

## Using OpenType and TrueType fonts with X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X and LuaL<sup>A</sup>T<sub>E</sub>X

Herbert Voß

### Abstract

For both “new” T<sub>E</sub>X engines X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X and LuaL<sup>A</sup>T<sub>E</sub>X, which are admittedly no longer all that new, there are a few things to consider in connection with fonts that are important for users who have previously worked with pdfL<sup>A</sup>T<sub>E</sub>X. Until now, only the fonts that came with the T<sub>E</sub>X distribution in use were easily available to “inexperienced” users.

### 1 Introduction

With the use of X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X or LuaL<sup>A</sup>T<sub>E</sub>X the following facts have to be considered:

- The default font is by definition Latin Modern, regardless of whether `fontspec` is loaded or not. For pdfL<sup>A</sup>T<sub>E</sub>X, the default is Computer Modern.
- The packages `inputenc` and/or `fontenc` should not be loaded. First, UTF-8 is already for years the all-but-universal default input encoding, and second, `fontspec` automatically loads `fontenc` with the TU (Unicode) encoding.
- To support OpenType fonts for math, load the `unicode-math` package instead of the `amsmath` and/or `amssymb` packages. The `amsmath` package can still be used, but should be loaded *before* `fontspec/unicode-math`.
- The `xltxtra` and `xunicode` packages are obsolete and should no longer be used.

The `fontspec` package can be used with both X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X and LuaL<sup>A</sup>T<sub>E</sub>X:

```
\usepackage[options]{fontspec}
```

It greatly simplifies the integration of OpenType and TrueType fonts that are not part of the T<sub>E</sub>X distribution. In order for the system to find them automatically, they must either be in the current document directory or, depending on the operating system, in (usually) one of the following directories:

GNU/Linux	<code>/usr/share/fonts</code>
	<code>/usr/local/share/fonts</code>
	<code>~/ .fonts</code> (user-specific)
Windows	<code>C:\Windows\Fonts</code>
macOS	<code>/System/Library/Fonts</code>
	<code>/Library/Fonts</code>
	<code>~/Library/Fonts</code> (user-specific)

Any other readable directory can be used, if the path is passed to the package `fontspec`.

First published in *Die T<sub>E</sub>Xnische Komödie* 1/2022, pp. 49–57; translation by the author.

The properties of a font can be displayed with an appropriate program, e.g. `otfinfo`, which is available on every T<sub>E</sub>X distribution. For a font which is part of the T<sub>E</sub>X directory tree, one can simplify the argument: the complete path is not needed, since it can be found with `kpsewhich`:

---

```
$ otfinfo -i 'kpsewhich ComicNeue-Regular.otf'
Family:           Comic Neue
Subfamily:        Regular
Full name:        Comic Neue Regular
PostScript name:  ComicNeue-Regular
Version:          Version 2.003;hotconv 1.0.109
Unique ID:        2.003;;Comic Neue Light
Designer:         Craig Rozynski
Designer URL:     http://www.craigrozynski.com
Manufacturer:     Craig Rozynski
Vendor URL:       http://www.comicneue.com
Copyright:        Copyright 2014 The Comic Neue
                  Project Authors (https://github.com/crozynski/
                  comicneue)
License URL:      https://scripts.sil.org/OFL
License Description: This Font Software is licensed
                  under the SIL Open Font License, Version 1.1. [...]
Vendor ID:        UKWN
```

---

### 2 Font search: `luaotfload-tool` and `luafindfont`

The program `luaotfload-tool` can be used to list the fonts installed on the system. Both system fonts and T<sub>E</sub>X distribution fonts are taken into account. However, the program is geared more to the needs of T<sub>E</sub>X itself than to those of users. For example, searching for the font “Times” typically returns something like this:

---

```
$ luaotfload-tool --find=times
luaotfload | resolve : Font "times" found!
luaotfload | resolve : Resolved file name "/System/
Library/Fonts/Times.ttc", subfont nr. 0
```

---

That is, the output of the script is only one font, although several variants are installed, albeit with different file names. The search can be expanded by using the `--fuzzy` option. Now searching for “times new roman”, for example:

---

```
$ luaotfload-tool --fuzzy --find="times new roman"
luaotfload | resolve : Font "times new roman" found!
luaotfload | resolve : Resolved file name "/System/
Library/Fonts/Supplemental/Times New Roman.ttf"
```

---

But even this search is not especially successful. With the Lua utility `luafindfont`, which is part of the T<sub>E</sub>X distribution, searching for *times* yields more informative results (see Listing 1).

The `luafindfont` program can be started with various options and also allows an AND condition when specifying the font to search for. A brief listing of the options:

---

```
-h, --help
-i, --info      font number to use (default 0)
```

---

```
-m,--max_string (default 90)
-n,--no-symbolic-names
-o,--otfinfo font number and options (default 0)
-v,--verbose verbose output
-x,--xetex
(string)
```

The main *(string)* argument is usually (part of) a font name, but can contain extra conditions, as explained below. A longer description of the options:

- i *(number)* The existing font styles are output for the font with the specified number.
- m *(number)* Number of characters to be used for the output of the font name including the path. The full path specification can be very long and thus can be limited, for example, to 50 characters by specifying `-m 50`. Characters in the middle of the path are replaced with “...”.
- n Omit the symbolic (family) names column from the output.
- o *(number[option])* The `otfinfo` program is run on the font with the specified number to supply additional font information. Options for `otfinfo` must immediately follow the font number.
- x Test if font is found by `kpsewhich` (1/0).

Some example applications of `luafindfont`:

```
# search for times in the filename with path:
luafindfont times

# search for palatino and run otfinfo -v on font no 3:
luafindfont palatino -o 3v

# search for arial, font no 3, max 50 characters:
luafindfont -i 3 -m 50 arial
```

To search for both a font and style, the specifications are linked with the ‘&’ character, which must be quoted for the shell. As an example, Listing 2 searches for all the Myriad semibold fonts.

Various options are supported by the `otfinfo` program, with its `i` option being the default. Listing 3 has an example of its output.

```
----- otfinfo options -----
a Report font's family name.
f Report font's GSUB/GPOS features.
g Report font's glyph names.
i Report font's names and designer/vendor info.
p Report font's PostScript name.
s Report font's supported scripts.
t Report font's OpenType tables.
v Report font's version information.
z Report font's optical size information.
```

The following examples are marked with a file name in the margin. All example files can be downloaded from <https://tug.org/~hvoss/tb135.zip>.

These are complete documents whereas in this article I show only the important parts of the preamble and document body, with the output in a frame below. All examples must be run with Lua $\TeX$ .

### 3 Font selection by name

Selecting a font by its symbolic name assumes the font can be found in one of the directories listed above. X $\TeX$  and Lua $\TeX$  go different ways: X $\TeX$  searches for its fonts with `fontconfig`; this library ([freedesktop.org/www/Software/fontconfig](http://freedesktop.org/www/Software/fontconfig)) is included in X $\TeX$ , while Lua $\TeX$  determines the fonts from a self-created font catalogue. However, the normal user does not have to be particularly interested in this.

```
\usepackage{fontspec}

\fontspec{Cambria}A short test with the font Cambria
and now a change to {\fontspec{DejaVu Serif}%
[Scale=0.85] the font DejaVu and now at last
{\fontspec{Arial}[Scale=0.9]to the font Arial.}}
```

tb135voss-1.tex

A short test with the font Cambria and now a change to the font DejaVu and now at last to the font Arial.

As shown above, the correct font names can be found using various programs, such as `pdf fonts`. Listing 4 shows the fonts used in the above example.

### 4 Font selection by file name

This variant should always be used for X $\TeX$  if you want to use OpenType or TrueType fonts from the existing  $\TeX$  tree, which are normally not recognized by the underlying operating system. This restriction does not exist for Lua $\TeX$ ;  $\TeX$  font trees are also searched.

```
\usepackage{fontspec}

\fontspec{Iwona}A test in the Iwona font and now a
switch to the {\fontspec{Kurier}Kurier font and now
to the {\fontspec{Antykwa Poltawskiego}Antykwa
Poltaws-kiego font}} which doesn't look like
\fontspec{HelveticaNeue}Helvetica Neue.
```

tb135voss-2.tex

A test in the Iwona font and now a switch to the Kurier font and now to the Antykwa Poltawskiego font which doesn't look like Helvetica Neue.

Since Lua $\TeX$  manages its own cache for the fonts, there is a pause in the  $\TeX$  run the first time it is called because this cache has to be created:

```
[...]
luaotfload | db : Font names database not found,
generating new one.
```

Listing 1: Searching for Times on my local system

```
$ luafindfont times
```

No.	Filename	Symbolic Name	Path
1.	Times New Roman Bold Italic.ttf	timesnewroman	/Users/voss/Library/Fonts/Times/
2.	Times New Roman Bold Italic.ttf	timesnewroman	/System/Library/Fonts/Supplemental/
3.	Times New Roman Bold.ttf	timesnewroman	/System/Library/Fonts/Supplemental/
4.	Times New Roman Bold.ttf	timesnewroman	/Users/voss/Library/Fonts/Times/
5.	Times New Roman Italic.ttf	timesnewroman	/Users/voss/Library/Fonts/Times/
6.	Times New Roman Italic.ttf	timesnewroman	/System/Library/Fonts/Supplemental/
7.	Times New Roman.ttf	timesnewroman	/System/Library/Fonts/Supplemental/
8.	Times New Roman.ttf	timesnewroman	/Users/voss/Library/Fonts/Times/
9.	Times.ttc	times	/System/Library/Fonts/
10.	Times_Sans_Serif.ttf	timesansserif	/Users/voss/Library/Fonts/Times/
11.	TimesNewRomanMTStd-Bold.otf	timesnewromanmtstd	/Users/voss/Library/Fonts/Times/
12.	TimesNewRomanMTStd-BoldCond.otf	timesnewromanmtstd	/Users/voss/Library/Fonts/Times/
13.	TimesNewRomanMTStd-BoldIt.otf	timesnewromanmtstd	/Users/voss/Library/Fonts/Times/
14.	TimesNewRomanMTStd-Cond.otf	timesnewromanmtstd	/Users/voss/Library/Fonts/Times/
15.	TimesNewRomanMTStd-CondIt.otf	timesnewromanmtstd	/Users/voss/Library/Fonts/Times/
16.	TimesNewRomanMTStd-Italic.otf	timesnewromanmtstd	/Users/voss/Library/Fonts/Times/
17.	TimesNewRomanMTStd.otf	timesnewromanmtstd	/Users/voss/Library/Fonts/Times/
18.	TimesNewRomanPS-BoldItalicMT.otf	timesnewromanpsmt	/Users/voss/Library/Fonts/Times/
19.	TimesNewRomanPS-BoldMT.otf	timesnewromanpsmt	/Users/voss/Library/Fonts/Times/
20.	TimesNewRomanPSMT.otf	timesnewromanpsmt	/Users/voss/Library/Fonts/Times/
21.	TimesNewRomanPSStd-Bold.otf	timesnewromanpsstd	/Users/voss/Library/Fonts/Times/
22.	TimesNewRomanPSStd-BoldIt.otf	timesnewromanpsstd	/Users/voss/Library/Fonts/Times/
23.	TimesNewRomanPSStd-Italic.otf	timesnewromanpsstd	/Users/voss/Library/Fonts/Times/
24.	TimesNewRomanPSStd-Regular.otf	timesnewromanpsstd	/Users/voss/Library/Fonts/Times/

```
luaotfload | db : This can take several minutes; please
be patient.(compiling luc: /home/voss/texlive/2022/
texmf-var/luatex-cache/generic/fonts/otf/lmroman10-
regular.luc)(compiling luc: /home/voss/.texlive2022
/texmf-var/luatex-cache/generic/fonts/otf/lmroman10
-regular.luc)(save: /home/voss/texlive/2022/texmf-
var/luatex-cach
[...]
```

Running the example above with  $X_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$  instead of  $\text{L}_{\text{u}}\text{a}_{\text{T}}\text{E}_{\text{X}}$  will produce an error message because the fonts Iwona, Kurier, and Antykwa Poltawskiego will not be found by  $X_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$ , since a full file name is not specified. In the following example, the file name with an extension must therefore be specified for the first three fonts, whereas *HelveticaNeue* is still loaded via the name, since it is a system font in my local macOS system and not part of the  $\text{T}_{\text{E}}\text{X}$  distribution.

tb135voss-3.tex

```
\usepackage{fontspec}
```

```
\fontspec{Iwona-Regular.otf} A test in the font Iwona
and now a change to the font
{\fontspec{Kurier-Regular.otf} Kurier and now to
{\fontspec{antpolt-regular.otf} most recently the
font Antykwa Poltawskiego,}} which doesn't look like
\fontspec{HelveticaNeue} Helvetica.
```

A test in the font Iwona and now a change to the font Kurier and now to most recently the font Antykwa Poltawskiego, which doesn't look like Helvetica.

The file extension can be specified using the optional argument `Extension` and a directory that is not in the normal search path using the `Path` option. In this case, however, only the specified font style is activated; in the following example, `\bfseries` does not display bold because no corresponding bold variant was declared or was not found by `fontspec`.

```
\usepackage{fontspec}
\setmainfont{BertholdWalbaumBook.ttf}
```

```
A test with the Berthold Walbaum font:\par
A completely normal text in the old beautiful
font, which was \bfseries embedded as TrueType.
```

tb135voss-4.tex

A test with the Berthold Walbaum font:  
A completely normal text in the old beautiful font, which was embedded as TrueType.

The bold variant can be assigned using the optional argument `BoldFont`:

**Listing 2:** Searching for a font with a special shape

```
$ luafindfont -i 4 "myriad & semibold"
No.          Filename Symbolic Name          Path
1.          MyriadPro-Semibold.otf myriadpro /Users/voss/Library/Fonts/MyriadPro/
2.          MyriadPro-SemiboldCond.otf myriadpro /Users/voss/Library/Fonts/MyriadPro/
3.          MyriadPro-SemiboldCondIt.otf myriadpro /Users/voss/Library/Fonts/MyriadPro/
4.          MyriadPro-SemiboldIt.otf myriadpro /Users/voss/Library/Fonts/MyriadPro/
5.          MyriadPro-SemiboldSemiCn.otf myriadpro /Users/voss/Library/Fonts/MyriadPro/
6.          MyriadPro-SemiboldSemiCnIt.otf myriadpro /Users/voss/Library/Fonts/MyriadPro/
7.          MyriadPro-SemiboldSemiExt.otf myriadpro /Users/voss/Library/Fonts/MyriadPro/
8.          MyriadPro-SemiboldSemiExtIt.otf myriadpro /Users/voss/Library/Fonts/MyriadPro/

Font: myriadpro
Fonttype otf(system) --> | Regular | Bold | Italic | BoldItalic |
```

**Listing 3:** Printing OpenType features

```
$ luafindfont -o 2f "myriad & semibold"
(output from Listing 2 omitted)
Running otfinfo -f for font no.2
otfinfo -f "/Users/voss/Library/Fonts/MyriadPro/
MyriadPro-SemiboldCond.otf"
aalt Access All Alternates
case Case-Sensitive Forms
csp Capital Spacing
dnom Denominators
fina Terminal Forms
frac Fractions
kern Kerning
liga Standard Ligatures
lnum Lining Figures
numr Numerators
onum Oldstyle Figures
ordn Ordinals
pnum Proportional Figures
sinf Scientific Inferiors
sups Superscript
tnum Tabular Figures
zero Slashed Zero
```

tb135voss-5.tex

```
\usepackage{fontspec}
\setmainfont{BertholdWalbaumBook}
[Path=fonts/,
Extension=.ttf,
BoldFont=BertholdWalbaumMediumBook]
```

A test with the Berthold Walbaum font:

A completely normal text in the beautiful old font, which was `\bfseries` integrated as `TrueType`.

A test with the Berthold Walbaum font:  
A completely normal text in the beautiful old font, which was **integrated as TrueType**.

Entering the fonts via a name can be simplified when using Lua $\LaTeX$  if you work with the wildcard `*`. Then the part of the name that is the same for all variants needs to be specified only for the base name.

In the following example, the fonts are only found using the base name `BertholdImagoBQ`, whereby this base name itself is not a font name. Therefore, a definition must also be made for `UprightFont`.

```
BertholdImagoBQ-Book.otf
BertholdImagoBQ-BookItalic.otf
BertholdImagoBQ-MediumItalic.otf
BertholdImagoBQ-Medium.otf
```

```
\usepackage{fontspec}
\setmainfont{BertholdImagoBQ}[
UprightFont=*-Book,
ItalicFont=*-BookItalic,
BoldItalicFont=*-MediumItalic,
BoldFont=*-Medium]
```

```
A test with the Berthold Imago font:\par
A completely normal text in the beautiful
new font, which was integrated as
\textbf{OpenType}. \textit{The font as
Italic and now additionally
\bfseries as a bold variant}
```

tb135voss-6.tex

A test with the Berthold Imago font:  
A completely normal text in the beautiful new font, which was integrated as **OpenType**. *The font as Italic and now additionally as a bold variant*

## 5 Font families

With the previous examples, the main font was defined in each case, which can also be changed later by further calls; thus `\setromanfont` is rarely used, since it generally corresponds to the main font. The old syntax with `\setmainfont[options]{font name}` is still possible for all macros for reasons of compatibility.

In general, defining the fonts and their associated features is very time-consuming if the naming of the fonts is not organized in such a way that the `fontspec` package can do the assignment itself.

**Listing 4:** Font list of the first example pdf

---

```
$ pdffonts tb135voss-1.pdf
```

name	type	encoding	emb	sub	uni	object	ID
EKHFKG+Cambria	CID Type 0C	Identity-H	yes	yes	yes	8	0
DSBAVG+DejaVuSerif	CID TrueType	Identity-H	yes	yes	yes	9	0
CELWW+ArialMT	CID TrueType	Identity-H	yes	yes	yes	10	0

---

There are many packages that relieve the user of this work. As of this writing, CTAN lists 61 packages which do all the font setting internally. Here are some of the more commonly-used ones (capitalized according to the .sty filename, as distributed): accanthis, Alegreya, bitter, cantarell, CharisSIL, Chivo, CormorantGaramond, crimson, CrimsonPro, dejavu-otf, droidsans, droidserif, ebaramond, garamondlibre, gfsneohellenicot, imfellEnglish, kpfonts-otf, lato, lexend, libertinus-otf, librebaskerville, LibreBodoni, libreacaslon, linguisticspro, marathi, newpxtext, newtxtext, noto, noto-serif, opensans, plex-otf, plex-serif, quattroceto, roboto, Rosario, sourceserifpro

The complete list, with links, is available at <https://hvoss.org/Books/fontpackages.html>. For more information about any package, you can visit <https://ctan.org/pkg/<name>>.

We already differentiate between the T<sub>E</sub>X engines used and independently load the required font formats. Also, all font packages also load fontspec by default. As a final example, we show the main part of the package file Alegreya.sty:

```

_____ Font package _____
[...]
\ifAlegreya@otf
  \RequirePackage{fontspec}
\else
  \RequirePackage{fontenc,fontaxes,mweights}
\fi

\ifAlegreya@otf
  \setmainfont[
    Numbers      = {\Alegreya@figurealign,
                   \Alegreya@figurestyle},
    UprightFont  = *-\Alegreya@regstyle ,
    ItalicFont   = *-Italic,
    BoldFont     = *-\Alegreya@boldstyle ,
    BoldItalicFont = *-\Alegreya@boldstyle Italic ,
  ]{Alegreya}
[...]

```

---

By virtue of this work in the package, the user need only write the `\usepackage` command shown below. In contrast, we use another font, Anonymous-Pro, for the typewriter text, for which there exists no font package and thus we have to use `\setmonofont` explicitly.

```
\usepackage{Alegreya,AlegreyaSans}
\setmonofont[FakeStretch=0.8,
             Scale=MatchLowercase]{AnonymousPro}
```

```
The normal font is \textsc{Alegreya}, which is
also possible as \textbf{bold}.\par
\sfamily The sans serif is \textsc{Alegreya Sans},
which is also available
in \textbf{bold}.\par
\ttfamily And the mono font is Anonymous Pro, which
is yet again available as a \textbf{bold font}.\par
\addfontfeature{FakeStretch=0.65}We
can further condense the mono font;
the \textbf{bold version} gets the same treatment.
```

The normal font is ALEGREYA, which is also possible as **bold**.

The sans serif is ALEGREYA SANS, which is also available in **bold**.

And the mono font is Anonymous Pro, which is yet again available as a **bold font**.

We can further condense the mono font; the **bold version** gets the same treatment.

◇ Herbert Voß  
Wasgenstraße 21  
14129 Berlin, Germany  
herbert (at) dante dot de  
<https://hvoss.org/>

tb135voss-7.tex