

**NAME**

autoinst – wrapper script around *otftotfm*, for installing (PostScript-flavored) OpenType fonts.

**SYNOPSIS**

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

**DESCRIPTION**

Eddie Kohler’s *otftotfm* is a great tool for preparing OpenType fonts for use with LaTeX, but its use (even in automatic mode) is quite difficult because it needs lots of long command lines and still requires you to write the *fd* and *sty* files by hand. **autoinst** simplifies using *otftotfm* by generating and executing all commands for *otftotfm* and by generating all necessary *fd* and *sty* files. All you need to do is move these *fd* and *sty* files to a suitable location (`$LOCALTEXMF/tex/latex/<supplier>/<FontFamily>/` is usually a good choice) and update TeX’s filename database.

Given one or more OpenType fonts, **autoinst** will create several LaTeX font families:

- A text family with lining figures, containing (for each weight and width) the following shapes:
  - n* Roman (upright) text
  - sc* Small caps
  - nw* ‘Upright swash’; usually roman text with extra 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), with different letterspacing and positioning of hyphens, parentheses etc. This doesn’t use the separate ‘titling’ glyphs found in some fonts; these are best installed manually.
  - it* Italic text
  - si* Italic small caps
  - sw* Normal (italic) swash
  - ti* Italic titling
- A text family with oldstyle figures; this contains the same shapes as the text family with lining figures, except for the titling shape (there’s no point in using oldstyle figures with all-caps text).
- For each text family: a family of TS1–encoded symbol fonts, in roman and italic shapes.
- Two families with superior and inferior letters and figures, in roman and italic shapes.
- Two families with numerators and denominators (for creating fractions), in roman and italic shapes.
- An ornament family, in roman and italic shapes.

Of course, if your font doesn’t contain oldstyle figures, small caps etc., the corresponding shapes or families are not created. Creation of the non-text families is controlled using command line options (see below).

The generated font families are named `<FontFamily><suffix>`, where `<suffix>` is one of

- X lining figures
- J oldstyle figures
- 1 superior letters and figures
- 0 inferior letters and figures
- 11 numerators
- 00 denominators

The generated fonts don’t follow the ‘fontname’ (a.k.a. ‘Berry’) scheme, but are named more verbosely: `<ENC>--<FontFile>--<features>`, where `<ENC>` is the encoding (e.g., ‘LY1’), `<FontFile>` is the name of the OpenType file (minus the extension ‘.otf’) and `<features>` is a list of some of the OpenType features (just enough to make the filename unique) that were used to create this font. A typical name in this scheme is `LY1 --MinionPro-Regular --onum --smcp`.

**autoinst** fully supports font families with multiple optical sizes.

### 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 so-called ‘Pro’ ones) contain alternate characters or additional ligatures such as fj and Th, and LY1 has some empty slots to accommodate these.

A different encoding can be selected using the `--encoding` option (see below).

### Using the fonts with LaTeX

**autoinst** generates a LaTeX style file for using the font in your documents, named ‘*FontFamily.sty*’. Using the font is as simple as putting the command `\usepackage{MinionPro}` (or whatever your font is called) in the preamble of your document.

The generated style file defines a few options:

*lining*

Use lining figures for text.

*oldstyle*

(Only if the font contains oldstyle figures.) Use oldstyle figures for text.

*medium*

*demibold*

*semibold*

*bold*

*black*

(Only if the font family contains the corresponding weight.) Choose the default weight that LaTeX will use when you ask for ‘bold’.

*normalfootnotes*

Use the standard footnotes (don’t redefine `\@makefnmark` to use superior figures).

If the font contains oldstyle figures, these are used by default; you can use the lining figures by explicitly specifying the *lining* option. The style file also calls the *fontenc* and *textcomp* packages if necessary, and defines a number of declarations (which don’t take arguments, but affect all text until the end of the current group) and commands (which only affect their — one — argument) for easy access to the various font styles and shapes:

DECLARATION	COMMAND	EFFECT
<code>\lnstyle</code>	<code>\textln</code>	Use lining figures
<code>\osstyle</code>	<code>\textos</code>	Use oldstyle figures
<code>\sishape</code>	<code>\textsi</code>	Use italic small caps
<code>\swshape</code>	<code>\textsw</code>	Use italic swash
<code>\tlshape</code>	<code>\texttl</code>	Use the (roman) titling font
<code>\tishape</code>	<code>\textti</code>	Use the italic titling font
<code>\sustyle</code>	<code>\textsu</code>	Use superior letters and figures
<code>\instyle</code>	<code>\textin</code>	Use inferior letters and figures

Most of these definitions were taken verbatim from Philipp Lehman’s *nfssect.sty* (but we can’t include that package directly, because it assumes that fonts are named using the ‘fontname’ scheme).

There are no commands for accessing numerators and denominators; these are only useful for creating fractions, so the style file provides a command `\fraction{<numerator>}{<denominator>}` instead.

Ornaments are accessed using the `\ornament{<number>}` command, where *<number>* is a number between 1 and the total number of ornaments. Ornaments are typeset in the current fontseries and *–shape*.

If the font contains superior glyphs, the generated style file redefines `\@makefnmark` so that the superior figures are used for footnote marks. These usually look better than scaled-down normal figures. The style file option *normalfootnotes* can be used to switch back to normal footnote marks.

### Caveat: using multiple font families in one document

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

- All fonts should use the same encoding;
- Sanserif and typewriter fonts should be installed using the `--sanserif` and `--typewriter` options, respectively; otherwise the various style files will override each other's changes to `\rmdefault`;
- The style file for the main text font should be loaded *last*;
- If some of the fonts don't contain superior figures, *all* style files should be loaded using the *normal-footnotes* option.

The generated style files try pretty hard to make sure that the above-mentioned commands and declarations (`\lnstyle et al`) use fonts from the correct family; but let me know if they fail.

There is no easy way to use several serif fonts (or several sanserif fonts, or several typewriter fonts) in the same document. As any good book on typography will tell you, this is something you shouldn't do anyway, so this limitation will probably not be a big problem.

### A note for MiKTeX users

Calling *otftotfm* with the `--automatic` option (as **autoinst** does by default) requires a TeX-installation that uses the *kpathsea* library; with MiKTeX (and probably other TeX-installations that implement their own directory searching as well) *otftotfm* complains that it cannot find a writable *texmf* directory and leaves all generated *tfm*, *vf* and *map* files in your current working directory. You should move these manually to their destinations. You'll also need to manually tell *dvips* and *pdfTeX* about your new font map files.

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.

## OPTIONS

You need only type as many characters as needed to make the option name unique.

### `--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 family as a sanserif font; the font is accessed through `\sffamily` and `\textsf` rather than `\rmfamily` and `\textrm`. (The generated style file redefines `\familydefault`, so including the style file will still make this font the default text font.) This option is mutually exclusive with the `--typewriter` option.

### `--typewriter`

Install the font family as a typewriter font; the font is accessed through `\ttfamily` and `\texttt` rather than `\rmfamily` and `\textrm`. (The generated style file redefines `\familydefault`, so including the style file will still make this font the default text font.) This option is mutually exclusive with the `--sanserif` option.

### `--ts1`

### `--nots1`

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

### `--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 characters on or off. The default is **--noinferiors**.

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

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

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

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

**--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 it executable 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.

**RESTRICTIONS**

- **autoinst** needs Perl (at least version 5.6) and the *LCDF TypeTools*.
- Each font's weight, shape and width are determined by parsing the filename. This supposes Adobe's naming scheme; it probably won't work with fonts from other vendors. Also, Adobe's naming scheme seems to vary slightly from font to font. **autoinst** tries hard to make sense of the filenames and most font families will install without problems, but some families (mostly those with highly 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'), you may need to edit the generated files by hand to fix the problem.
- **autoinst** does a pretty good job of handling many standard font families; however, its one-size-fits-all approach is less well suited to more 'exotic' families such as Poetica, Silentium and Zapfino. 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*.
- You can't install fonts from more than one family at the same time.

**SEE ALSO**

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

The other programs in the *fontools* bundle: *afm2afm*, *cmap2enc*, *font2afm*, *ot2kpx*, *pfm2kpx*, *showglyphs*.

If **autoinst** doesn't work for you, I recommend you take a look at John Owens' *oftex\_install.py* (to be found at [http://www.ece.ucdavis.edu/~jowens/code/oftex\\_install/](http://www.ece.ucdavis.edu/~jowens/code/oftex_install/)). This represents a very different approach to wrapping *otftotfm*, and may work in situations where **autoinst** fails.

**AUTHOR**

Marc Penninga <[marc@penninga.info](mailto:marc@penninga.info)>

If you're sending a bug report, please give as much information as possible, including the output from running **autoinst** with the **--verbose** option. Also be sure to mention the name *fontools* somewhere in the subject line; otherwise you might get caught by my spam filter.

## COPYRIGHT

Copyright (c) 2005 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

- 2005-10-03 When creating LY1, T1, OT1 or TS1 encoded fonts, the `--coding-scheme` option is added to the commands for *otftotfm*; this makes 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; and added the *normalfootnotes* option to the style file.
- 2005-09-12 Cleaned up the code (it now runs under the *strict* and *warnings* pragmas); fixed a (rather obscure) obscure bug that occurred when creating TS1-encoded fonts for families with multiple optical masters and oldstyle figures; added the numerator and denominator families and the `\fraction` command; 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 filenames 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 *nfssect.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.