

The typeface Package

A package for simplifying \LaTeX Type1 font setup

Geoffrey Jones Version 0.1 (preliminary release)
geoffrey.jones@uqconnect.edu.au 18 June 2012

Contents

1	Introduction	1	2.5	Font Scaling	12
1.1	About typeface	1	2.6	Font Load Order	13
1.2	License	2	2.7	Ancillary Packages	13
1.3	About this Manual	2	2.8	Package Testing	13
1.4	Package Contents	2	3	Author Commands	14
1.5	Installation	3	3.1	Text and Lining Figures	14
1.6	Typeface Exhibits File	3	3.2	Typeface Print Names	15
1.7	Prerequisite Conditions	3	4	Package Configuration	15
1.8	Auxiliary File	3	4.1	Typeface Options	15
2	Package Options	3	4.2	Ancillary Package Options	16
2.1	Options Syntax	4	4.3	<code>\newtypeface</code> Command	16
2.2	Default Options	4	4.4	Redefining Predefined Typefaces	18
2.3	Typeface Selection	4	5	Revision History	19
2.4	Text Figures	10			

1 Introduction

1.1 About typeface

The typeface package provides a relatively simple facility for setting up default roman, sans serif, teletype, math, symbols and lining/text figures Type1 fonts in \LaTeX documents. Its purpose is to make it as easy as possible to mix, match, scale, test and use whatever combinations you might like from the main Type1 font sets available at CTAN¹. Other Type1 fonts can be readily integrated into typeface package management by modifying the provided configuration file.

As a document font manager, typeface bears certain similarities to Will Robertson’s and Khaled Hosny’s more extensive fontspec package. While the role of typeface is to simplify configuring and loading document default Type1 fonts, fontspec’s purpose is to simplify most things connected with using OpenType and, to some extent, TrueType fonts. However, unlike fontspec which cannot be used under \LaTeX , with a few restrictions, typeface performs as advertised under \LaTeX and \XeTeX . That said, each package has its natural role and home environment—typeface for simplifying setting up default Type1 fonts in \LaTeX documents, and fontspec for managing OpenType fonts in \LaTeX and \XeTeX documents.

1.1.1 Motivation

This work is grounded in the observation that, despite the tremendous strides that have been made by more modern \TeX engines and compilation formats, \LaTeX^2 remains a document production mainstay for many individuals, institutions and publication houses. Given this and its relative stability, things will most likely remain this way for some time to come. However, the task of setting up fonts in \LaTeX documents all too frequently takes more effort than it reasonably should. The approaches for

1 <http://www.ctan.org/tex-archive/fonts>

2 By which I mean $\text{\LaTeX}2_{\epsilon}$, including reasonably strict supersets such as $\text{\LaTeX}3$.

loading fonts typically depend more on the programming styles of font package authors than on actual necessity—given, that is, that relevant font loading packages even exist. The Berry names³ for individual typefaces can prove frustrating to recall when using fonts that have not been thus supplied. The methods for scaling fonts can often prove exasperating to remember and apply. Worse, font scaling is often not included as a load-time option in font loading packages or font descriptor (.fd) files. For such a routine task, scaling one font to match another is seldom, if ever, automated. Notwithstanding the aesthetic appropriateness of the task, freely mixing and matching combinations of text and math typefaces can prove technically difficult to get right. Switching between text figures (12345) and lining figures (12345) is often so burdensome that \TeX users all too often give up, preferring to render body text, titles and adornments like page numbers equally, either in all lining figures or in all text figures.⁴ Nevertheless, the beauty of a finished \TeX document typeset with well-configured fonts can make memories of the frustrations that went into creating it quickly recede. The `fontspec` package overcomes many of these difficulties in the \LaTeX and \XeTeX worlds. This work sets out to reduce the effort of default Type1 font setup in the \TeX world.

1.2 License

Copyright © 2011–2012 Geoffrey Jones. Permission is granted to copy, distribute and/or modify this software under the terms of the LaTeX Project Public License, version 1.3 or later.⁵ This package is author-maintained.

1.3 About this Manual

This manual was compiled using Philipp Lehman’s `ltxdockit` document class. Document fonts were specified with the following command:

```
\RequirePackage[typeface=adobeminionpro,
                 sanstypeface=adobemyriadpro,
                 monotypeface=latinmodern,
                 textfigures=oldstylefigures
                 ]{typeface}
```

As Adobe Myriad Pro, at least, is not a readily available Postscript Type1 font, you might need to make substitutions before regenerating this document.

As an early aside, note as you read this manual that the `typeface` package was not instructed about font scaling. Rather, during loading, the `typeface` package automatically scaled the sans serif and teletype font ex-heights to match the roman font ex-height size.

1.4 Package Contents

The `typeface` package contains the following files:

- `README`—the package README file
- `typeface.pdf`—the package manual (this document)
- `typeface.tex`—the package manual source
- `typeface.sty`—the `typeface` package
- `typeface.cfg`—installation customisation and reconfiguration file
- `typeface-test.tex`—`typeface` package test harness
- `typeface-all-rm.pdf`—a *very* large typeface exhibits file (16MB or thereabouts)
- `typeface-all-rm.bat`—the MS-DOS script used to produce the file above

³ <http://www.tug.org/fontname/fontname.pdf>

⁴ \TeX ’s native `\oldstylenums` command provides some support for this. Unfortunately, it is surrounded by several noteworthy limitations, including patchy coverage across the breadth of Type1 fonts and missing reciprocal function (`\liningstylenums?`).

⁵ <http://www.ctan.org/tex-archive/macros/latex/base/lppl.txt>

1.5 Installation

Installation involves placing `typeface.sty` somewhere in your `texmf` tree then rehashing \TeX 's name database using `texhash` or whatever. These actions are performed automatically if you install this package using MiK \TeX 's or \TeX Live's system package managers.

Likewise, configuration file `typeface.cfg` should be placed in a suitable location in your system or local `texmf` trees, re-`texhash`-ing when you are done. Alternatively, you could save a customised version of `typeface.cfg` in your document directory should your document have highly particular `typeface` configuration needs.

Caveat: Please take extra precautions to ensure any site-local versions of `typeface.cfg` are not overwritten if using MiK \TeX 's or \TeX Live's system package managers. As usual, versioning or otherwise backing up your configuration files might prove a useful idea.

Finally, note that `typeface` cannot access \LaTeX Type1 fonts unless they are properly installed. Please ensure that this is done before attempting to access them through this package. MiK \TeX 's and \TeX Live's system package managers make this chore a breeze.

1.6 Typeface Exhibits File

The `typeface` package includes a large `typeface` exhibits file presenting all the roman fonts and many of the other font sets that it manages. This file, `typeface-all-rm.pdf`, was produced by compiling `typeface-test.tex` with parameters set out in `typeface-all-rm.bat`⁶. The first few pages in `typeface-all-rm.pdf` contain the output produced using default options, `debug=true`, `fontencoding=OT1`, `typeface` options set to (typographically hideous) example non-default values, and suboption `scale` set to 1, a value that switches off font scaling. The remaining pages contain exhibits of CTAN-available roman fonts accompanied by randomly selected sans serif, teletype, math and substitute text figures companion typefaces automatically scaled to match the roman font size. Exhibits of some widely available commercial fonts are included at the end of the file.

1.7 Prerequisite Conditions

This package requires `e-TeX`.

This package requires `microtype` if (and only if) text figures substitute font letter spacing is desired (advanced usage; see section 2.4, below). Note that `microtype` generally does not partner well with bitmap fonts, especially if any scaling is involved. Should you select bitmap fonts (`concrete`, `cmbright`, etc.), `microtype` will probably raise “auto expansion is only possible with scalable fonts” errors during document compilation. Bottom line: either avoid bitmap fonts (for example, install the `hfbright` package to use a Type1 variant of `cmbright`), or avoid `microtype`.

1.8 Auxiliary File

The `typeface` package produces auxiliary file `\jobname.tf` during the course of each run. The package will issue a warning whenever changes made to the auxiliary file necessitate a recompile.

2 Package Options

This package performs most of its work at package load time. Except for a small number of commands for typesetting text and lining figures and a few others for printing font information, `typeface` exposes very little functionality to users after it has been loaded. Thus, users should ordinarily focus on understanding the package's main options before turning to use its few available commands.

⁶ It should be relatively trivial to convert this script to Unix-compatible form.

2.1 Options Syntax

All package options are given in $\langle key \rangle = \langle option \rangle \langle suboptions \rangle$ syntax, where $\langle suboptions \rangle$ is constructed by appending zero or more $:\langle suboption \rangle$ terms to $\langle key \rangle = \langle option \rangle$ productions.

Spaces are ignored within these constructs, so may be freely included wherever you like, presumably for better aiding readability.

2.2 Default Options

Default behaviour occurs whenever a particular $\langle key \rangle = \langle value \rangle$ option is not explicitly supplied or, given the keyword `default` has not been modified, package option $\langle key \rangle = \text{default}$ is explicitly specified.

Please note that this package can be extensively reconfigured in order to meet the requirements of the different locales, institutions or installations where it might be used. Both the keyword `default` and the default behaviour of package keys can be modified. Thus, you should check the contents of `typeface.cfg` if you feel concerned that the package's default behaviour has been overridden in some way. See section 4, below, for further details.

2.3 Typeface Selection

`typeface` = $\langle roman font \rangle \langle suboptions \rangle$ default: `default`
`rm` = $\langle roman font \rangle \langle suboptions \rangle$

The `typeface` (syn: `rm`) option can be used to set up `\rmdefault`, that is, the document's default roman font. The value $\langle roman font \rangle$ may be one of the font names or shorthand terms (aliases) set out in table 1. This list can be extended by modifying the package configuration file `typeface.cfg` (see section 4).

A number of roman typefaces can take suboptions. These are set out in table 2. Users should refer to the relevant font documentation or simply try them out to determine what they do.

Note that `\encodingdefault`, the font encoding in place at font load time, can influence the typeface that is used. Font encodings are discussed in section 2.7, below.

Unless reconfigured in the configuration file `typeface.cfg`, package option

`typeface=default`

is essentially a no-op. In other words, `typeface` will not alter whatever value `\rmdefault` might expand to at `typeface` package load time.

`sanstypeface` = $\langle sans serif font \rangle \langle suboptions \rangle$ default: `default`
`sf` = $\langle sans serif font \rangle \langle suboptions \rangle$

`monotypeface` = $\langle teletype font \rangle \langle suboptions \rangle$ default: `default`
`tt` = $\langle teletype font \rangle \langle suboptions \rangle$

The `sanstypeface` (syn: `sf`) and `monotypeface` (syn: `tt`) options can be used to set up `\sfdefault` and `\ttdefault`, that is, the document's default sans serif and teletypewriter fonts. As with setting `typeface=default`, setting `sanstypeface=default` and/or `monotypeface=default` does very little at all—`\sfdefault` and `\ttdefault` are not altered in these cases.

Key values $\langle sans serif font \rangle$ and $\langle teletype font \rangle$ can be selected from the font names or shorthand terms (aliases) set out in tables 3 and 5. These lists can be expanded by modifying package configuration file `typeface.cfg`.

In addition to fine-tuning font selections by applying the $\langle suboptions \rangle$ listed in tables 4 and 6, sans serif and teletype fonts can be scaled using the `scale` suboption. This suboption is discussed in section 2.5, below.

Typeface Name	Option	Aliases
ADF Baskervald	adfbaskervald	baskervald
ADF Berenis	adfberenis	berenis
ADF Electrum	adfelectrum	electrum
ADF Romande	adfromande	romande
ADF Venturis	adfventuris	venturis
Adobe Minion Pro	adobeminionpro	minionpro
Adobe Utopia	adobeutopia	utopia
AE	ae	
Antykwa Półtawskiego	antykwapoltawski	antpol, poltawski
Antykwa Toruńska	antykwaterunska	anttor, torunska
Artificial Uncial	uncial	
Augie	augie	
Auriocus Kalligraphicus	auriocuskalligraphicus	auriocus, kalligraphicus
Bera Serif	beraserif	bera
Bitstream Charter	bitstreamcharter	charter
CM Bright	cmbright	
CM Dunhill	cmdunhill	
CM Fibonacci	cmfibonacci	
Computer Modern	computermodern	cm, cmr
Concrete	concrete	
Day Roman	dayroman	dayrom
DejaVu	dejavu	
Droid	droid	
European Modern	europeanmodern	em
GFS Artemisia	gfsartemisia	artemisia
GFS Bodoni	gfsbodoni	bodoni
GFS Didot	gfsdidot	didot
GFS Neohellenic	gfsneohellenic	neohellenic
Gyre Bonum	gyrebonum	bonum, tgbonum
Gyre Pagella	gyrepagella	pagella, tgpagella
Gyre Schola	gyreschola	schola, tgschola
Gyre Termes	gyretermes	termes, tgtermes
Iwona	iwona	
JAM Times	jamtimes	jam
KP Fonts	kppfonts	kp, kepler
Kerkis	kerkis	
Kurier	kurier	
Latin Modern	latinmodern	lm, lmodern
Libertine	libertine	
Lucida Bright	lucidabright	lucidabr, lucida
New Century Schoolbook	newcenturyschoolbook	nc, newcentury
PT Serif	ptserif	pt, paratype
PX Fonts	pxfonts	px
Palatino	palatino	
TX Fonts	txfonts	tx
Times Roman	times	
URW Antiqua	urwantiqua	antiqua
URW Bookman	urwbookman	bookman
URW Garamond	urwgaramond	garamond
URW Nimbus	urwnimbus	nimbus
Zapf Chancery	zapfchancery	chancery, zapf

Table 1: Roman typeface options. Usage: `typeface=⟨option⟩⟨suboptions⟩`, where `⟨option⟩` is an item from the `Option` or `Aliases` columns, and `⟨suboptions⟩` is formed by concatenating zero or more `:⟨suboption⟩` terms, for which, see table 2.

Typeface Name	Suboptions	Reference Packages
ADF Baskervald	lig	baskervald
ADF Berenis	tab, lig	berenis
ADF Electrum	lig	electrum
ADF Romande	alt	romande
ADF Venturis	2, old	venturis, venturis2, venturisold
AE	slides	ae
Adobe Minion Pro	smallfamily, medfamily, fullfamily, noopticals, opticals, slides, normalsize, nonnormalsize	MinionPro
Antykwa Półtawskiego	light	
Antykwa Toruńska	mathnoalias, light, condensed	anttor
Auriocus Kalligraphicus	backslant	aurical
CM Bright	slantedGreek, standard-baselineskips	cmbright
Concrete	exscale, amsfonts, amssymb, sansbold, boldsans	concmath
Day Roman	s	dayrom
DejaVu	condensed	DejaVuSerif, DejaVuSerifCondensed
GFS Artemisia	euler	gfsartemisia
GFS Neohellenic	symbols	gfsneohellenic
Iwona	light, condensed	iwona
KP Fonts	oldstyle, veryoldstyle, rmx, light, largesmallcaps, nofligatures, easyscsl, oldstylenumsmath (syn: osf, oldstylenums), noamsmath, sfmath, sfmathbb, rmmathbb, nomathscript, mathcalasscript, classicReIm, uprightRoman, frenchstyle, upright, oldstylenumsmath, oldstylemath, veryoldstylemath, narrowiints, partialup, widermath, noDcommand, intllimits, nointlimits, fullintllimits, sumlimits, nosumlimits, fullsumlimits, uprightgreeks, slantedGreeks	kpfonts
Kurier	light, condensed	kurier
Latin Modern	boldsc ^a	
Lucida Bright	seriftt, expert, noexpert, lucidascale, nolucidascale, lucidasmallscale, mathitalic1, mathitalic2, mathitalic3, slantedgreek, uprightgreek, vargreek, noamssymbols, amsmath, OT1, T1, LY1, seriftt, fax, casual, calligraphic, handwriting, albullet, errorshow, warningshow, nofontinfo	lucidabr, lucbmath
TX Fonts	new ^b	txfonts, newtxtext

Table 2: Roman typeface suboptions. See reference packages for the meaning of these terms.

^aDraws bold smallcaps from Computer Modern. Font encoding T1 only.

^bLoads TX Fonts from the newtx package.

Typeface Name	Option	Aliases
ADF Libris	adflibris	libris
ADF Venturis Sans	adfventuris	venturis
AE Sans	ae	
Arev Sans	arevsans	arev
Augie	augie	
Avant Garde	avantgarde	
Biolinum	biolinum	libertine
Bitstream Vera Sans	berasans	bera
CM Bright Sans	cmbright	
Cantarell	cantarell	
Comfortaa	comfortaa	
Computer Modern Sans	computermodern	cm, cms
Cyklop	cyklop	
DejaVu Sans	dejavu	
Droid Sans	droid	
European Modern Sans	europeanmodern	em
GFS Neohellenic	gfsneohellenic	neohellenic
Gyre Adventor	gyreadventor	adventor, tgadventor
Gyre Heros	gyreheros	heros, tgheros
Helvetica	helvetica	helv
Iwona	iwona	
KP Fonts Sans	kpfonts	kp, kepler
Kerkis Sans	kerkis	
Kurier Sans	kurier	
Latin Modern Sans	latinmodern	lm, lmodern
Lato	lato	
Lucida Bright Sans	lucidabright	lucidabr, lucida
Open Sans	opensans	
PT Sans	ptsans	pt, paratype
PX Fonts Sans	px fonts	px
TX Fonts Sans	tx fonts	tx
URW Arial	urwarial	arial
URW Classico	urwclassico	classico, optima
URW Grotesq	urwgrotesq	grotesq
URW Nimbus Sans	urwnimbus	nimbus

Table 3: Sans serif typeface options. See table 4 for associated suboptions.

Typeface Name	Suboptions	Reference Packages
ADF Venturis Sans	2	venturis
AE Sans	slides	ae
Biolinum	osf	libertine
Computer Modern Sans	quotation	
DejaVu Sans	condensed	dejavu
GFS Neohellenic	symbols	gfsneohellenic
Iwona	light, condensed	iwona
KP Fonts Sans	osf	kpfonts
Kurier Sans	light, condensed	kurier
Latin Modern Sans	extended	
Open Sans	osf	opensans

Table 4: Sans serif typeface suboptions. See reference packages for the meaning of each term.

Typeface Name	Option	Aliases
AE Mono	ae	
Bitstream Vera Mono	beramono	bera
Computer Modern Mono Light	cmbright	
Courier	courier	
DejaVu Mono	dejavu	
Droid Mono	droid	
European Modern Mono	europeanmodern	em
Gyre Cursor	gyrecursor	cursor, tgcursor
Inconsolata	inconsolata	
KP Fonts Mono	kpfonts	kp, kepler
Latin Modern Mono	latinmodern	lm, lmodern
Lucida Bright Mono	lucidabright	lucidabr, lucida
Luxi Mono	luximono	luxi
PT Mono	ptmono	pt, paratype
PX Fonts Mono	pxfonts	px
TX Fonts Mono	txfonts	tx
	computermodern	cm, cmt

Table 5: Teletype typeface options. See table 6 for associated suboptions.

Typeface Name	Suboptions	Reference Packages
AE Mono	slides	ae
Computer Modern Mono Light	light, proportional, L	
KP Fonts Mono	osf	kpfonts
Latin Modern Mono	proportional	lmodern
Lucida Bright Mono	serif, seriff, sans	lucidabr

Table 6: Teletype typeface suboptions. See reference packages for their meaning.

Typeface Name	Option	Aliases
Adobe Minion Pro	adobeminionpro	
AMS Math	amsmath	ams
Antykwa Toruńska	antykwaterunska	anttor, torunska
Arev Math	arevmath	arev
CM Bright	cmbright	
Concrete Math	concrete	
Euler	euler	
European Modern	europeanmodern	em
Fourier	fourier	
GFS Artemisia	gfsartemisia	artemisia
GFS Bodoni	gfsbodoni	bodoni
GFS Didot	gfsdidot	didot
GFS Neohellenic	gfsneohellenic	neohellenic
Iwona	iwona	
KP Fonts	kpfonts	kp, kepler
Kerkis	kerkis	
Kurier	kurier	
LX Fonts	lxfonts	lx
Latin Modern	latinmodern	lm, lmodern
Lucida Bright	lucidabright	lucidabr, lucida
Math Design	mathdesign	
MathTime 2	mathtimepro	mtpro
Math PTMX	mathptmx	ptmx
PX Fonts	pxfonts	px
Pazo	mathpazo	pazo
TX Fonts	txfonts	tx

Table 7: Math typeface options. See table 8 for associated suboptions.

Typeface Name	Suboptions	Reference Packages
Adobe Minion Pro	smallfamily, medfamily, fullfamily, noopticals, opticals, slides, normalsize, nonormalsize, liningstylefigures (syn: lf, lsf, lining, liningstyle, mathlf), oldstylefigures (syn: osf, oldstyle, oldfigures, mathosf), mathtabular, mnsy, cmsy, swash, abx, amsbb, fourierbb, lucidabb, mixedgreek, italicgreek, frenchmath, minionint, openg, loosequotes, footnotefigures	MinionPro
AMS Math	tbtags, nosumlimits, intllimits, nonamelimits, donotfixamsmathbugs	amsmath
Antykwa Toruńska	light, condensed	anttor
Arev Math	origletters, vara, vari, varI, varf, oldf, varl, varu, varv, varw, varGamma, varXi, varPi, varSigma, varPhi	arevmath
CM Bright	slantedGreek, standard-baselineskips	cmbright
Concrete Math	exscale, amsfonts, amssymb, sansbold, boldsans	concrete
Euler	small, euler-digits, euler-hat-accent, T1, OT1, LY1, icomma	eulervm
European Modern	T1, LY1, LM1	em
Fourier	nc, newcentury, newcenturyschoolbook, utopia, adobeutopia, upright, widespace, expert, oldstyle, fulloldstyle	fourier
GFS Artemisia	euler	gfsartemisia
Iwona	light, condensed	iwona
KP Fonts	light, noamsmath, sfmath, sfmathbb, rmmathbb, nomathscript, mathcalasscript, classicReIm, uprightRoman, frenchstyle, upright, oldstylefigures (syn: osf, oldstyle, oldfigures, oldstylenums, oldstylenumsmath), oldstylemath, veryoldstylemath, narrowiints, partialup, widermath, noDcommand, intllimits, nointlimits, fullintllimits, sumlimits, nosumlimits, fullsumlimits, uprightgreeks, slantedGreeks	kpfonts
Kerkis	light, condensed	kerkis
Lucida Bright	expert, noexpert, lucidascale, nolucidascale, lucidasmallscale, mathitalic1, mathitalic2, mathitalic3, slantedgreek, uprightgreek, vargreek, amsmath, noamssymbols, OT1, T1, LY1, seriftt, fax, casual, calligraphic, handwriting, altbullet, errorshow, warningshow, nofontinfo	lucidabr, lucbmath
Math Design	expert, uppercase:upright, uppercase:italicized, greekuppercase:upright, greekuppercase:italicized, greeklowercase:upright, greeklowercase:italicized, greekuppercase, greeklowercase, urwgaramond, garamond, adobeutopia, utopia, bitstreamcharter, charter	mathdesign
MathTime 2	amssymbols, noamssymbols, slantedGreek, uprightGreek, slantedoperators, uprightoperators, subscriptcorrection, nosubscriptcorrection, zswash, nozswash, curlybraces, straightbraces, morphedbraces, cmcal, lucidacal, eucal, mtpluscal, mtpcal, mtpccal, lucidascr, mtplusscr, mtpscr, eufrak, mtpfrak, compatiblegreek, amsbb, mtpbb, mtpbbd, mtphrb, mtphrd, mtpbbi, mtpbhi, errorshow, warningshow, nofontinfo, lite	mtpro2
Pazo	slantedGreek, noBBpl	mathpazo
TX Fonts	new ^a , varg, libertine, cmintegrals, uprightGreek, cmbraces, varbb, nosymbolsc, amssymbols, noamssymbols, ptmxitalics	txfonts, newtxmath

Table 8: Math typeface suboptions. See reference packages for their meaning.

^aLoads TX Fonts from the newtx package.

Typeface Name	Option	Aliases
AMS Symbols	amssymb	ams
FD Symbol	fdsymbol	fd
Math abx	mathabx	abx
MdSymbol	mdsymbol	md
MnSymbol	mnsymbol	mn

Table 9: Symbols typeface options. See table 10 for associated suboptions.

Typeface Name	Suboptions	Reference Packages
FD Symbol	normalweightRegular, normalweightAuto, boldweightMedium, boldweightAuto, largedelims	fdsymbol
MdSymbol	autolight, autoregular, autosemibold, Bold, bold, largedelims, Light, light, onllysansmath, regular, Regular, retainmissing, semibold, Semibold	mdsymbol

Table 10: Symbols typeface suboptions. See reference packages for their meaning.

`mathtypeface`= $\langle math font \rangle \langle suboptions \rangle$ default: default
`math`= $\langle math font \rangle \langle suboptions \rangle$

`symbolstypeface`= $\langle symbols font \rangle \langle suboptions \rangle$ default: default
`sym`= $\langle symbols font \rangle \langle suboptions \rangle$

The `mathtypeface` (syn: `math`) and `symbolstypeface` (syn: `sym`) options can be used to set up math and symbols fonts. See tables 7 and 9 for values of $\langle math font \rangle$ and $\langle symbols font \rangle$, respectively. Like other $\langle typeface \rangle$ options, these lists can be extended by modifying package configuration file `typeface.cfg`.

Apart from registering math and symbols typeface print names, `mathtypeface=default` and `symbolstypeface=default` do nothing noticeable at all.

Math and symbols $\langle suboptions \rangle$ are listed in tables 8 and 10.

In addition to the functions invoked by these suboptions, math and symbols typefaces can be scaled using the `scale` suboption with the `mathtypeface` key. As math and symbols fonts are scaled in unison, the `scale` suboption is ignored, and therefore should not be used, with the `symbolstypeface` key. The `scale` suboption only applies to `mathtypeface` options. See section 2.5, below.

2.4 Text Figures

`textfigures`= $\langle option \rangle \langle suboptions \rangle$ default: default

This option is used to specify the text figures font configuration. Text figures may be sourced from the typeface established using the `typeface=\langle roman font \rangle` package option, they may be sourced from a substitute text figures font, or they may be switched off entirely.

Once configured, the text figures font is selected using the `\textstylenums` command or `textnums` environment. The `\liningstylenums` command or `liningnums` environment selects the lining style figures font. See section 3.1 for usage.

Text figures $\langle option \rangle$ s include:

`default` This option does nothing at all.

`false` (syn: `lf`, `lsf`, `lining`, `liningstyle`, `liningfigures`, `liningstylefigures`)

This option sets `\rmdefault` to the lining figures variant of the font requested using the `typeface=\langle roman font \rangle` package option. Thus, given `typeface=adobeminionpro` and `textfigures=false`, the three lines of code:

```
12345
\textstylenums{12345}
\begin{textnums}12345\end{textnums}
```

each produce 12345, that is, 12345 rendered in Adobe Minion Pro's lining style (uppercase) figures typeface variant.

Note again: the `\textstylenums` command and the `textnums` environment generate *lining style figures* if `textfigures=false`.

On the other hand, no matter what the value of the `textfigures` option, the \LaTeX command:

```
\oldstylenums{12345}
```

will use the font's text figures (lowercase) variant, if it exists, to produce 12345.

Note that in a few rare cases, the selected roman font cannot render true lining figures. Examples include Day Roman and Auriocus Kalligraphicus typefaces. An error will be issued if `textfigures=false` is specified under these circumstances.

`true` (syn: `osf`, `oldstyle`, `oldfigures`, `oldstylefigures`, `textfigures`)

This option sets `\rmdefault` to the text figures variant of the `typeface=<roman font>` font. Given `textfigures=true`, arbitrary `typeface=<roman font>`, and input string:

```
12345
```

the output will be typeset using the text style figures font variant (12345) if and only if *<roman font>* is natively capable of supplying these glyphs. Fortunately, several roman fonts are capable of this. Unfortunately, many more are not.

The following lines of code:

```
\textstylenums{12345}
\begin{textnums}12345\end{textnums}
```

are semantically identical. If *<roman font>* is natively capable of rendering text style figures, output is the same as that immediately above (12345). If not, that is, if input 12345 produces 12345, then `\textstylenums{12345}` and `\begin{textnums}12345\end{textnums}` will attempt to generate text style figures by sourcing glyphs from the font's TS1-encoded or `smallcaps` character sets. Around one half of the roman fonts managed by `typeface` succeed with this approach, leaving around a final one third unable to produce text figures from their own font sets in any shape or form. An error will be issued if the `textfigures=true` option is specified in these instances.

No matter what these outcomes, the command:

```
\liningstylenums{12345}
```

will, as described above, attempt to typeset its argument using the font's lining figures variant.

<substitute font> This option provides a last ditch technique for typesetting text figures in cases when the `typeface=<roman font>` document font provides no other means for doing so. Substitute text figures should be used carefully, that is, with keen attention for blending in with the established `\rmdefault` font. Perfect seamlessness may be difficult to achieve. In any event, the mere attempt will certainly provoke derision by typographical purists. Life can be hard.

<substitute font> may be one of the following:

```
adobeminionpro also accepting suboption tabular (syn: tab) for tabular, that is, monospaced figures
dayroman (syn: dayrom)
kpfonts (syn: kp, kepler), also accepting suboption light for a lighter weight font
latinmodern (syn: lm, lmodern)
libertine
lucidabright (syn: lucidabr, lucida)
palatino
times
```

In these cases, arguments of the `\textstylenums` command or `textnums` environment:

```
\textstylenums{12345}
\begin{textnums}12345\end{textnums}
```

will be typeset in the nominated substitute font. All other text will be rendered in the non-substitute typeface. This includes figures passed to the `\liningstylenums` command or the `liningnums` environment, and all output generated while `\sffamily` or `\ttfamily` selections are in force.

Caveat: Advanced usage. The inter-character spacing, inter-unit kerning and the overall scaling of substitute text figures will generally require fine-tuning to promote seamless integration with the surrounding roman text. The following suboptions provide the means for making some very fine-grained adjustments:

`scale`: *<scale factor>* adjust the size of the substitute text figures font
`spacing`: *<adjustment>* adjust the substitute text figures font inter-character spacing
`lkern`: *<adjustment>* kern the space preceding the substitute text figures block
`rkern`: *<adjustment>* kern the space after the substitute text figures block

Substitute text figures font scaling is described in section 2.5, below.

The `spacing`, `lkern` and `rkern` suboptions rely on the `microtype` package for the fine-grained adjustments they provide. Consequently, they cannot be applied unless `microtype` is loaded before loading typeface. The *<adjustment>* parameter of these suboptions accepts integers between -1000 and 1000 , inclusive.

2.5 Font Scaling

<typeface>=*<typeface name>**<suboptions>*[:*scale**<scale factor>*]*<suboptions>* default: `ex`

<typeface> Package option `sanstypeface`, `monotypeface`, `mathtypeface`, `textfigures` or any of their synonyms.

<typeface name> Any typeface name accepted by *<typeface>*.

<suboptions> Zero or more :*<suboption>* terms accepted by the *<typeface>*=*<typeface name>* tuple.

<scale factor> Optional: preceded by a colon separator, `ex`, uppercase, lowercase, `figures` or any positive real number.

Scaling defaults to `ex` if the `scale`*<scale factor>* suboption is not included among *<typeface>* suboptions or if *<scale factor>* is empty.

The `scale` suboption scales *<typeface name>* by an absolute or reference font relative amount. In all cases, the reference font is the font loaded by the `typeface`=*<roman font>* package option.

Reference font relative scaling parameters include:

`ex` the ex-height of *<typeface name>* is auto-scaled to the ex-height of the reference font
`uppercase` (syn: `uc`) the height of full-ascender uppercase characters in *<typeface name>* is auto-scaled to the height of full-ascender uppercase characters in the reference font
`lowercase` (syn: `lc`) the height of full-ascender lowercase characters in *<typeface name>* is auto-scaled to the height of full-ascender lowercase characters in the reference font
`figures` (syn: `fig`) the height of numerals in *<typeface name>* is auto-scaled to the height of numerals in the reference font

Scaling parameters `uppercase`, `lowercase`, `figures` and synonyms do not apply, and will be rejected, whenever *<typeface>* is `textfigures`.

Font scaling, whether by absolute or reference font relative amounts, is not guaranteed. Much depends on the quality of the internal font metrics in the installed Type1 fonts⁷, or on the load time information in their font descriptor `.fd` files. The supplied test harness and font exhibits file might prove useful for determining individual font scaling capabilities (see section 1.4, above).

⁷ Extant varieties of `biolinum sans` Type1 provide particularly egregious examples of poor internal font metrics. Unless recently fixed, users should stick to scaling `biolinum sans` by absolute and not reference font relative amounts.

2.6 Font Load Order

`fontloadorder=<load order list>` default: default

Caveat: Advanced usage. You probably do not want to use this option in any serious way. It can cause no end of trouble.

Typeface’s default font load order is `textfigures:symbols:math:rm:sf:tt`. That is, the substitute text figure font, if specified, is loaded, then the symbols, math, roman, sans serif, and teletype fonts in succession.

Internally, much of typeface’s work involves orchestrating font package loading to achieve the user’s intended results. In addition to the primary fonts they load, many font packages also load a range of complementary fonts, ostensibly to provide greater value for their users. This situation introduces considerable potential for downstream font loading conflicts when, rather than wanting to use the package’s complementary typeface choices, one would prefer to match other typefaces with the package’s primary font instead. Font package loading nonorthogonalities can be particularly troublesome between symbols and math or between math and roman choices. The typeface package does its best to insulate users from these problems. On extremely rare occasions, however, better results might be obtained by adjusting typeface’s inbuilt font load order rule to suit.

All six terms must be included in the colon-separated *<load order list>*. However, be mindful that `fontloadorder` is “more what you’d call ‘guidelines’ than actual rules”—the font load order cannot always be rearranged, no matter what your instruction.

Example: symbols will *probably* be loaded after math fonts rather than before them by specifying `fontloadorder=textfigures:math:symbols:rm:sf:tt`.

Take this option with a grain of salt—it is there should you need it. However, you should best avoid it if you can.

2.7 Ancillary Packages

<code>fontencoding=default</code>	<code> dontload</code>	<code> ignore</code>	<code> <fontencodings list></code>	default: T1	
<code>inputencoding=default</code>	<code> dontload</code>	<code> ignore</code>	<code> <inputencoding></code>	default: utf8	
<code>textcomp=default</code>	<code> dontload</code>	<code> ignore</code>	<code> <textcomp></code>	default: full	
<code>cmap=default</code>	<code> dontload</code>	<code> ignore</code>	<code> resetfonts</code>	<code> noresetfonts</code>	default: resetfonts

The typeface package loads the `fontenc`, `inputenc`, `textcomp` and `cmap`⁸ packages by default. Descriptions of these can be found in many places, for example, in the UK T_EX FAQ.⁹ Typeface will not load these packages if `dontload` or `ignore` are supplied. Default values and the sets of valid parameters can be reconfigured by modifying `typeface.cfg` as described in section 4, below.

The `fontencoding` option accepts a colon-separated list of font encodings. In processing these, the `fontenc` package records the last item in the list in `\encodingdefault`.

2.8 Package Testing

`debug=true` | `false` default: false

If `debug=true`, the typeface package writes extra load-time and run-time processing information in the job log file. It also colour-codes output text according to selected font family. The short form `debug` is equivalent to `debug=true`. Debug colours can be reconfigured by modifying `typeface.cfg`.

⁸ The `cmap` option might be withdrawn in an upcoming “release” version. Alternatively, it might be replaced by an option for loading `glyphtounicode.tex`. Bearing in mind that typeface’s principal constituency is novice L^AT_EX users, ideas about the best way to proceed would be warmly welcomed by the author.

⁹ <http://www.tex.ac.uk/faq>

`printinfo=true|false`

default: `false`

Package test harness `typeface-test.tex` provides a useful testbed for exploring and testing various font scenarios. The `typeface` package exposes several commands for supporting this if `printinfo=true`. The short form `printinfo` is equivalent to `printinfo=true`.

The following commands become available when `printinfo=true`:

- `\tfprintpackageoptions` Prints current package option values.
- `\tfprintinfo` Prints roman, substitute text figures (if specified), sans serif, teletype, math and symbols typeface information, including \TeX font name, em size, ex height, and scale factor metrics.
- `\tfprinttext{#1}` Prints a small exhibit of alphanumeric text in different shapes and weights. Text is drawn from #1 family where #1 may be `rm`, `sf` or `tt`.
- `\tfprintfigures{#1}` Prints a small exhibit of numeric text in different shapes and weights, including output from `math`, `liningnums` and `textnums` environments and from `\textstylenums` and `\oldstylenums` commands. Figures are drawn from #1 family, given #1 contains `rm`, `sf` or `tt`.
- `\tfprintmathsample{#1}` Prints a small block of math-oriented prose, including calligraphic, blackboard and greek fonts where defined. Text is drawn from #1 family, with #1 being `rm`, `sf` or `tt`.
- `\tfprintmathfonts` Prints a 16 element vector of math family \TeX font names.
- `\tfprinttextsample{#1}` Prints a small block of prose. Text is drawn from #1 family where #1 is `rm`, `sf` or `tt`.
- `\tfprinttextalphabets` Prints `math`, `rm`, `sf` and `tt` intermixed lines of lowercase, uppercase and numeric text. Useful for comparing relative print sizes across different font families.
- `\tfprintinfopage` Executes several of the commands described above.

See `typeface-all-rm.pdf`, the font exhibits file included with this package for example usage of these commands.

3 Author Commands

3.1 Text and Lining Figures

The `typeface` package provides very few author-level commands. Those that it provides include commands and environments for switching between text and lining figures:

- `\textstylenums` This command is a homologue of \TeX 's native `\oldstylenums`. Unlike `\oldstylenums`, `\textstylenums` renders text style figures if:
 - a) package option `textfigures=true` (or a synonym for such) and the roman font established by setting `typeface=<roman font>` provides text figures natively, via TS1-encoding or through smallcaps selection; or
 - b) package option `textfigures=<substitute font>`.However, `\textstylenums` generates *lining style figures* if `textfigures=false` (or a synonym for such). The result is indeterminate if `textfigures=default`.
- `textnums` The `textnums` environment is semantically equivalent to the `\textstylenums` command.
- `\liningstylenums` This command provides a reciprocal of sorts to `\textstylenums`. In all but a few exceptional cases, its argument will be rendered in lining style figures font.
- `liningnums` The `liningnums` environment is semantically equivalent to the `\liningstylenums` command.

While their use is not particularly recommended, the commands `\tsn` and `\lsl` can be used in place of `\textstylenums` and `\liningstylenums`.

See section 2.4 for information about configuring text figure fonts.

3.2 Typeface Print Names

Typeface print names can be accessed with the following commands:

```
\rmprintname
\sffprintname
\ttprintname
\mathprintname
\symbolsprintname
\textfiguresprintname
```

For example, using `\rmprintname` in this manual produces Adobe Minion Pro. All bets are off if the user or some package modified `\rmdefault`, `\sfdefault`, etcetera after typeface was loaded.

4 Package Configuration

The typeface package was developed with locale and institution specific customisations in mind. Customisation is carried out by modifying the `typeface.cfg` file. A sample `typeface.cfg` file is bundled with this package.

Note: `typeface.cfg` is loaded early during typeface package processing. At that time, it has complete freedom to access or introduce new typeface package commands. Due care should be applied. Many of typeface's internal commands contain the `@` character in their names. There is no need to, and therefore you should not, wrap `@`-containing identifiers between `\makeatletter` and `\makeatother` parse-time modifiers.

4.1 Typeface Options

Each of the six typeface option defaults can be redefined. Here are some (typographically dubious) examples:

```
\renewcommand*\TF@DEFAULTRMFONT{kpfonts:veryoldstyle:largesmallcaps}
\renewcommand*\TF@DEFAULTSFFONT{helv:scale:0.75}
\renewcommand*\TF@DEFAULTTTFFONT{luxi:scale:lowercase}
\renewcommand*\TF@DEFAULTMATHFONT{iwona:light:condensed}
\renewcommand*\TF@DEFAULTSYMBOLSFONT{ams}
\renewcommand*\TF@DEFAULTTEXTFIGURESFONT{palatino:spacing:20:lkern:-20}
```

Observe that these definitions literally substitute for default user package options. Accordingly, note that list separators are colons and not commas.

Extra code can be executed immediately before and/or after each typeface family is loaded. To do so, define `\beforeloading<fam>` and/or `\afterloading<fam>` in `typeface.cfg`. Here, `<fam>` is any of `textfigures`, `symbols`, `math`, `rm`, `sf` or `tt`.

For example:

```
\newcommand*\beforeloadingmath{
  \ifthenelse{\equal{\tf@mathfont}{amsfonts}}{
    % undefine the following to prevent amsfonts namespace clashes
    \tf@undefinecommands{\underrightarrow,\underleftarrow}
    \TF@PackageDebugInfoNoLine[\TF@CONFIGFILE]{%
      Blew away \string\underrightarrow\space and
      \string\underleftarrow\MessageBreak
      symbols before loading amsfonts package}
  }{}
}
```

4.2 Ancillary Package Options

Ancillary package option defaults can be configured in `typeface.cfg`. For example:

```
\renewcommand*\TF@DEFAULTFONTENCODING{EU1:T1}
\renewcommand*\TF@DEFAULTINPUTENCODING{dontload}
\renewcommand*\TF@DEFAULTTEXTCOMP{safe}
\renewcommand*\TF@DEFAULTCMAP{ignore}
```

Extra choices can be added to `typeface`'s `fontencoding` and `inputencoding` package options lists. For example, suppose you wanted to make (fictitious) K9 and K10 font encodings available to `typeface` package users. Then, after ensuring `k9enc.def` and `k10enc.def` encoding files are installed, simply include:

```
\renewcommand*\tf@fontencodingchoices{K9,K10}
```

in `typeface.cfg`. Similarly, use `\tf@inputencodingchoices` to add more options to `typeface`'s inbuilt `inputencoding` choice list.

4.3 `\newtypeface` Command

The `\newtypeface` command can be used to place additional Type1 fonts under `typeface` package access management. This command takes one optional and four mandatory arguments:

- #1 optional comma-separated list of `typeface` name aliases (enclose in square brackets)
- #2 primary `typeface` name
- #3 human-readable print name, accessible through commands outlined in section 3.2
- #4 `typeface` family: `rm`, `sf`, `tt`, `math`, `symbols` or `textfigures`
- #5 `typeface` initialisation code

Example 1.

```
% Adobe Bembo
\newtypeface[bembo]{adobebembo}{Adobe Bembo}{rm}{
  \renewcommand*\rmdefault{pbb}
  \tf@SCencodedosf
}
```

Assuming Adobe Bembo is installed on the system and accessible by its `pbb` Berry font name, `typeface` users can now set their document's default roman font to Adobe Bembo using options `typeface=adobebembo` or `typeface=bembo`. Text figures will be selected from Adobe Bembo's smallcaps font.

Text figures declarations like `\tf@SCencodedosf` apply only to `\newtypeface rm` definitions. Four commands are available for text figures configuration:

- `\tf@TSencodedosf` Source text figures via TS1 font encoding
- `\tf@SCencodedosf` Source text figures by switching to smallcaps
- `\TF@NativeOSFError` Advise user that the selected font does not accept package option `textfigures=true`
- `\TF@NativeLFError` Advise user that the selected font *requires* package option `textfigures=true` (in practice, this is very rare)

Absent these declarations, text figures are assumed to be natively available in the requested roman font. Nevertheless, without further configuration, the `\textstylenums` and `\liningstylenums` commands (resp: the `textnums` and `liningnums` environments) do very little at all. Therefore, at a minimum, you should specify the font's text figures and lining figures configuration. One way of doing this is shown below:

Example 2.

```
% Adobe Sabon
\newtypeface[sabon]{adobesabon}{Adobe Sabon}{rm}{
  \ifthenelse{\equal{\tf@osf@method}{\TF@OSF@JF}}{
    \renewcommand*\rmdefault{psbj} % Native old style figures
    \tf@define@liningfigures{psb}
  }{
    \renewcommand*\rmdefault{psb} % Native lining figures
  }
}
```

By the time `\newtypeface rm` initialisation code is reached, internal macro `\tf@osf@method` will have been initialised with one of the following four values¹⁰:

```
\TF@OSF@LF the user set package option textfigures=false (or synonym)
\TF@OSF@JF the user set package option textfigures=true (or synonym)
\TF@OSF@XF the user set package option textfigures=(substitute font)
\TF@OSF@NF the user has implicitly or explicitly set package option textfigures=default
```

Thus, in this example, we set the document's default roman font to `psbj` (Adobe Sabon with old style figures) and the lining figures font to `psb` (Adobe Sabon with lining style figures) whenever the user sets `textfigures=true`, and to `psb` otherwise.

Example 3.

```
% Adobe Myriad Pro
\newtypeface{adobemyriadpro}{Adobe Myriad Pro}{sf}{
  \tf@ifsuboption{osf}{\tf@sffontoptions}{
    % Myriad Pro has an OSF variant (depending on your installation)
    \renewcommand*\sfdefault{Myriad-OSF}
    \xdef\tf@sfprintname{\tf@sfprintname\ OSF}
  }{
    \renewcommand*\sfdefault{Myriad-LF}
  }
}
```

We set up sans serif font Adobe Myriad Pro in this example. This `sf` font is quite unusual in possessing lining and old style figures variants. We allow users their choice with this by testing for `sanstypeface` suboption `osf` in the `\newtypeface` code, initialising the font and its print name appropriately.

Note that we did nothing special to enable `rm` font relative scaling in this case. The `typeface` package automatically handles this in (most) cases where the corresponding `sf` or `tt` font descriptor `.fd` file *does not* contain scaling instructions.¹¹

Example 4.

```
% Helvetica
\newtypeface[helv]{helvetica}{Helvetica}{sf}{
  \scaletypeface{sf}
  {\PassOptionsToPackage{scaled=\tf@sfscalefactor}{helvet}}
  \tf@usefontpackage{helvet}
}
```

¹⁰ `\tf@osf@method` initialisation occurs during `textfigures` option processing. This processing will not have taken place before `\newtypeface rm` initialisation code is executed if `rm` precedes `textfigures` in package option `fontloadorder` (another reason to use this option with great care—see section 2.6). In that event, `\tf@osf@method` will expand to `\TF@OSF@NF`.

¹¹ Compare `PXFonts Sans t1pxss.fd` which does not contain scaling instructions and `Helvetica t1phv.fd` which contains scaling machinery. The `\newtypeface` command automatically accommodates typeface scaling in the former case without special intervention. However, we must provide the `\newtypeface` command explicit instruction about how to initialise scaling in the latter case. This is explored in examples 4 and 5.

In this example, using `typeface`'s `\scaletypeface` command, we simply pass the font family appropriate scale factor, `\tf@sfscalefactor`, to the scaled option in the `helvet` package.

Note that `typeface` command `\tf@usefontpackage` is an instrumented version of `\usepackage`. It should be used whenever loading font packages (and only then) since it facilitates debug information logging whenever package option `debug=true`.

Example 5.

```
% Helvetica
\newtypeface[helv]{helvetica}{Helvetica}{sf}{
  \scaletypeface{sf}{\newcommand*\Hv@scale{\tf@sfscalefactor}}
  \renewcommand*\sfdefault{phv}
}
```

We take an alternative approach to sans serif font scaling configuration in this example. As outlined in example 4, internal `typeface` command `\scaletypeface` accepts any scaling code eventually recognised by the target font definition (`.fd`) file. Here, we initialise `\Hv@scale` to `typeface`'s pre-computed sans serif font scale factor and specify the font we want in `\sfdefault`. Again, there is no more to do in order to set up reference font (`rm`) relative scaling for `sans serif` and `teletype` fonts.

Example 6.

```
% Adobe Sabon text figures
\newtypeface[sabon]{adobesabon}{Adobe Sabon}{textfigures}{
  \tf@initialisetextfiguresfont{T1}
  \tf@DeclareOsfFontShape{m}{n} {psbr9d}
  \tf@DeclareOsfFontShape{m}{sc} {psbrc9d}
  \tf@DeclareOsfFontShape{m}{it} {psbri9d}
  \tf@DeclareOsfFontShape{m}{sl} {psbri9d}
  \tf@DeclareOsfFontShape{b}{n} {psbb9d}
  \tf@DeclareOsfFontShape{b}{it} {psbbi9d}
  \tf@DeclareOsfFontShape{b}{sl} {psbbi9d}
  \tf@DeclareOsfFontShape{bx}{n} {psbb9d}
  \tf@DeclareOsfFontShape{bx}{it} {psbbi9d}
  \tf@DeclareOsfFontShape{bx}{sl} {psbbi9d}
}
```

This example demonstrates how to set up a new text figures typeface. This task requires just two commands. First, we declare the text figures font encoding using `\tf@initialisetextfiguresfont`. That done, we transliterate the information from the appropriate font descriptor (`.fd`) file¹² into a series of `\tf@DeclareOsfFontShape` commands.

Further examples of `\newtypeface` usage can be found in the bundled configuration file and, of course, in package file `typeface.sty` itself.

4.4 Redefining Predefined Typefaces

`Typeface` package `typeface` definitions might not always be what you want them to be. In such cases, you can override `typeface`'s inbuilt `\newtypeface` declarations by defining and saving new versions in `typeface.cfg`. Simply use the `\newtypeface` command as before, the only requirement being that, to override an internal `\newtypeface` definition, you must use the same key identifier #2 (the primary typeface name) and #4 (the typeface family) in your new typeface declaration.

¹² In our case, turning to file `t1psbj.fd` for the required values.

5 Revision History

0.1 2012-06-18

Initial release for comments (pre-release).