NAME

autoinst – wrapper around the LCDF TypeTools, for installing and using OpenType fonts in (La)TeX.

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, each in both tabular and proportional variants), all with the following shapes:

n	Roman (i.e., upright) text
it, *sl*	Italic and slanted (sometimes called oblique) text
sc	Small caps
sw	Swash
tl	Titling shape. Meant for all-caps text; 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.)
scit, *scsl*	Italic and slanted small caps
nw	"Upright swash"; usually roman text with a few "oldstyle" ligatures like ct, sp and st.
tlit, *tlsl*	Italic and slanted titling text

– For each T1–encoded text family: a family of TS1–encoded symbol fonts, in roman, italic and slanted shapes.

– Families with superiors, inferiors, numerators and denominators, in roman, italic and slanted shapes.

– An ornament family, also in roman, italic and slanted shapes.

Of course, if your fonts don't contain italics, oldstyle digits, small caps etc., the corresponding shapes and families are not created. In addition, the creation of most families and shapes can be controlled by the user (see "COMMAND-LINE OPTIONS" below).

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

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

The individual 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 (also in lowercase). A typical name in this scheme would be "FiraSans−Light−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 accommodate the additional ligatures found in many OpenType fonts. Other encodings can be chosen using the *−encoding* option (see "COMMAND-LINE OPTIONS" below).

Using the fonts in your LaTeX documents

autoinst generates a style file for using the fonts in LaTeX documents, named *<FontFamily>.sty*. This style file also takes care of loading the *fontenc* and *textcomp* packages. To use the fonts, add the command `\usepackage{<FontFamily>}` to 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>*. E.g., 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.

`light`, `medium`, `book`, `text`, `regular`
  Select the weight that LaTeX will use as the "regular" weight; the default is `regular`.

`heavy`, `ultrablack`, `extrablack`, `black`, `ultrabold`, `extrabold`, `demibold`, `semibold`, `bold`
  Select the weight that LaTeX will use as the "bold" weight; the default is `bold`.

The previous 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) to access titling, superior and inferior characters:

```
DECLARATION        COMMAND            SHORT FORM OF COMMAND


\tlshape           \texttitling       \texttl
\sufigures         \textsuperior      \textsu
\infigures         \textinferior      \textin
```

In addition, the `\swshape` and `\textsw` commands are redefined to place swash on *fontaxes*' secondary shape axis (*fontaxes* places it on the primary shape axis) to make them 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 ornament glyphs, **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 was turned off by the user. Also these commands are built on top of *fontaxes*, so if that package cannot be found, you're limited to using the lower-level commands from standard NFSS (`\fontfamily`, `\fontseries`, `\fontshape` etc.).

NFSS codes

NFSS identifies fonts by a combination of family, series (the concatenation of weight and 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, and so 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 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 beginning 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 on the standard *Fontname* scheme and Philipp Lehman's *Font Installation Guide*, but some changes had to be made to avoid name clashes in font families with many widths and weights, such as Helvetica Neue and Fira Sans.

```
WEIGHT                                  WIDTH
-----------------------                 -------------------------------
Two            2                        Ultra Compressed      up
Four           4                        Extra Compressed      ep
Eight          8                        Compressed            p
Hair           a                        Compact               p
Thin           i                        Ultra Condensed       uc
Ultra Light    ul                       Extra Condensed       ec
Extra Light    el                       Condensed             c
Light          l                        Narrow                n
Regular        -    [1]                 Semicondensed         sc
Text           t    [2]                 Regular               -       [1]
Book           o    [2]                 Semiextended          sx
Medium         mb                       Extended              x
Demibold       db                       Expanded              e
Semibold       sb                       Wide                  w
Bold           b
Extra Bold     eb                       SHAPE
Ultra (Bold)   ub                       -------------------------------
Black          k                        Roman, Upright        n       [3]
Extra Black    ek                       Italic, Cursive,
Ultra Black    uk                          Kursiv             it
Heavy          h                        Oblique, Slanted,
Poster         r                           Incline(d)         sl
```

[1]   When *both* weight and width are empty, the NFSS "series" attribute becomes "m".

[2]   Until release 2017–06–16, "Text" and "Book" were treated as synonyms for "Regular". As there are some fonts (IBM Plex, Fira Sans) that contain separate "Text" or "Book" in addition to "Regular" weights, I decided to give them their own codes. When there is no "Regular" weight, **autoinst** will generate *ssub* rules to substitute either the "Text" or the "Book" font in its place.

[3]   Adobe Silentium Pro contains two roman shapes; "Roman I" is mapped to "n", "Roman II" to "it".

A note for MiKTeX users

Automatically installing the fonts into a suitable TEXMF tree (as **autoinst** tries to do 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 seem to have this problem.

COMMAND-LINE OPTIONS

autoinst tries hard to do The Right Thing (TM) by default, so you usually won't really need these options; but most aspects of its operation can be fine-tuned 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 (it may mean either **–encoding** or **–extra**)).

–dryrun
> Don't generate any output files; only parse the input fonts and create *autoinst.log* showing which fonts would have been generated.

–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* (in all *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 (without spaces!). The generated style file passes these encodings to *fontenc* in the specified order, so the last one will become the default text encoding for your documents.

–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=[**sinf** | **subs** | **dnom**]
> The OpenType standard defines several kinds of digits that might be used as inferiors or subscripts: ''Scientific Inferiors'' (OpenType feature ''sinf''), ''Subscripts'' (''subs'') and ''Denominators''

("dnom"). This option allows the user to determine which of these styles **autoinst** should use for the inferior characters. The default is not to create fonts with inferior characters.

*Note that many fonts contain only one (or even none) of these types of inferior characters. If you specify a style of inferiors that isn't actually present in the font, **autoinst** silently falls back to its default of not creating fonts with inferiors; it doesn't try to substitute one of the other features.*

–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**.

–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, **autoinst**'s default values (**–defaultlining** and **–defaulttabular**) are usually correct.

–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). This option adds extra *−−ligkern* options to the commands for *otftotfm* to suppress such kerns. Note that this option leads to very long commands (it adds one hundred *−−ligkern* options), which may cause problems on some systems.

–mergewidths / **–nomergewidths**
Some font families put Condensed and Extended fonts in separate families; this option tells **autoinst** to merge those separate families into the "main" font family. The default is **–nomergewidths**.

–extra=*text*
Append *text* as extra options to the command lines for *otftotfm*. To prevent *text* from accidently being interpreted as options to **autoinst**, it should be properly quoted.

–manual
Manual mode. By default, **autoinst** immediately executes all *otftotfm* commands it generates; with the **–manual** option, these commands are instead written to a file *autoinst.bat*. Furthermore it adds the *−−pl* option (which tells *otftotfm* to generate human readable/editable *pl* and *vpl* files instead of the default *tfm* and *vf* files) and omits the *−−automatic* option (which causes *otftotfm* to leave all generated files in the current directory, rather than install them into your TEXMF tree). Manual mode is meant to enable tweaking the generated commands and post-processing the generated files.

When using this option, run *pltotf* and *vptovf* after executing the commands (to convert the *pl* and *vf* files to *tfm* and *vf* format) and move all generated files to their proper destinations.

All 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*. This option allows the user to override **autoinst**'s default behaviour, which is to search the `$TEXMFLOCAL` and `$TEXMFHOME` paths and install 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 obtained from *http://www.perl.org*; it is included in most Linux distributions. For Windows, try ActivePerl (*http://www.activestate.com*) or Strawberry Perl (*http://strawberryperl.com*).
>
> **XeTeX** (*http://www.tug.org/xetex*) and **LuaTeX** (*http://www.luatex.org*) are Unicode-aware TeX engines that can use OpenType fonts directly, without any (La)TeX−specific support files.
>
> The **FontPro** project (*https://github.com/sebschub/FontPro*) offers very complete LaTeX support (even for typesetting maths) for Adobe's Minion Pro, Myriad Pro and Cronos Pro font families.

AUTHOR

> Marc Penninga (*marcpenninga@gmail.com*)
>
> When sending a bug report, please give as much relevant information as possible. If you see any error messages (either from **autoinst** itself, from the *LCDF TypeTools*, from Perl or from the OS), include these *verbatim*; don't paraphrase.

COPYRIGHT

> Copyright (C) 2005−2018 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 for the full story, all the way back to 2005.)
>
> *2018−03−26* Added the "Text" weight and the *–(no)mergewidths* option. Changed the NFSS codes for "Thin" and "Book" to "i" and "o", respectively. Tried to improve the documentation.
>
> *2018−01−09* Added the "sl" weight for font families (such as Fira Sans) that contain both "Book" and "Regular" weights (reported by Bob Tennent). Added the "Two", "Four", "Eight" and "Hair" weights (for Fira Sans).
>
> *2017−06−16* Changed the *–inferiors* option from a binary yes-or-no choice to allow the user to choose one of the "sinf", "subs" and "dnom" features. **autoinst** now always creates a log file.
>
> *2017−03−21* Updated the *fontools_ot1.enc* encoding file to include the "Lslash" and "lslash" glyphs (thanks to Bob Tennent).