NAME

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

SYNOPSIS

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

DESCRIPTION

   Eddie Kohler's *TypeTools*, mainly *otftotfm*, are great tools for installing OpenType fonts for use with LaTeX, but their use (even in automatic mode) is quite complicated; they need lots of long command lines and don'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 the user then needs 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 family of font files (in either *.ttf* or *.otf* format), **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 punctuation 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; furthermore, the creation of most families and shapes can be controlled by command-line options (see below).

   The generated font families are named *<FontFamily>−<Suffix>*, where *<Suffix>* is one of

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 (many fonts only have an incomplete set of superiors: figures, some punctuation and the letters *abdeilmnorst*; normal forms will be used for the other characters)

Inf   inferior characters; usually only figures and punctuation, normal forms for the other characters

Orn   ornaments

Numr   numerators

Dnom   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 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 your 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).

ultrablack, ultrabold, heavy, extrablack, black, extrabold, demibold, semibold, bold
    Choose the weight that LaTeX will use for the 'bold' weight (i.e., the value of `\bfdefault`).

light, medium, regular
    Choose the weight that LaTeX will use for the 'regular' weight (i.e., the value of `\mddefault`).

scaled=<scale>
    Scale the font by a factor of *<scale>*. For example: to increase the size of the font by 5%, use the command `\usepackage[scaled=1.05]{MyriadPro}`.

This option is only available when the *xkeyval* package is found in your TeX installation.

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 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


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

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* (part of a standard LaTeX installation) 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 high-level access to the more 'exotic' font shapes and styles. In that case, you're limited to using the lower-level commands from standard NFSS, or even plain TeX's `\font` primitive (and it's called 'primitive' for a reason!)

Using multiple font families in one document

If you want to use several font families in one document, make sure all fonts were installed using the same version of **autoinst**. **autoinst** has seen some non-backward-compatible changes in the past, and .*sty* and .*fd* files that were generated by different versions of **autoinst** may not be able to coexist peacefully.

NFSS codes

In NFSS, weight and width are concatenated into a single 'series' attribute. (*Note:* versions of **autoinst** before 2007−07−27 erroneously formed the series as 'width plus weight' instead of the reverse.) **autoinst** maps widths, weights and shapes to NFSS codes using the following tables. These are based on the tables in Lehman's *Font Installation Guide*, but some changes had to be

made to avoid name clashes for font families with many different widths and weights (such as Helvetica Neue).

WEIGHT			WIDTH		
Thin	t		Ultra Condensed	uc	
Ultra Light	ul		Extra Condensed	ec	
Extra Light	el		Condensed	c	
Light	l		Semicondensed	sc	
Book		[1]	Regular		[1]
Regular		[1]	Semiextended	sx	
Medium	mb		Extended	x	
Demibold	db				
Semibold	sb				
Bold	b				
Extra Bold	eb		SHAPES		
Black	a				
Extra Black	ea		Roman	n	
Heavy	h		Italic	it	
Ultra	ub		Oblique	it	[2]
Ultra Bold	ub		RomanI	n	[3]
Ultra Black	ua		RomanII	it	[3]

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

[2] Mapping the 'Oblique' shape to 'it' instead of the canonical 'sl' simplifies **autoinst**. Since font families with both 'Italic' and 'Oblique' shapes probably do not exist (apart from Computer Modern), this shouldn't cause problems in real life.

[3] To the best of my knowledge, the only font family that has two 'Roman' shapes is Silentium; since this has no 'Italic' shape, the 'it' code is (ab)used for the 'RomanII' shape.

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 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 *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[,encoding]*

Use the encoding *encoding* for the text fonts. The default is 'LY1'. A file named '*<encoding>*.enc' (in all *lowercase*) should be somewhere where *otftotfm* can find it. Suitable encoding files for LY1, OT1 and T1/TS1 come with *fontools*. (Note that these files are called *fontools_xxx.enc* to avoid name clashes with other packages; the 'fontools_' prefix

doesn't need to be specified.)

Multiple text encodings can be specified as well: `--encoding=OT1,T1,LY1`. The encodings are passed to *fontenc* in the order specified, so the last one will be the default text encoding.

−−sanserif

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

−−typewriter

Install the font as a typewriter font, accessed via `\ttfamily` and `\texttt`. Note that the generated style file redefines `\familydefault`, so including it will 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 encodings (see *−−encoding* above) include 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 accidently being interpreted as options to **autoinst**, it should be properly quoted.

SEE ALSO

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

Perl is usually pre-installed on Linux and Unix systems; for Windows, good (and free) Perl implementations are Strawberry Perl (*http://strawberryperl.com*) and ActivePerl (available from *http://www.activestate.com*);

John Owens' *otfinst* (*http://www.ece.ucdavis.edu/˜jowens/code/otfinst/*; also available from CTAN) is another wrapper around *otftotfm*, and may work for you when **autoinst** doesn't.

Ready-made support files for MinionPro, providing more options and features 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 installed in the operating system (including both flavours of OpenType fonts) without additional support files. It also isn't hindered by standard TeX's limitation to 8−bit fonts, so it is especially well suited to fonts with many ligatures and alternate glyphs, such as Bickham, Poetica and Zapfino.

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. Please make sure that this output includes all error messages (if any); this can be done using the command

```
autoinst (... all options and files ...)  >autoinst.log 2>&1
```

COPYRIGHT

Copyright (c) 2005−2009 Marc Penninga.

LICENSE

This program is free software; you can redistribute it and/or modify it under the terms of version 2 of the GNU General Public License as published by the Free Software Foundation.

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.

RECENT CHANGES

(See the source code for the full story.)

2009−04−09 Prefixed the filenames of the included encoding files with 'fontools_', to prevent name clashes with other packages.

2009−04−06 A small patch to the make_ornament_encoding subroutine; it now also recognises the *bullet.xxx* ornament glyphs in Adobe's Kepler Pro.