NAME
autoinst − wrapper around the LCDF TypeTools, for installing and using OpenType fonts in LaTeX.

SYNOPSIS
autoinst −*help*

autoinst [*options*] **font(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
scit, *scsl*	Italic and slanted small caps
sw	Swash
nw	"Upright swash"

– 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.

– Families with "Titling" characters; these *"... replace the default glyphs with corresponding forms designed specifically for titling. These may be all-capital and/or larger on the body, and adjusted for viewing at larger sizes"* (according to the OpenType Specification).

– 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
Titl	Titling characters; see above
Orn	ornaments
Numr, *Dnom*	numerators and 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" or "swash", and *‹enc›* is the encoding (also in lowercase). A typical name in this scheme would be *FiraSans−Light−osf−sc−ly1*.

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 loads the *fontenc* and *textcomp* packages, if necessary. To use the fonts, add the command `\usepackage{<FontFamily>}` to the preamble of your document.

This style file has a few options:

`mainfont`
Redefine `\familydefault` to make this font the main font for the document. This is a no-op if the font is installed as a serif font; but if the font is installed as a sanserif or typewriter font, this option saves you from having to redefine `\familydefault` yourself.

`lining`, `oldstyle`, `tabular`, `proportional`
Choose which figure style to use. The defaults are "oldstyle" and "proportional" (if available).

`scale=<factor>`, `scale=MatchLowercase`
Scale the font by *<factor>*; as an example, `\usepackage[scale=1.05]{<FontFamily>}` will increase the font's size by 5%. The special value `MatchLowercase` may be used to scale the font so that its x−height matches that of the current main font (which is usually Computer Modern Roman, unless you have loaded another font package before this one). The word "scale" may also be spelled as "scaled".

`medium`, `book`, `text`, `normal`, `regular`
Select the weight that LaTeX will use as the "regular" weight.

`heavy`, `black`, `extrabold`, `demibold`, `semibold`, `bold`
Select the weight that LaTeX will use as the "bold" weight.

The last two groups of options will only work if you have the *mweights* package installed. The default here is not to change LaTeX's default, i.e. use the "m" and "b" weights.

The style file will also try to load the *fontaxes* package (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         \textttl
\supfigures        \textsuperior        \textsup, \textsu
\inffigures        \textinferior        \textinf, \textin
```

In addition, the existing `\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.

Finally, the style file redefines Latex's `\textsuperscript` and `\textsubscript` commands to use the fonts' superior and inferior figures, and modifies Latex's footnote mechanism to use `\textsuperscript` instead of reduced-size numerals from the regular text font. The old versions of these commands are still available as `\textsuperscript*` and `\textsubscript*`.

There are no commands for accessing the numerator and denominator fonts; these can be selected using *fontaxes*' standard commands, e.g., `\fontfigurestyle{numerator}\selectfont`.

Once again: all these commands are only generated for existing shapes and number styles; no commands are generated for shapes and styles that are missing from your fonts. Note that all these commands are built on top of *fontaxes*; if that package cannot be found, you're limited to using lower-level commands from standard NFSS (`\fontfamily`, `\fontseries`, `\fontshape` etc.).

By default, **autoinst** generates text fonts with OT1, LY1 and T1 encodings, and the generated style files use T1 as the default text encoding. Other encodings can be chosen using the *−encoding* option (see "COMMAND-LINE OPTIONS" below).

Maths

This is an experimental feature; **USE AT YOUR OWN RISK!** Test the results thoroughly before using them in real documents, and be warned that future versions of **autoinst** may introduce incompatible changes.

The −*math* option tells **autoinst** to generate basic math fonts. When enabled, the generated style file defines a few extra options to access these math fonts:

`math`
> Use these fonts for the maths in your document.

`mathlining`, `matholdstyle`, `mathtabular`, `mathproportional`
> Choose which figure style and alignment to use in maths. The defaults are "mathlining" and "mathtabular".

`mathcal`
> Use the swash characters from these fonts as the `\mathcal` alphabet. (This option will only exist if your fonts actually contain both swash characters and a `swsh` feature to access them).

`nomathgreek`
> Don't redeclare greek letters in math.

`math-style=<style>`
> Choose the "math style" to use. With `math-style=ISO`, all latin and greek letters in math are italic; with `math-style=TeX` (the default), uppercase greek is upright; with `math-style=french`, all greek as well as uppercase latin is upright; and with `math-style=upright` all letters are upright.

The style file also defines commands `\up...` and `\it...` to access upright and italic greek characters regardless of the chosen math style: `\upGamma`, `\upalpha`, `\itDelta`, `\itbeta` etc.

Note that this `math` option only affects digits, latin and greek letters, plus a few basic punctuation characters; all other mathematical symbols, operators, delimiters etc. are left as they were before. If you don't want to use TeX's default versions of those symbols, load another math package (such as *mathdesign* or *newtxmath*) before loading the **autoinst**−generated style file.

Finally, note that **autoinst** doesn't check if your fonts actually contains all of the required characters; it just assumes that they do and sets up the style file accordingly. Even if your fonts do contain greek, characters such as `\varepsilon` may be missing. You may also find that some glyphs *are* present in your fonts, but don't work well in equations or don't match other symbols; edit the generated style file to remove the declarations of these offending characters. Once again: test the results before using them! If the characters themselves are fine but spaced too tightly, you may try increasing the side bearings in math fonts with the −*mathspacing* option (see below), e.g. -mathspacing=50.

NFSS codes

LaTeX's New Font Selection System (NFSS) identifies fonts by a combination of family, series (the concatenation of weight and width), shape and size. **autoinst** parses the font's metadata to determine these parameters. When this fails (usually because the font family contains uncommon weights, widths or shapes), **autoinst** ends up with multiple fonts having the same values for these font parameters; such fonts cannot be used in NFSS, since there's no way distinguish them. When **autoinst** detects such a situation, it will print an error message and abort. If that happens, either rerun **autoinst** on a smaller set of fonts, or add the missing widths, weights and shapes to the tables `@WIDTH`, `@WEIGHT` and `%SHAPE` in the source code. Please also send a bug report (see AUTHOR below).

The mapping of shapes to NFSS codes is done using the following table:

```
SHAPE                                   CODE
──────────────────────────────         ────
Roman, Upright                          n
Italic                                  it
Oblique, Slant(ed), Incline(d)          sl
```

(*Exception:* Adobe Silentium Pro contains two Roman shapes; we map the first of these to n, for the second

one we (ab)use the `it` code as this family doesn't contain an Italic shape.)

For weights and widths, **autoinst** tries to the standard NFSS codes (ul, el, l, sl, m, sb, b, eb and ub for weights; uc, ec, c, sc, m, sx, x, ex and ux for widths) as much as possible. Of course, not all 81 combinations of these NFSS weights and widths will map to existing fonts; and conversely it may not be possible to assign every existing font a unique code in a sane way (especially for the weights, some font families offer more variants than NFSS's codes can handle; e.g., Fira Sans contains fifteen different weights!). Therefore every font is also assigned a "series" name that is the concatenation of its weight and width, after expanding any abbreviations and converting to lowercase. A font of "Cond" width and "Ultra" weight will then be known as "ultrablackcondensed".

The exact mapping between fonts and NFSS codes can be found in the generated *fd* files and in the log file (you may want to run **autoinst** with the −*dryrun* option to check the chosen mapping beforehand). The −*nfssweight* and −*nfsswidth* command-line options can be used to finetune the mapping between NFSS codes and fonts.

To access specific weights or widths, one can always use the `\fontseries` command with the full series name (i.e., `\fontseries{demibold}\selectfont`).

Ornaments

Ornament fonts are regular LY1−encoded fonts, with a number of "regular" characters replaced by ornament glyphs. The OpenType specification says that fonts should only put their ornaments in place of the lowercase ASCII letters, but some fonts put them in other positions (such as those of the digits) as well.

Ornaments can be accessed like `{\ornaments a}` and `{\ornaments\char"61}`, or equivalently `\textornaments{a}` and `\textornaments{\char"61}`. To see which ornaments a font contains (and at which positions), run LaTeX on the file *nfssfont.tex* (which is included in any standard LaTeX installation), supply the name of the ornament font (i.e., `GaramondLibre-Regular-orn-u`) and give the command `\table\bye`; this will create a table of all glyphs in that font.

Note that versions of **autoinst** up to 20200428 handled ornaments differently, and fonts and style files generated by those versions are not compatible with files generated by newer versions.

WARNINGS AND CAVEATS

OpenType fonts and licensing issues

Since *pdfTeX* cannot subset otf-flavoured OpenType fonts, *otftotfm* will convert such fonts to Type1 (pfb) format. However, many fonts (at least those licensed under the SIL Open Font License) do not allow redistributing such converted versions under their original name.

In order to try to meet such licensing requirements, **autoinst** provides a −*t1suffix* command-line option that appends a suffix to the names (both the filename and the internal font name) of all generated Type1 fonts; see "COMMAND-LINE OPTIONS" below.

Please note that I am not a lawyer and do not guarantee that this suffix is sufficient to meet the license's requirements. When in doubt, consult a real lawyer!

Sorry, LIGTABLE too long for me to handle

The LIGTABLE in TeX's *tfm* files, which contains a font's ligatures and kerning pairs, is limited to about 32,500 entries ($2^{15} - 256$). If the number of ligatures plus kerns in a font is higher than that limit, *pltotf* and *vptovf* will complain loudly and ignore the excess entries. This happens at least with Adobe's Source Serif 4 and Minion 3 font families. Even when *pltotf* and *vptovf* don't warn about the LIGTABLE's size, you may still find that *pdftex* crashes with a `Bad metric (TFM) file` error. The best way to handle this situation is to use **autoinst**'s `-extra` option to raise *otftotfm*'s value for the `--min-kern` parameter, which causes it to ignore small kerning pairs: `-extra='--min-kern=6.0'`. Finding the correct value for the `--min-kern` parameter may require some trial and error.

A note for MiKTeX users

Automatically installing the fonts into a suitable TEXMF tree (as **autoinst** tries to do by default) only works for TeX-installations that use 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 move all generated files into a subdirectory `autoinst_output/` 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 manually specify a TEXMF tree.

Also, some OpenType fonts contain so many kerning pairs that the resulting *pl* and *vpl* files 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.

A note for MacTeX users

By default, **autoinst** will try to install all generated files into the $TEXMFLOCAL tree; when this directory isn't user-writable, it will use the $TEXMFHOME tree instead. Unfortunately, MacTeX's version of updmap-sys doesn't search in $TEXMFHOME, and hence MacTeX will not find the new fonts.

To remedy this, either run **autoinst** as root (so that it can install everything into $TEXMFLOCAL) or manually run updmap -user to tell TeX about the files in $TEXMFHOME. This latter option does, however, come with some caveats; see *https://tug.org/texlive/scripts−sys−user.html*.

COMMAND-LINE OPTIONS

autoinst tries hard to do The Right Thing (TM) by default, so you usually won't 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**)).

General options

−help

> Print a (relatively) short help text and exit.

−dryrun

> Don't generate output; just parse input fonts and write the results to the log file.

−verbose

> Add more details to the log file.

−version

> Print **autoinst**'s version number and exit.

Font creation options

−encoding=*encoding[,encoding]*

> Generate the specified encoding(s) for the text fonts. Multiple encodings may be specified as a comma-separated list (without spaces!); the default choice of encodings is OT1,LY1,T1.

> For each encoding argument, **autoinst** will first check if it is the filename of an encoding file, and if found it will use that; otherwise the argument is assumed to be the name of one of the built-in encodings. Currently **autoinst** comes with built-in support for the OT1, T1/TS1, LY1, T2A/B/C, T3/TS3, T4, T5, LGR, CS, L7X and QX encodings. (These files are called *fontools_ot1.enc* etc. to avoid name clashes with other packages; the *fontools_* prefix may be omitted.)

−ts1/−nots1

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

−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 families. The default is **–titling**.

–superiors/–nosuperiors

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

–inferiors [**= none** | **auto** | **subs** | **sinf** | **dnom**]
–noinferiors

 The OpenType standard defines several kinds of digits that might be used as inferiors or subscripts: "Subscripts" (OpenType feature subs), "Scientific Inferiors" (sinf), and "Denominators" (dnom). This option allows the user to determine which of these styles **autoinst** should use for the inferior characters. Alternatively, the value "auto" tells **autoinst** to use the first value in sinf, subs or dnom that is supported by the font. Saying just **–inferiors** is equivalent to **–inferiors=auto**; otherwise the default is **–noinferiors**.

 *If you specify a style of inferiors that isn't present in the font, **autoinst** will fall back to its default behaviour of not creating fonts with inferiors at all; it won't try to substitute one of the other styles.*

–fractions/–nofractions

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

–ligatures/–noligatures

 Some fonts contain glyphs for the standard f–ligatures (ff, fi, fl, ffi, ffl), but don't provide a liga feature to access these. This option tells **autoinst** to add extra LIGKERN rules to the generated fonts to enable the use of these ligatures. The default is **–ligatures**, except for typewriter fonts.

 Specify **–noligatures** to disable generation of ligatures even for fonts that do contain a liga feature.

–ornaments/–noornaments

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

–serif/–sanserif/–typewriter

 Install the font as a serif, sanserif or typewriter font, respectively. This changes how you access the font in LaTeX: with \rmfamily/\textrm, \sffamily/\textsf or \ttfamily/\texttt.

 Installing the font as a typewriter font will cause two further changes: it will – by default – turn off the use of f–ligatures (though this can be overridden with the *–ligatures* option), and it will disable hyphenation for this font. This latter effect cannot be re-enabled in **autoinst**; if you want typewriter text to be hyphenated, use the *hyphenat* package.

 If none of these options is specified, **autoinst** tries to guess: if the font's filename contains the string "mono" or if the field isFixedPitch in the font's post table is True, it will select **–typewriter**; else if the filename contains "sans" it will select **–sanserif**; otherwise it will opt for **–serif**.

–math

 Tells **autoinst** to create basic math fonts (see above).

–mathspacing=*amount*

 Letterspace each character in the math fonts by *amount* units, where 1000 units equal one em. In my opinion, many text fonts benefit from letterspacing by 50 to 100 units when used in maths; some fonts need even more. Use your own judgement!

Output options

–t1suffix [**=** *SUFFIX*]

 Tell **autoinst** to modify the font names of all generated Type1–fonts, by adding *SUFFIX* to the family name. If you use this option without specifying a *SUFFIX* value, **autoinst** will use the value "PS". The default behaviour when this option is not given is to not modify font names at all.

See also ''OpenType fonts and licensing issues'' in ''WARNINGS AND CAVEATS'' above.

−target=*DIRECTORY*

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

By default, **autoinst** searches the `$TEXMFLOCAL` and `$TEXMFHOME` trees and installs all files into the first user-writable TEXMF tree it finds. If **autoinst** cannot find such a user-writable directory (which shouldn't happen, since `$TEXMFHOME` is supposed to be user-writable) it will print a warning message and put all files into the subdirectory `autoinst_output/` of the current directory. It's then up to the user to move the generated files to a better location and update all relevant databases (usually by calling *texhash* and *updmap*).

−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. These options change *only* directory names, not the names of any generated files.

−logfile=*filename*

Write log data to *filename* instead of the default *<fontfamily>.log*. If the file already exists, **autoinst** appends to it; it doesn't overwrite an existing file.

Specialist options

−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.

−nfssweight=*code=weight*
−nfsswidth=*code=width*

Map the NFSS code *code* to the given weight or width, overriding the built-in tables. Each of these options may be given multiple times, to override more than one NFSS code. Example: to map the `ul` code to the ''Thin'' weight, use `−nfssweight=ul=thin`. To inhibit the use of the `ul` code completely, use `−nfssweight=ul=`.

−extra=*extra options*

Pass *extra options* to the commands for *otftotfm*. To prevent *extra options* from accidently being interpreted as options to **autoinst**, they should be properly quoted.

−manual

Manual mode; for users who want to post-process the generated files and commands. By default, **autoinst** immediately executes all *otftotfm* commands it generates; in manual mode, these are instead written to a file *autoinst.bat*. Furthermore it tells *otftotfm* to generate human readable (and editable) *pl/vpl* files instead of the default *tfm/vf* ones, and to place all generated files in a subdirectory `./autoinst_output/` of the current directory, rather than install them into your TeX installation.

When using this option, you need to execute the following manual steps after **autoinst** has finished:

− run *pltotf* and *vptovf* on the generated *pl* and *vf* files, to convert them to *tfm/vf* format;
− move all generated files to a proper TEXMF tree, and, if necessary, update the filename database;
− tell TeX about the new *map* file (usually by running `updmap` or similar).

Note that some options (*−target*, *−vendor* and *−typeface*) are meaningless, and hence ignored, in manual mode.

Also note that font name modification doesn't happen in manual mode.

−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; hence it is not active by default.

SEE ALSO

Eddie Kohler's **TypeTools** and **T1Utils** (*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*).

LuaTeX (*http://www.luatex.org*) and **XeTeX** (*http://www.tug.org/xetex*) 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; this usually includes (but may not be limited to) the log file (please add the −*verbose* command-line option, for extra info). If you see any error messages, please include these *verbatim*; don't paraphase.

COPYRIGHT

Copyright (C) 2005−2023 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.

VERSION

This document describes **autoinst** version 20231230.

RECENT CHANGES

(See the source for the full story, all the way back to 2005.)

2023−12−30   Bugfix: font info parsing now works for Junicode 2.

2023−09−14   Added the \up... and \it... commands to the style file.

2023−06−28   Adapted font metadata parsing for Adobe Source Serif 4.

2023−02−01   Added support for the T4, T5, CS, L7X and QX encodings.

2023−01−31   If the fonts contain superior and/or inferior figures, the generated style file now redefines the \textsuperscript and \textsubscript commands, and patches Latex's footnote mechanism to use these figures (inspired by the *realscripts* package). Fixed a few bugs in metadata parsing, style file generation and the −t1suffix option, so that the latter also works for dvips and dvipdfmx.