *dvips* (1), *tex* (1), *gs* (1)

PostScript Language Reference, Third Edition. Published by Addison-Wesley, ISBN 0−201−37922−8, <http://www.adobe.com/products/postscript/pdfs/PLRM.pdf>.

AUTHOR

Scott Pakin, *scott+pkfh@pakin.org*

COPYRIGHT AND LICENSE

Copyright (C) 2011, Scott Pakin

This file may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3c of this license or (at your option) any later version. The latest version of this license is in <http://www.latex−project.org/lppl.txt> and version 1.3c or later is part of all distributions of LaTeX version 2006/05/20 or later.