

**NAME**

**autoinst** – wrapper around the LCDF TypeTools, for installing OpenType fonts in LaTeX.

**SYNOPSIS**

**autoinst** [*options*] **fontfile(s)**

**DESCRIPTION**

Eddie Kohler’s *LCDF TypeTools* are superb tools for installing OpenType fonts in LaTeX, but they can be hard to use: they need many, often long, command lines and don’t generate the *fd* and *sty* files LaTeX needs. **autoinst** simplifies the use of the *TypeTools* for font installation by generating and executing all commands for *otftotfm* and by creating and installing all necessary *fd* and *sty* files.

Given a family of font files (in *otf* or *ttf* format), **autoinst** will create several LaTeX font families:

- Four text families (with lining and oldstyle digits, in both tabular and proportional variants), each with the following shapes:

<i>n</i>	Roman text
<i>it, sl</i>	Italic and slanted (sometimes called oblique) text
<i>sc</i>	Small caps
<i>sw</i>	Swash
<i>tl</i>	Titling shape. Meant for all-caps text only (even though it sometimes contains lowercase glyphs as well), where letterspacing and the positioning of punctuation characters have been adjusted to suit all-caps text. (This shape is only generated for the families with lining digits, since old-style digits make no sense with all-caps text.)
<i>scit, scsl</i>	Italic and slanted small caps
<i>nw</i>	“Upright swash”; usually normal text with “oldstyle” ligatures such as ct, sp and st.
<i>tlit, tsl</i>	Italic and slanted titling text

- For each text family: a family of TS1–encoded symbol fonts, in roman, italic and slanted shapes.
- Four families with superiors, inferiors, numerators and denominators, in roman, italic and slanted shapes.
- An ornament family, in roman, italic and slanted shapes.

Of course, if the fonts don’t contain italics, oldstyle digits, small caps etc., the corresponding shapes and families are not created. Furthermore, the creation of most families and shapes can be controlled by command-line options (see “COMMAND-LINE OPTIONS” below).

These families use the *FontPro* project’s naming scheme: *<FontFamily>–<Suffix>*, where *<Suffix>* is:

<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 (many fonts have only an incomplete set of superior characters: digits, some punctuation and the letters <i>abdeilmnorst</i> ; normal forms are used for other characters)
<i>Inf</i>	inferior characters; usually only digits and some punctuation, normal forms for other characters
<i>Orn</i>	ornaments
<i>Numr</i>	numerators
<i>Dnom</i>	denominators

The generated fonts are named `<FontName>-<suffix>-<shape>-<enc>`, where `<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 “LinLibertineO-osf-sc-ly1”.

### On the choice of text encoding

By default, **autoinst** generates text fonts with OT1, T1 and LY1 encodings, and the generated style files use LY1 as the default text encoding. LY1 has been chosen over T1 because it has some empty slots to accomodate the additional ligatures provided by many OpenType fonts. Different encodings can be chosen using the `-encoding` command-line option (see “COMMAND-LINE OPTIONS” below).

### Using the fonts in your LaTeX documents

**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. To use the font, simply put `\usepackage{<FontFamily>}` in the preamble of your document.

This style file defines a number of options:

`lining, oldstyle, tabular, proportional`

Choose which figure style to use. The defaults are “oldstyle” and “proportional” (if available).

`scale=<number>`

Scale the font by a factor of `<number>`. For example: to increase the size of the font by 5%, use `\usepackage[scale=1.05]{<FontFamily>}`. May also be spelled `scaled`.

This option is only available when you have the *xkeyval* package installed.

`ultrablack, ultrabold, heavy, extrablack, black, extrabold, demibold, semibold, bold`

Choose the weight that LaTeX will use for the “bold” weight.

`light, medium, regular`

Choose the weight that LaTeX will use for the “regular” weight.

These last two groups of options will only work if you have the *mweights* package installed.

The style file will also try to load the *fontaxes* package (available on CTAN), which gives easy access to various font shapes and styles. Using the machinery set up by *fontaxes*, the generated style file defines a number of commands (which take the text to be typeset as 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	SHORT FORM OF COMMAND
<code>\tlshape</code>	<code>\texttitling</code>	<code>\textttl</code>
<code>\sufigures</code>	<code>\textsuperior</code>	<code>\textsu</code>
<code>\infigures</code>	<code>\textinferior</code>	<code>\textin</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 makes these commands behave properly when nested, so that `\swshape\upshape` will give upright swash.

There are no commands for accessing the numerator and denominator fonts; these can be selected using `fontaxes'` standard 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* (part of a standard LaTeX installation) and supplying the name of the ornament font.

To access the ornaments, **autoinst** creates a font-specific encoding file `<FontFamily>_orn.enc`, but only if that file doesn't yet exist in the current directory. This is a deliberate feature that allows you to provide your own encoding vector, e.g. if your fonts use non-standard glyph names for ornaments.

These commands are only generated for existing shapes and number styles; no commands are generated for shapes and styles that don't exist, or whose generation has been turned off using command-line options. Also: these commands are built on top of *fontaxes*; if that package cannot be found, you're limited to using the lower-level commands from standard NFSS (`\fontfamily`, `\fontseries`, `\fontshape` etc.).

### Using multiple font families in one document

Style files generated by versions of **autoinst** older than 2013-07-25 redefined `\mddefault` and `\bfdefault`, whereas newer style files use the *mweights* package instead. If you use multiple **autoinst**-generated font families in the same document, it is best if all style files are generated by the same version of **autoinst**; re-generate the older families if necessary.

### NFSS codes

NFSS identifies fonts by a combination of family, series (weight plus width), shape and size. **autoinst** parses the output of `otfinfo --info` to determine these parameters. When this fails (e.g., because the font family contains uncommon widths or weights), **autoinst** ends up with different fonts having the *same* values for these font parameters, which means that these fonts cannot be used in NFSS. In that case, **autoinst** will split the font family into multiple subfamilies (based on each font file's "Subfamily" value) and try again. (Since many font vendors misunderstand the "Subfamily" concept and make each font file its own separate subfamily, this strategy is only used as a last resort.)

If such a proliferation of font families is unwanted, either run **autoinst** on a smaller set of fonts or add the missing widths, weights and shapes to the tables `%FD_WIDTH`, `%FD_WEIGHT` and `%FD_SHAPE`, at the top of the source code. Please also send a bug report (see AUTHOR below).

**autoinst** maps widths, weights and shapes to NFSS codes using the following tables. These are based as much as possible on the standard *Fontname* scheme and Philipp Lehman's *Font Installation Guide*, but some changes were made to avoid name clashes in font families with

many widths and weights.

WEIGHT		WIDTH	
Thin	t	Ultra Compressed	up
Ultra Light	ul	Extra Compressed	ep
Extra Light	el	Compressed	p
Light	l	Compact	p
Book	[1]	Ultra Condensed	uc
Regular	[1]	Extra Condensed	ec
Medium	mb	Condensed	c
Demibold	db	Narrow	n
Semibold	sb	Semicondensed	sc
Bold	b	Regular	[1]
Extra Bold	eb	Semiextended	sx
Ultra	ub	Extended	x
Ultra Bold	ub	Expanded	e
Black	k	Wide	w
Extra Black	ek		
Ultra Black	uk		
Heavy	h	SHAPE	
Poster	r		
		Roman, Upright	n [2]
		Italic	it
		Cursive, Kursiv	it
		Oblique	sl [3]
		Slanted	sl [3]
		Incline(d)	sl [3]

Notes:

- [1] When *both* weight and width are empty, the “series” attribute becomes “m”.
- [2] Adobe Silentium Pro contains two “Roman” shapes (“RomanI” and “RomanII”); the first of these is mapped to “n”, the second one to “it”.
- [3] New in release 2014-01-21; before that, slanted fonts were mapped to “it”.

#### A note for MiKTeX users

Automatically installing the fonts into a suitable TEXMF tree (as **autoinst** does by default) requires a TeX-installation that uses the *kpathsea* library; with TeX distributions that implement their own directory searching (such as MiKTeX), **autoinst** will complain that it cannot find the *kpsewhich* program and install all generated files into subdirectories of the current directory. If you use such a TeX distribution, you should either move these files to their correct destinations by hand, or use the *-target* option (see “COMMAND-LINE OPTIONS” below) to specify a TEXMF tree.

Also, some OpenType fonts may lead to *pl* and *vpl* files that are too big for MiKTeX’s *pltotf* and *vptovf*; the versions that come with W32TeX (<http://www.w32tex.org>) and TeXLive (<http://tug.org/texlive>) don’t have this problem.

## COMMAND-LINE OPTIONS

**autoinst** tries hard to do The Right Thing (TM) by default, so in many cases you won't need these options; but most aspects of its operation can be changed if you want to.

You may use either one or two dashes before options, and option names may be shortened to a unique prefix (e.g., **-encoding** may be abbreviated to **-enc** or even **-en**, but **-e** is ambiguous (**-encoding**, **-extra**).

### **-dryrun**

Don't actually generate any fonts and files, only create a logfile showing which fonts would be generated. By default, this information is written to *autoinst.log*; use the **-logfile** option to specify a different filename.

### **-encoding=encoding[,encoding]**

Generate the specified encoding(s) for the text fonts. The default is "OT1,T1,LY1". For each encoding, a file *<encoding>.enc* (lowercase) should be somewhere where *otftotfm* can find it. Suitable encoding files for OT1, T1/TS1 and LY1 come with **autoinst**. (These files are called *fontools\_ot1.enc* etc. to avoid name clashes with other packages; the "fontools\_" prefix may be omitted.)

Multiple text encodings can be specified as a comma-separated list: **-encoding=OT1,T1**. The encodings are passed to *fontenc* in the order specified, so the last one will be the default text encoding.

### **-ts1 / -nots1**

Control the creation of TS1-encoded fonts. The default is **-ts1** if the text encodings (see *-encoding* above) include T1, **-nots1** otherwise.

### **-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.

### **-lining / -nolining**

Control the creation of fonts with lining figures. The default is **-lining**.

### **-oldstyle / -nooldstyle**

Control the creation of fonts with oldstyle figures. The default is **-oldstyle**.

### **-proportional / -noproportional**

Control the creation of fonts with proportional figures. The default is **-proportional**.

### **-tabular / -notabular**

Control the creation of fonts with tabular figures. The default is **-tabular**.

### **-smallcaps / -nosmallcaps**

Control the creation of small caps fonts. The default is **-smallcaps**.

### **-swash / -noswash**

Control the creation of swash fonts. The default is **-swash**.

**–titling / –notitling**

Control the creation of titling fonts. The default is **–titling**.

**–superiors / –nosuperiors**

Control the creation of fonts with superior characters. The default is **–superiors**.

**–inferiors / –noinferiors**

Control the creation of fonts with inferior digits. The default is **–noinferiors**.

**–fractions / –nofractions**

Control the creation of fonts with numerators and denominators. The default is **–nofractions**.

**–ornaments / –noornaments**

Control the creation of ornament fonts. The default is **–ornaments**.

**–verbose**

Verbose mode; print detailed information about which fonts **autoinst** is generating. By default, this information is written to *autoinst.log*; a different filename can be specified using the **–logfile** option. Repeat this option for even more detailed information.

**–logfile=LOGFILE**

Write the logging information to *LOGFILE* instead of *autoinst.log*.

**–defaultlining / –defaultoldstyle****–defaulttabular / –defaultproportional**

Tell **autoinst** which figure style is the current font family’s default (i.e., which figures you get when you don’t specify any OpenType features).

*Don’t use these options unless you are certain you need them! They are only needed for fonts that don’t provide OpenType features for their default figure style; and even in that case, the default values (**–defaultlining** and **–defaulttabular**) are usually correct.*

**–figurekern / –nofigurekern**

Some fonts provide kerning pairs for tabular figures. This is very probably not what you want (e.g., numbers in tables won’t line up exactly). The option **–nofigurekern** adds extra **–ligkern** options to the commands for *otftotfm* to suppress such kerns (but of course only for the families with tabular figures). Since this leads to very long commands (one hundred such options in total!) and the problem only occurs in very few fonts, the default is **–figurekern**.

**–extra=text**

Add *text* to the command line to *otftotfm*. To prevent *text* from accidentally being interpreted as options to **autoinst**, it should be properly quoted.

**–manual**

Manual mode. By default, **autoinst** executes all *otftotfm* commands it generates; with the **–manual** option, these commands are instead written to a file *autoinst.bat*. Also, the generated *otftotfm* commands specify the **–pl** option (which tells *otftotfm* to generate human readable/editable *pl* and *vp* files instead of the default *tfm* and *vf* files) and omit the **–automatic** option (which causes *otftotfm* to leave all generated files in the current directory, rather than install them into your TEXMF tree).

When using this option, you should run *pltotf* and *vptovf* after executing all commands, to convert the *pl* and *vf* files to *tfm* and *vf* format.

The following options are only meaningful in automatic mode, and hence ignored in manual mode:

**–target=***DIRECTORY*

Install all generated files into the TEXMF tree at *DIRECTORY*.

By default, **autoinst** searches your \$TEXMFLOCAL and \$TEXMFHOME paths and installs all files into subdirectories of the first writable TEXMF tree it finds (or into subdirectories of the current directory, if no writable directory is found).

**–vendor=***VENDOR*

**–typeface=***TYPEFACE*

These options are equivalent to *otftotfm*’s **–vendor** and **–typeface** options: they change the “vendor” and “typeface” parts of the names of the subdirectories in the TEXMF tree where generated files will be stored. The default values are “lcdftools” and the font’s FontFamily name.

Note that these options change *only* directory names, not the names of any generated files.

**–updmap / –noupdmap**

Control whether or not *updmap* is called after the last call to *otftotfm*. The default is **–updmap**.

## SEE ALSO

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

**Perl** can be downloaded from <http://www.perl.org>; it is pre-installed on many Linux distributions. For Windows, try ActivePerl (<http://www.activestate.com>) or Strawberry Perl (<http://strawberryperl.com>).

The **FontPro** project (<https://github.com/sebschub/FontPro>) offers very complete LaTeX support for Adobe’s Minion Pro and Myriad Pro (including math), and is currently working on Cronos Pro.

**XeTeX** (<http://www.tug.org/xetex>) and **LuaTeX** (<http://www.luatex.org>) are TeX engines that can use fonts in many formats (including both flavours of OpenType) without TeX-specific support files.

John Owens’ **otfinst** (available from CTAN) is another wrapper around *otftotfm*.

## AUTHOR

Marc Penninga <marcpenninga@gmail.com>

When sending a bug report, please give as much relevant information as possible; this includes at least (but may not be limited to) the output from running **autoinst** with the **–verbose** option. Please include all (if any) error messages as well.

## COPYRIGHT

Copyright (C) 2005–2015 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 text of the GNU General Public License is included in the *fontools* distribution; 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.

**RECENT CHANGES**

(See the source code for the rest of the story.)

*2015-05-13* Fixed an error message that mixed up width and weight.

*2014-04-04* Fixed a bug in the font info parsing code.

*2014-01-21* “Oblique” or “slanted” fonts are now mapped to NFSS code “sl” instead of “it”; added “ssub” rules to the <fd> files to substitute slanted fonts for italic ones if the latter are missing. Fixed a few bugs.

*2014-01-03* Added the *-dryrun* and *-logfile* options; changed which info is logged. Added the *-lining*, *-oldstyle*, *-tabular* and *-proportional* options; the old options with those names have been renamed to *-defaultlining*, *-defaultoldstyle* etc.