

**NAME**

autoinst – wrapper script around *otftotfm*, for installing OpenType fonts.

**SYNOPSIS**

autoinst [options] *fontfile* [*fontfile* ...]

**DESCRIPTION**

Eddie Kohler’s *otftotfm* is a great tool for installing OpenType fonts for use with LaTeX, but its use (even in automatic mode) is quite complicated because it needs lots of long command lines and doesn’t generate the *fd* and *sty* files LaTeX needs. **autoinst** simplifies the font installation process by generating and executing all commands for *otftotfm* and by creating all necessary *fd* and *sty* files. All that’s then left to do is move these files to a suitable location ( $\$LOCALTEXMF/tex/latex/<Supplier>/<FontFamily>/$  is the canonical choice) and update TeX’s filename database.

Given a number of fonts (from the same family!), **autoinst** will create several LaTeX font families:

- Four text families (with lining and oldstyle figures, in tabular and proportional variants), each with the following shapes:
  - n* Roman text
  - sc* Small caps
  - nw* ‘Upright swash’; usually normal text with some extra ‘oldstyle’ ligatures, such as ct, sp and st.
  - tl* Titling shape. Meant for all-caps text only (even though it sometimes contains lowercase glyphs as well), where letterspacing and the positioning of interpunction characters have been adjusted to suit all-caps text. This shape is generated only for the families with lining figures.
  - it* Italic or oblique text
  - scit* Italic small caps
  - sw* Swash
  - tlit* Italic titling
- For each text family: a family of TS1–encoded symbol fonts, in roman and italic shapes.
- Four families with superiors, inferiors, numerators and denominators, in roman and italic shapes.
- An ornament family, in roman and italic shapes.

Of course, if the font doesn’t contain oldstyle figures, small caps etc., the corresponding shapes or families are not created; and in addition, command line options allow fine-grained control over the creation of many families and shapes (see below).

The generated font families are named  $<FontFamily>-<Suffix>$ , where  $<Suffix>$  is one of

<i>LF</i>	proportional (i.e., figures have varying widths) lining figures
<i>TLF</i>	tabular (i.e., all figures have the same width) lining figures
<i>OsF</i>	proportional oldstyle figures
<i>TOsF</i>	tabular oldstyle figures
<i>Sup</i>	superior characters (note: most fonts have an incomplete set of superiors (usually figures, punctuation and the letters <i>abdeilmnorst</i> ); normal forms will be used for the other characters)
<i>Inf</i>	inferior figures and punctuation (normal letters)
<i>Orn</i>	ornaments
<i>Numr</i>	numerators
<i>Dnom</i>	denominators

The generated fonts are named  $<FontFile>-<suffix>-<shape>-<enc>$ , where  $<FontFile>$  is the name of the OpenType file,  $<suffix>$  is the same as above (but in lowercase),  $<shape>$  is either empty, ‘sc’, ‘swash’ or ‘titling’, and  $<enc>$  is the encoding. A typical name in this scheme is *MinionPro-Regular-osf-sc-ly1*.

### On the choice of text encoding

By default, all text families use the LY1 encoding. This has been chosen over T1 (Cork) because many OpenType fonts (especially the ‘Pro’ ones) contain additional ligatures such as fj and Th, and LY1 has a number of empty slots to accommodate these.

A different encoding can be selected using the **—encoding** command line option (see below).

### Using the fonts with LaTeX

**autoinst** generates a style file for using the font in LaTeX documents, named ‘<FontFamily>.sty’. This style file also takes care of loading the *fontenc* and *textcomp* packages, if necessary. To use the font, simply put `\usepackage{MinionPro}` (or whatever the font is called) in the preamble of the document.

This style file defines a number of options:

*lining, oldstyle, tabular, proportional*

Choose which figures will be used for the text fonts. The defaults are ‘oldstyle’ and ‘proportional’ (if available).

*light, regular*

Choose the weight that LaTeX will use for the ‘normal’ weight (i.e., the value of `\mddefault`).

*medium, demibold, semibold, bold, black*

Choose the weight that LaTeX will use for the ‘bold’ weight (i.e., the value of `\bfdefault`).

The style file will also try to load the *fontaxes* package (part of the MinionPro for LaTeX project), which gives easy access to various font shapes and styles. This package can be downloaded from the project’s homepage (<http://developer.berlios.de/projects/minionpro>) or directly through the CVS web interface (<http://cvs.berlios.de/cgi-bin/viewcvs.cgi/minionpro/MinionPro/tex/>), and is also available from CTAN as part of the archive *base-v2.zip* (<http://www.ctan.org/tex-archive/fonts/minionpro/base-v2.zip>).

Using the machinery set up by *fontaxes*, the generated style file also defines a few commands (which take one argument) and declarations (which don’t take arguments, but affect all text up to the end of the current group) of its own:

DECLARATION	COMMAND
<code>\tlshape</code>	<code>\textttl, \texttitling</code>
<code>\sufigures</code>	<code>\textsu, \textsuperior</code>
<code>\infigures</code>	<code>\textin, \textinferior</code>

In addition, the `\swshape` and `\textsw` commands are redefined to place swash on the secondary shape axis (*fontaxes* places it on the primary shape axis); this allows the use of ‘upright swash’. Just saying `\swshape` will still give normal (italic) swash, but `\swshape\upshape` results in upright swash.

Note that there is no separate command for accessing the italic titling shape; but these commands behave properly when nested, so `\tlshape\itshape` gives italic titling. There are also no commands for accessing the numerator and denominator fonts; these can be selected using *fontaxes*’ low-level commands, e.g., `\fontfigurestyle{numerator}\selectfont`.

The style file also provides a command `\ornament{<number>}`, where *<number>* is a number from 0 to the total number of ornaments minus one. Ornaments are always typeset using the current family, series and shape. A list of all ornaments in a font can be created by running LaTeX on the file *nfssfont.tex* (which comes with LaTeX) and specifying the ornament font (e.g., *MinionPro-Regular-orn-u*).

This whole machinery builds on *fontaxes*; if that package cannot be found, the style file doesn’t provide access to the various font shapes and styles.

## Using multiple font families in one document

When using several font families in one document, keep the following points in mind:

- All text fonts should preferably use the same encoding.
- All fonts should be installed using the same version of **autoinst**; style files that were generated by different versions of **autoinst** are unlikely to be able to coexist peacefully.
- Sanserif and typewriter fonts are best installed using the **—sanserif** and **—typewriter** command line options (see below).
- The style file for the main text font should be loaded *last*.

## A note for non-**te**TeX users

Calling *otftotfm* with the **—automatic** option (as **autoinst** does by default) requires a TeX-installation that uses the *kpathsea* library; with TeX-installations that implement their own directory searching (such as MiKTeX) *otftotfm* might complain that it cannot find a writable *texmf* directory and leave all generated *tfm*, *vf*, *enc* and *map* files in the current working directory. In that case, you need to move these to their correct destinations. You also need to tell the dvi-driver (*dvips*, *dvipdfm*, *pdfTeX* etc.) about the new font map files; this usually means editing some configuration file.

Furthermore, some OpenType fonts lead to *pl* and *vpl* files that are too big for MiKTeX's implementation of *pltotf* and *vptovf*; the versions that come with TeXLive (<http://tug.org/ftp/texlive/Contents/live/bin/win32/>) don't have this problem.

## COMMAND LINE OPTIONS

### **—encoding=encoding**

Use the encoding *encoding* for the text fonts. The default is 'ly1'. A file named '*<encoding>.enc*' should be somewhere where *otftotfm* can find it. Suitable encoding files (named in all *lowercase*) for LY1, T1 and TS1 come with the *fontools* package.

### **—sanserif**

Install the font as a sanserif font, accessed via `\sffamily` and `\textsf`. The generated style file redefines `\familydefault`, so including it will still make this font the default text font.

### **—typewriter**

Install the font as a typewriter font, accessed via `\ttfamily` and `\texttt`. The generated style file redefines `\familydefault`, so including it will still make this font the default text font.

### **—ts1**

### **—nots1**

Turn the creation of TS1-encoded fonts on or off. The default is **—ts1** if the text encoding is T1, **—nots1** otherwise.

### **—smallcaps**

### **—nosmallcaps**

Turn the creation of small caps fonts on or off. The default is **—smallcaps**.

### **—swash**

### **—noswash**

Turn the creation of swash fonts on or off. The default is **—swash**.

### **—titling**

### **—notitling**

Turn the creation of titling fonts on or off. The default is **—notitling**.

### **—superiors**

### **—nosuperiors**

Turn the creation of fonts with superior characters on or off. The default is **—superiors**.

**--inferiors****--noinferiors**

Turn the creation of fonts with inferior figures on or off. The default is **--noinferiors**.

**--fractions****--nofractions**

Turn the creation of fonts with numerators and denominators on or off. The default is **--nofractions**.

**--ornaments****--noornaments**

Turn the creation of ornament fonts on or off. The default is **--ornaments**.

**--manual**

Manual mode. By default, **autoinst** immediately executes all *otftotfm* command lines it generates; with the **--manual** option, these commands are instead written to a batch command file (named '*<font>.bat*', to make things easier for our friends on Windows). Also, the generated *otftotfm* command lines specify the **--pl** option and leave out the **--automatic** option; this causes human readable (and editable) *pl* and *vpl* files to be created instead of the default *tfm* and *vf* files.

**--verbose**

Verbose mode; print detailed info about what **autoinst** thinks it's doing.

**--extra=***text*

Pass *text* as options to *otftotfm*. To prevent *text* from accidentally being interpreted as options to **autoinst**, it's best to quote it.

**CAVEATS**

- **autoinst** needs Perl (at least version 5.6) and the *LCDF TypeTools*. The *fontaxes* package is highly recommended, but not required.
- The *LCDF TypeTools* recently acquired the ability to handle TrueType-flavored OpenType fonts as well. **autoinst**, however, has only been tested with PostScript-flavored OpenType fonts.
- Each font's weight, shape and width are determined by parsing the filename. This supposes the naming scheme used by Adobe; it may not work with fonts from other vendors. Even Adobe's own fonts don't always follow that scheme to the letter. **autoinst** tries hard to make sense of the filenames and most font families will install without problems, but some families (especially those with unusual weights or widths) will break it. If that happens to you, send me a bug report and I'll try to fix it.
- When choosing which shapes and families to build, **autoinst** relies on information in the font; when that information isn't accurate (e.g., CourierStd claims a 'supers' feature but only contains superior variants of 'one', 'two' and 'three', not of the other figures or letters), **autoinst** won't notice anything wrong (but the user will!).
- **autoinst** does a pretty good job of handling many standard font families; however, its one-size-fits-all approach is less well suited to 'exotic' families such as Poetica, Silentium and Zapfino that contain many alternate glyphs and shapes. For fonts like these, it's usually better to write the commands for *otftotfm* by hand, or even to convert the font to Type 1 format and use *fontinst*; or one might try the XeTeX system, which natively supports OpenType fonts.
- Don't install fonts from more than one family at the same time.

**SEE ALSO**

Eddie Kohler's *LCDF TypeTools* (<http://www.lcdf.org/type>).

John Owens' *otftex\_install* ([http://www.ece.ucdavis.edu/~jowens/code/otftex\\_install/](http://www.ece.ucdavis.edu/~jowens/code/otftex_install/)) is another wrapper around *otftotfm*, and may work in situations where **autoinst** fails.

Ready-made support files for MinionPro, providing more options and flexibility than **autoinst** ever will (including math), are available from <http://developer.berlios.de/projects/minionpro/>.

XeTeX (<http://scripts.sil.org/xetex>) is a TeX extension that can use any font supported by the operating system (including OpenType fonts); it doesn't need *tfm*, *vf*, *fd*, *enc* or *map* files. It also isn't hindered by

standard TeX's limitation to 8-bit fonts, so it is well suited to fonts with many ligatures and alternate glyphs.

The other programs in the *fontools* bundle: *afm2afm*, *cmap2enc*, *font2afm*, *ot2kpx*, *pfn2kpx*, *showglyphs*.

A good (and free-as-in-beer) Perl-implementation for Windows (and Linux) is ActivePerl, available from <http://www.activestate.com>.

## AUTHOR

Marc Penninga <marc@penninga.info>

When sending a bug report, please give as much relevant information as possible; this usually includes (but may not be limited to) the output from running **autoinst** with the **--verbose** option. Also mention the name *fontools* somewhere in the subject line, or risk being caught by my spam filter.

## COPYRIGHT

Copyright (c) 2005–2006 Marc Penninga.

## LICENSE

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

A copy of the GNU General Public License is included with the *fontools* collection; see the file *GPLv2.txt*.

## DISCLAIMER

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

## HISTORY

- 2006–08–31 Made the generated style files try to include 'fontaxes.sty'; changed the names of the generated fonts and families (to make the previous change possible); added command line options for most font styles and shapes; tweaked the filename parsing code for Cronos Pro and Gill Sans Pro; added runtime generation of encoding vectors for ornament fonts (because GaramondPremier's ornament names differ from other fonts); changed the NFSS-code for italic small caps and titling to 'scit' and 'tlit' (to work with *fontaxes*); and edited (and hopefully improved) the documentation.
- 2005–10–03 When creating LY1, T1, OT1 or TS1 encoded fonts, the **--coding-scheme** option is added to the commands for *otftotfm*; this should make the generated *pl* and *vpl* files acceptable to *fontinst*. Also elaborated the documentation somewhat and fixed a small bug.
- 2005–09–22 Added check to see if filename parsing succeeded; updated the filename parsing code to cater for GaramondPremier, Silentium and some non-Adobe fonts; added the **--sanserif** and **--typewriter** options and hacked the style files to support using several different font families in one document.
- 2005–09–12 Cleaned up the code (it now runs under the *strict* and *warnings* pragmas); fixed a (rather obscure) bug that occurred when creating TS1-encoded fonts for families with multiple optical masters and oldstyle figures; added the *medium*, *semibold* etc. options to the style file; and improved the layout of the generated files.
- 2005–08–11 The generated commands weren't actually executed, only printed. Also added a small hack to cater for fonts (such as some recent versions of MinionPro) that contain swash characters but don't provide a 'swsh' feature.
- 2005–08–10 Dropped the 'fontname' scheme in favor of a more verbose naming scheme, since many file-names were still more than eight characters long. Added *nfssect.sty*-like commands to the generated style file. Changed the default encoding to LY1 and added the 'inferior' shape.

- 2005-08-01 Rewrote (and hopefully improved) the user interface; changed the program to by default execute the generated *otftotfm* command lines rather than writing them to a file; added automatic determination of the ‘fontname’ code for the font family; changed the NFSS code for italic small caps to ‘si’; added titling shapes; changed the generated style file to include an interface for the ornaments and to load Lehman’s NFSS extensions *nfssext.sty* if this is installed; corrected the ‘fontname’ codes for OT1, T1, LY1 and user-specific encodings; extended the output generated by the **--verbose** option; and rewrote and extended the documentation.
- 2005-06-16 Did some more finetuning to the filename-parsing code.
- 2005-05-31 Generate correct fontname for OT1-encoded fonts.
- 2005-05-18 Tried to make the filename-parsing code a bit more robust by adding several weights and widths; changed the error that’s displayed when filename parsing fails; commented the code.
- 2005-04-29 Rewrote large parts of the code (yes it *was* even worse).
- 2005-04-18 Changed default text-encoding to T1, added TS1.
- 2005-03-29 Added support for font families with multiple widths.
- 2005-03-15 First version.