

lua-font-config

Clea F. Rees

v0.1*

Abstract

An opentype font configuration package for Lua^AT_EX 2_ε. The aim is to provide an alternative to fontspec which is both faster and more powerful, by doing as much processing as possible in Lua. To avoid loading fonts unnecessarily, the data used is corrected during compilation and the results cached for future use.

This package is experimental.

1 Requirements

The package requires Lua^AT_EX 2_ε. It will not work with any other engine or format.

Unicode maths is supported *via* lua-unicode-math. It will **not** work with unicode-math.

2 Usage

*The user interface is currently **extremely** basic.*

All that is properly supported is the choice of font family for the standard families — i.e. roman/serif, sans, typewriter/mono and (Unicode) maths (main, script and calligraphic) — and configuration of additional fonts, again by family. In addition to this, scaling and raw feature selection are supported. That is, most things are supported in the sense that they are possible, but the interface is rudimentary in the extreme.

That said, the automated configuration is quite extensive. By default, the package tries to configure font families which are as rich as possible, insofar as the fonts support this. This includes

- multiple widths and shapes, insofar as NFSS supports them;
- italic, oblique, upright italic, reverse italic, reverse oblique;
- small-caps, italic small-caps, oblique small-caps¹;

*SVN: 12127 2026/10/15

¹This should work whether small-caps are in separate fonts (e.g. Latin Modern) or implemented by the smcp feature (e.g. T_EX-Gyre Pagella).

- optical sizes.

Of course, this depends on the quality of the data the fonts provide. It is not always good or, even, vaguely accurate. If things do not fit, some (currently minimal) effort is made to accommodate them as distinct font families, so they are still available at document level.

At present, this is a bit chaotic and a bit hard to use, as the results are ‘hidden’ from you (in, of course, plain sight). Even so, you should typically need to do much less to access the fonts you want than you might otherwise expect.

The advantage of this is that if a font has several widths, you can easily adjust the one used for bold by simply redefining the relevant default. This is not very user friendly yet, but it is not completely terrible.

Simple usage looks like this:

```
\usepackage{lua-font-config}
\fontconfig{%
  rm = texgyrepagella,
  sf = texgyreheros,
  tt = texgyrecursor,
  math = texgyrescholamath,
}
```

`\fontconfig {⟨key-val list⟩}`

Specify font configuration.

2.1 Keys

Top-level keys are described here. Keys for use in key-value lists are described in section 2.2.

`rm (key) = ⟨family name⟩`

`rm (key) = ⟨key-val list⟩`

Specify family to use as the main document font, typically a serif or ‘roman’ family.

Available keys are listed below.

`sf (key) = ⟨family name⟩`

`sf (key) = ⟨key-val list⟩`

Specify family to use as the default sans family, typically a sans-serif.

Available keys are listed below.

`tt (key) = ⟨family name⟩`

`tt (key) = ⟨key-val list⟩`

Specify family to use as the default ‘typewriter’ family, typically monowidth.

Available keys are listed below.

`fam (key) = <family name>`

`fam (key) = <key-val list>`

Specify an additional text family. This will create a macro which can be used to switch to the requested family².

Available keys are listed below.

`math (key) = <family name>`

`math (key) = <key-val list>`

Specify the family to use for typesetting Unicode mathematics. This sets the fonts used for symbols, script and calligraphic alphabets. It does not change the fonts used to typeset operators, since these are taken from text fonts and configured separately.

Available keys are listed below.

This option uses lua-unicode-math. unicode-math is **not** supported. If you wish to use unicode-math, do not use this option. Instead, load unicode-math and use `\setmathfont` as usual.

`debug (key) = <key-val list>`

This can be used to enable debugging and/or modify the logging level. It is primarily intended for use in debugging the package and unlikely to be of use in production. However, it may be useful the use this key if reporting a bug which cannot be reproduced e.g. which is platform specific or depends on system fonts.

2.2 Keys for specifying font families

All keys described in section 2.1 accept either a simple family name, as shown in the example above, or a braced key-value list. For example,

```
\usepackage{lua-font-config}
\fontconfig{%
  rm = {%
    name = texgyrepagella,
    fea = {script=ltl;+tlig;+pnum}},
  },
  sf = {%
    name = texgyreheros,
    scale = 1.2,
    fea = {script=cyr1;}},
  },
  tt = {%
    name = texgyrecursor,
    fea = {+zero;}},
    %% for completeness: false is anyway the default
    force = false,
  },
}
```

²Usually you will want to specify which macro, since otherwise you must figure out the family name the code uses, which may or may not match the one requested. See section 2.2.4.

}

This section describes these ‘inner’ keys.

2.2.1 Common keys

These keys are accepted by all of `rm`, `sf`, `tt`, `fam` and `math`.

`name (key)` = `<lookup name>`

The ‘name’ to lookup. This may, but need not be, the name of a font family. Shorter, less specific names will configure more families; longer, more specific names will configure fewer.

For example, `name = texgyrepage11ax`, will configure fewer than `name = texgyrepage11a`, since the latter string matches both families, while the former does not.

The name is lower-cased prior to lookup. The data used to generate actual font families uses a cleaned up version of the given name or, sometimes, a variation, depending on the data returned from the database.

For example, `tex gyre PaGe11A` yields the same results as `texgyrepage11a`.

Numbers are likewise removed³.

`force (key)` = `true | false`

Boolean which determines whether cached data is to be regenerated on the basis of a new search of the font-loader’s database.

Note that this doesn’t regenerate the loader’s database itself. It simply zaps data stored by this module for one or more font families.

`scale (key)` = `<factor>`

Optional factor by which the font should be scaled e.g. `20` or `0.8`. Ignored if a family includes distinct optical sizes.

2.2.2 Text-mode keys

These keys are accepted in all key-value specifications except for the top-level `math` key used to configure Unicode maths.

Currently there is only one key in this category.

`fea (key)` = `<opentype feature spec.>`

Optional list of font features **for text fonts only**.

Note that the specification is currently passed to the font loader with minimal processing, so you must use ‘raw’ feature specifications.

³This is problematic for fonts which depend on digits in their names to distinguish them e.g. Venturis ADF has four distinct ‘titling’ fonts, whose names differ by a single character, which is 1, 2, 3 or 4. The algorithm used to sort the fonts into NFSS families currently gets confused by this, leading to misleading warnings about ‘apparent duplicate’ fonts. (Something is indeed awry, but they are not duplicates. Rather what should be four families appears to be only one.)

The default is `fea= {\mode=node;scrpt=dflt;language=dflt;+tlig}`. Unless provided in separate fonts, `+smcp` is added, if available, for upright, italic and oblique small-caps.

Features apply to *all* fonts configured as a result of the lookup, so you should not specify features you do not wish applied to all fonts to which the specification applies.

2.2.3 Default text family keys

These keys are accepted by `rm`, `sf` and `tt` i.e. when specifying the default text families for the document.

These keys all do the same thing, so you may use your preferred syntactic sweetener.

`math (key) = true | false`

`maths (key)` Should the current specification also be used for the corresponding text family in maths?

For example, if you are configuring the main document family with `rm`, should that family also be used for operators, `\mathrm` etc.?

The default is equivalent to `math = true`.

`nomath (key)` Equivalent to `math = false`.

`nomaths (key)`

2.2.4 Additional text family keys

`command (key) = \command`

Use `\command` to switch to the font family, rather than creating a command from the family name.

By default the command to switch to the font family depends on the discovered family name i.e. on the result of searching the database for the specified family. In many cases, this will be the value passed to the `name` key, but often it will not and sometimes this may not be the command name you want anyway. `command` allows you to override it.

For example, the following would setup `\playfair` to switch to Playfair Display, rather than the default choice of `\playfairdisplay`.

```
\usepackage{lua-font-config}
\fontconfig{%
  family = {% setup an additional non-default family
    name = Playfair Display,
    command = \playfair,
  },
}
```

3 Compatibility

3.1 ltx-talk

Load with `\RequirePackage` *before* loading the class. The package emulates the class settings by default, so the main benefit of loading it is speed. If you wish to change the fonts, use `\fontconfig` after the class is loaded⁴

4 Examples

example-lmr -- fig. 1

```
\documentclass{article}
\usepackage{lua-font-config}
\fontconfig{rm = latinmodernroman}
\begin{document}
This is Latin Modern Roman.
It features \textit{italics} and \textsl{oblique}.
\textsc{Small-caps are implemented as distinct fonts, as are
  ↳ \textsl{oblique small-caps}.}
{\fontshape{ui}\selectfont Upright italics are less commonly used.}

\normalfont\bfseries
This is the bold expanded weight.
It features \textit{italics} and \textsl{oblique}.

\normalfont\fontseries{db}\selectfont
The ‘demi-bold’ is of regular width, with \textsl{oblique}.

\normalfont
Where optical sizes are available, thes provide higher quality
  ↳ output than simple scaling.

\footnotesize
This is footnote-sized Latin Modern Roman.
It features \textit{italics} and \textsl{oblique}.
\textsc{Small-caps are implemented as distinct fonts, as are
  ↳ \textsl{oblique small-caps}.}
{\fontshape{ui}\selectfont Upright italics is less commonly used.}

\normalfont\bfseries
This is the bold expanded weight.
It features \textit{italics} and \textsl{oblique}.

\normalfont\fontseries{db}\selectfont
The ‘demi-bold’ is of regular width, with upright and
  ↳ \textsl{oblique} shapes.
```

⁴If this turns out to be something people want to do, I can adjust things to avoid configuring fonts twice. For now, this is just a good way for me to test the code and avoid the pain of watching fontspec load New Computer Modern.

This is Latin Modern Roman. It features *italics* and *oblique*. SMALL-CAPS ARE IMPLEMENTED AS DISTINCT FONTS, AS ARE *OBLIQUE SMALL-CAPS*. Upright italics are less commonly used.

This is the bold expanded weight. It features *italics* and *oblique*.

The ‘demi-bold’ is of regular width, with *oblique*.

Where optical sizes are available, they provide higher quality output than simple scaling.

This is footnote-sized Latin Modern Roman. It features *italics* and *oblique*. SMALL-CAPS ARE IMPLEMENTED AS DISTINCT FONTS, AS ARE *OBLIQUE SMALL-CAPS*. Upright italics is less commonly used.

This is the bold expanded weight. It features *italics* and *oblique*.

The ‘demi-bold’ is of regular width, with upright and *oblique* shapes.

The output should look very much as if ‘lua-font-config’ had not been loaded!

Figure 1: example-lmr

```
\normalfont
The output should look very much as if ‘lua-font-config’ had not
→ been loaded!
\end{document}
```

```
example-ncm-math -- fig. 2
\documentclass{article}
\usepackage{lua-font-config}
\fontconfig{%
  rm = newcomputermodern,
  sf = newcomputermodernsans,
  tt = newcomputermodernmono,
  math = newcomputermodernmath,
}
\parindent=0pt
\begin{document}
\[
  \sin\theta^2 + \cos\theta^2 = 1
\]
$\mathsf{x}$ $\mathtt{x}$ $\mathbf{x}$

\mathversion{bold}
\[
  \sin\theta^2 + \cos\theta^2 = 1
\]
$\mathsf{x}$ $\mathtt{x}$ $\mathbf{x}$
\end{document}
```

```
example-ncm-complete -- fig. 3
\documentclass{article}
\usepackage{lua-font-config}
\fontconfig{%
  rm    = New Computer Modern,
  sf    = New Computer Modern Sans,
  tt    = New Computer Modern Mono,
  math  = New Computer Modern Math,
```

$$\sin \theta^2 + \cos \theta^2 = 1$$

xxx

$$\sin \theta^2 + \cos \theta^2 = 1$$

xxx

Figure 2: example-ncm-math

```

}
\begin{document}
New Computer Modern with default settings.

\sffamily
New Computer Modern Sans with default settings.

\ttfamily
New Computer Modern Mono with default settings.

\normalfont
New Computer Modern Math with text fonts as above:
\[
\sin\theta^2 + \cos\theta^2 = 1
\]
\[
\mathscr{ABC\ abc} \mathcal{ABC\ abc}
\]
\mathversion{bold}
\[
\sin\theta^2 + \cos\theta^2 = 1
\]
\[
\mathscr{ABC\ abc} \mathcal{ABC\ abc}
\]
\end{document}

```

```

\documentclass{article}
\parindent=0pt
\usepackage{lua-font-config}
\fontconfig{
  rm = {% serif uses oldstyle, proportional figures
    name = TeX Gyre Pagella,
    fea = {+ONUM;+PNUM}},
  sf = {% sans uses proportional lining figures
    name = texgyreheros,
    fea = {+pnum;+lnum}},
}

```

New Computer Modern with default settings.
 New Computer Modern Sans with default settings.
 New Computer Modern Mono with default settings.
 New Computer Modern Math with text fonts as above:

$$\sin \theta^2 + \cos \theta^2 = 1$$

ABCabcABCabc

$$\sin \theta^2 + \cos \theta^2 = 1$$

ABCabcABCabc

Figure 3: example-ncm-complete

```

tt = {% typewriter uses tabular lining figures and no ligatures
  name = texgyrecursor,
  fea = {-TLIG;+tnum;+lnum;-liga},
},
}
\begin{document}
{%
\TeX-Gyre Pagella: 0123456789
}

{%
\sffamily
\TeX-Gyre Heros: 0123456789
}

{%
\ttfamily
\TeX-Gyre Cursor 0123456789
-- --- outfitted officials affluently affiliate fish finials
  ↳ flagrantly, first
  finalising infinite floral affinities of fish-filled fingers
}

\l[
  \sin\quad
  \mathrm{rm}\quad
  \mathsf{sf}\quad
  \mathtt{tt}\quad
  \mathit{it}\quad
  \mathbf{bf}
\r[
\mathversion{bold}
\l[
  \sin\quad
  \mathrm{rm}\quad

```

$\mathrm{T}_{\mathrm{E}}\mathrm{X}$ -Gyre Pagella: 0123456789
 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ -Gyre Heros: 0123456789
 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ -Gyre Cursor 0123456789 -- --- outfitted officials
 affluently affiliate fish finials flagrantly, first final-
 ising infinite floral affinities of fish-filled fingers

sin rm sf tt *it* **bf**

sin rm **sf** tt *it* **bf**

Figure 4: example-fea

```

\mathsf{sf}\quad
\mathtt{tt}\quad
\mathit{it}\quad
\mathbf{bf}
\]
\end{document}

```

example-add -- fig. 5

```

\documentclass{article}
\usepackage{lua-font-config}
\fontconfig{%
  family = {
    name      = Lobster Two,
    scale     = 2,
    command   = \lobster,
  },
}
\begin{document}
This is some ordinary text in the default \LaTeXe{} font.

{\lobster This is Lobster Two.
  \textit{Italics but no \textsc{small-caps}} \textsc{even in}
  ↪ upright}.
We do have \textbf{bold and \textit{bold italics}}, but there is
  ↪ no,
  \textsl{oblique, so here we get italics, too.}\par
}

```

Back to the default.

```

\end{document}

```

example-talk -- fig. 6

```

\DocumentMetadata{tagging=on,pdfstandard=ua-2,pdfversion=2.0,
  ↪ lang=en-GB}
\RequirePackage{lua-font-config}% MUST precede class.
\documentclass{ltx-talk}

\title{Some Title}

```

This is some ordinary text in the default L^AT_EX 2_ε font.

This is Lobster Two. *Italics but no small-caps*
even in upright. We do have **bold and bold italics**, but there is no, *oblique*, so here we get *italics*, too.

Back to the default.

Figure 5: example-add

```
\author{A. U. Thor}
\institute{Pantheon, Greece}
\date{Ancient}
\begin{document}
\begin{frame}
  \maketitle
\end{frame}
\begin{frame}
  \frametitle{Frame}
  Some text\footnote{With a footnote.} on a slide.

  \textsc{Small-Caps \textsl{italics \textbf{bold}}} \textbf{bold}.)

  Regular, \fontseries{k}\selectfont Book and \bfseries Bold.
  \normalfont
  \begin{equation}
    v = ut + \frac{1}{2}at^2
  \end{equation}
  \begin{equation}
    \sin^2\theta + \cos^2\theta = 1
  \end{equation}
  Script $\mathscr{ABC\ abc}$ and calligraphic $\mathcal{ABC\ abc}$.

  Bold maths:
  \mathversion{bold}
  \begin{equation}
    v = ut + \frac{1}{2}at^2
  \end{equation}
  \begin{equation}
    \sin^2\theta + \cos^2\theta = 1
  \end{equation}
  Script $\mathscr{ABC\ abc}$ and calligraphic $\mathcal{ABC\ abc}$.
\end{frame}
\end{document}
```

Some Title

A. U. Thor
Pantheon, Greece
Ancient

Frame

Some text¹ on a slide.
SMALL-CAPS *ITALICS* **BOLD BOLD**.
Regular, Book and **Bold**.

$$v = ut + \frac{1}{2}at^2 \tag{1}$$

$$\sin^2 \theta + \cos^2 \theta = 1 \tag{2}$$

Script *ABCabc* and calligraphic *ABCabc*.

Bold maths:

$$v = ut + \frac{1}{2}at^2 \tag{3}$$

$$\mathbf{\sin^2 \theta + \cos^2 \theta = 1} \tag{4}$$

Script *ABCabc* and calligraphic *ABCabc*.

¹With a footnote.

Figure 6: example-talk

5 TO DOs

These are not in any particular order, though there is some tendency for higher priority items to be earlier.

1. Conceptual

- Abandon the whole thing since packages are not supposed to meddle with fonts?

2. NFSS

- Support multiple font configurations e.g. for multiple languages with multiple scripts.
 - Support for babel?
babel provides e.g. `\babelfont`, which uses `\fontspec`. Should lua-font-config provide this?

3. Interface

- Provide a proper user interface (more similar to ConT_EXt, since we're using the same (MKIV) sorting?)
- Information for user.
- Accommodate `.ttc`.
 - The basis is done, but not sure what, if any, interface is needed?
 - How to test this??
- Way to define individual font commands for text and (maybe) maths.
 - Text is already in the infrastructure, but the defined macro is unpredictable & non-configurable.

4. Loader

- Test with `.ttf`.
 - What needs testing exactly?? Or is this all on the loader?
- Variable fonts.
 - How to do or test??
- Colour. Do??
 - What does 'Do??' refer to?
- Cope with (text) symbol fonts? Not sure is that really needed?
- Combo, fallback & all that.

5. Cache

- Cached data should depend on db/fnt versions.
 - Or is this automatic?
 - Or maybe it shouldn't?

6. Documentation

- Not everything implemented in terms of the UI is in the user doc.
- The Lua code has some comments, but these don't always amount to documentation.
- The overall infrastructure isn't really documented, except for a partially accurate ASCII database schema included inline.
- Registered Lua functions mapped to expl3 names are not documented.
- Use of callbacks, hooks etc. is more-or-less undocumented.

7. Code

- Should be cleaned up – I cannot need 2,500 lines to load a font!

Unnumbered Gripe The code is too long, too complex, too clunky and too simplistic. (Yes, of course, it can be both.)

- Loading the ConTeXt file differently?
- Improve formal $\LaTeX$ representation (see ltluatexbase).
- Improve performance.
 - There is some duplication and too many pre-definitions for my taste.
- Possibly modify database creation to avoid sorting the data twice??

8. Test suite

- There are no unit tests (cf. nfssect-cfr, where the tests are much more focused). I don't know how to implement this for (non-public) Lua functions.

6 Implementation

There is very little here, because almost everything is done in Lua.

<*sty> <@@=lfc>

```
1 \NeedsTeXFormat{LaTeX2e}[2021-11-15]
2 \RequirePackage{svn-prov}
3 \ProvidesPackageSVN[\filebase.sty]{$Id: lua-font-config.dtx 12127 2026-
  10-15 06:49:18Z cfrees $}[v0.1 \revinfo][\filebase: Font configuration
  for LuaLaTeX]
4 % This is needed only because l3doc overwrites \fileversion and \filedate.
5 % Neither ltxdoc nor doc does this.
6 % But it seems better to do it this way on the basis of the general
  precautionary principle.
7 \DefineFileInfoSVN[lfc]

8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9 % copied verbatim, excepting format from Joseph Wright's' siunitx.sty
  under LPPL
10 \@ifundefined{directlua}{%
```

```

11 \PackageError{lua-font-config}{Package requires LuaLaTeX}
12 {To use lua-font-config, you must compile your document with LuaLaTeX.}%
13 {}
14 \@ifundefined{ExplLoaderFileDate}{%
15 \RequirePackage{expl3}%
16 {}
17 %% almost verbatim from siunitx.sty
18 \@ifl@t@r\ExplLoaderFileDate{2022-02-24}{%
19 }{%
20 \PackageError{lua-font-config}{Support package expl3 too old}
21 {%
22 You need to update your installation of the bundles 'l3kernel' and
23 'l3packages'.\MessageBreak
24 Loading lua-font-config will abort!%
25 }%
26 \endinput
27 }
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
30 \ExplSyntaxOn
31 \prop_gput:NnV \g_msg_module_name_prop {lfc} \lfc@base

```

Messages.

Note that `\msg_*` are not recognised in the indexing; `\msg` appears instead. This is a limitation of the doc package, as far as I know.

```

32 \msg_new:nnnn {lfc} {needs-luatex} {
33 \msg_critical_text:n {lfc} :: ~requires~LuaLaTeX~\msg_line_context:
34 } { \msg_see_documentation_text:n {lfc} }

35 \sys_if_engine luatex:F { \msg_critical:nn {lfc} {needs-luatex} }

36 \msg_new:nnnn {lfc} {math-needs-lum} {
37 \msg_error_text:n {lfc} :: ~math~config~requires~lua-unicode-math,~but~
38 you've~loaded~unicode-math.~This~cannot~work~\msg_line_context:
39 } { To~configure~Unicode~maths~fonts~with~this~package,~we~need~to~load~
40 lua-unicode-math,~but~unicode-math,~which~is~incompatible~has~already~
41 been~loaded,~so~I'm~going~to~ignore~this~part~of~your~settings.~
42 \msg_see_documentation_text:n {lfc}
43 }
44 \msg_new:nnnn {lfc} {loading} {
45 \msg_info_text:n {lfc} :: ~loading~#1~for~maths~support~
46 \msg_line_context:
47 } { \msg_see_documentation_text:n {lfc} }

```

```

\l__lfc_tmpa_tl (var.) Custom ephemeral variables.
\l__lfc_tmpa_bool (var.)
48 \tl_new:N \l__lfc_tmpa_tl
49 \bool_new:N \l__lfc_tmpa_bool

```

```

rm (key) Top-level keys for default text families.
sf (key)
tt (key) 50 \clist_map_inline:nn {rm, sf, tt} {
51 \keys_define:nn {lfc} {

```

```

52   #1 .code:n          = {
53     \tl_clear:N \l__lfc_tmpa_tl
54     \keys_set_exclude_groups:nnnN {lfc/#1} {config} {##1} \l__lfc_tmpa_tl
55     \keys_set:nV {lfc/#1} \l__lfc_tmpa_tl
56     \bool_set_true:N \l__lfc_tmpa_bool
57   },
58   #1 .value_required:n = true,
59   #1 .usage:n          = preamble,
60 }

```

name (key) Configure keys for default text families.

```

fea (key)
force (key) 61 \keys_define:nn {lfc/#1} {
scale (key) 62   name      .code:n = { \use:c {__lfc_set_#1:n} {##1} },
maths (key) 63   fea       .code:n = { \keys_set:nn {lfc/fam} {fea}      = {##1} } },
math (key)  64   force     .code:n = { \keys_set:nn {lfc/fam} {force}    = {##1} } },
nomaths (key) 65   scale     .code:n = { \keys_set:nn {lfc/fam} {scale}    = {##1} } },
nomath (key) 66   maths     .code:n = { \__lfc_set_fam_cfg:nn {maths}{##1} },
67   math      .meta:n = { maths = {##1} },
68   nomaths   .meta:n = { maths = false },
69   nomath    .meta:n = { nomaths },
70   unknown   .code:n = {
71     \exp_args:NnV \use:c {__lfc_set_#1:n} \l_keys_key_str
72   },
73   name      .value_required:n = true,
74   math      .default:n        = true,
75   maths     .default:n        = true,
76   nomath    .value_forbidden:n = true,
77   nomaths   .value_forbidden:n = true,
78   unknown   .value_forbidden:n = true,
79   fea       .groups:n         = config,
80   force     .groups:n         = config,
81   maths     .groups:n         = config,
82   math      .groups:n         = config,
83   nomath    .groups:n         = config,
84   nomaths   .groups:n         = config,
85   scale     .groups:n         = config,
86 }
87 }

```

debug (key) Top-level keys for additional families, Unicode maths and debugging.

```

family (key)
math (key) 88 \keys_define:nn {lfc} {
89   debug .code:n          = {
90     \tl_clear:N \l__lfc_tmpa_tl
91     \keys_set_groups:nnnN
92       {lfc/debug} {toggle} {state = true, #1} \l__lfc_tmpa_tl
93     \keys_set:nV {lfc/debug} \l__lfc_tmpa_tl
94   },
95   debug .usage:n         = general,
96   family .code:n         = {
97     \tl_clear:N \l__lfc_tmpa_tl
98     \keys_set_exclude_groups:nnnN {lfc/fam} {config} {##1} \l__lfc_tmpa_tl
99     \keys_set:nV {lfc/fam} \l__lfc_tmpa_tl

```

```

100   \bool_set_true:N \l__lfc_tmpa_bool
101 },
102 family .value_required:n = true,
103 family .usage:n          = general,
104 math    .code:n          = {
105   \IfPackageLoadedTF {unicode-math} {
106     \msg_error:nn {lfc} {math-needs-lum}
107   } {
108     \msg_info:nnn {lfc} {loading} {lua-unicode-math}
109     \RequirePackage{lua-unicode-math}
110     \tl_clear:N \l__lfc_tmpa_tl
111     \keys_set_exclude_groups:nnnN {lfc/math} {config} {#1} \l__lfc_tmpa_tl
112     \keys_set:nV {lfc/math} \l__lfc_tmpa_tl
113     \bool_set_true:N \l__lfc_tmpa_bool
114   }
115 },
116 math    .value_required:n = true,
117 math    .usage:n          = preamble,
118 }

```

name (*key*) Configure keys for families.

```

fea (key)
force (key)
scale (key)
119 \keys_define:nn {lfc/fam} {
120   name      .code:n = { \__lfc_set_fam_name:n {#1} },
121   command   .code:n = {
122     \exp_args:Nne \__lfc_set_fam_cfg:nn {csname}{ \cs_to_str:N #1 }
123   },
124   fea        .code:n = { \__lfc_set_fam_cfg:nn {fea}{#1} },
125   force      .code:n = { \__lfc_set_fam_cfg:nn {force}{#1} },
126   scale      .code:n = { \__lfc_set_fam_cfg:nn {scale}{#1} },
127   unknown    .code:n = {
128     \exp_args:NV \__lfc_set_fam_name:n \l_keys_key_str
129   },
130   name        .value_required:n = true,
131   fea         .value_required:n = true,
132   force       .default:n        = true,
133   scale       .value_required:n = true,
134   unknown     .value_forbidden:n = true,
135   name        .usage:n          = general,
136   fea         .usage:n          = general,
137   force       .usage:n          = general,
138   scale       .usage:n          = general,
139   unknown     .usage:n          = general,
140   fea         .groups:n         = config,
141   force       .groups:n         = config,
142   scale       .groups:n         = config,
143 }

```

name (*key*) Configure keys for Unicode maths.

```

force (key)
scale (key)
144 \IfPackageLoadedF {unicode-math} {
145   \keys_define:nn {lfc/math} {
146     name      .code:n = { \use:c {__lfc_set_math:n} {#1} },
147     force     .code:n = { \keys_set:nn {lfc/fam} {force} = {#1} },

```

```

148   scale      .code:n = { \keys_set:nn {lfc/fam} {scale} = {#1}} },
149   unknown    .code:n = {
150     \exp_args:NnV \use:c {__lfc_set_math:n} \l_keys_key_str
151   },
152   name       .value_required:n = true,
153   force      .default:n = true,
154   scale      .value_required:n = true,
155   unknown    .value_forbidden:n = true,
156   name       .usage:n = preamble,
157   force      .usage:n = preamble,
158   scale      .usage:n = preamble,
159   unknown    .usage:n = preamble,
160   force      .groups:n = config,
161   scale      .groups:n = config,
162 }
163 }

```

state (key) Configure keys for debugging.
categories (key)

```

164 \keys_define:nn {lfc/debug} {
165   categories .code:n = { \__lfc_debug_set_cats_scanner:n {#1} },
166   state      .choice:,
167   state/false .code:n = { \__lfc_debug_set_false: },
168   state/true  .code:n = { \__lfc_debug_set_true: },
169   categories .usage:n = general,
170   state      .usage:n = general,
171   categories .value_required:n = true,
172   state      .default:n = true,
173   state      .groups:n = toggle,
174 }

```

\fontconfig The main (and currently(?) only) public macro.

```

175 \NewDocumentCommand\fontconfig { +m }{
176   \bool_set_false:N \l__lfc_tmpa_bool
177   \keys_set:nn {lfc} {#1}
178   \bool_if:NT \l__lfc_tmpa_bool {
179     \__lfc_configure_doc_families:
180   }
181 }

```

__lfc_set_mathrm: (fn.) TODO: move to hooks if module gets support for switching the families out
 __lfc_set_mathsf: (fn.) e.g. for language changes. Currently, the code is too simple-minded for this to
 __lfc_set_mathtt: (fn.) arise

```

182 \cs_new_protected_nopar:Npn \__lfc_set_mathrm: {
183   \SetMathAlphabet \mathrm {normal} {TU} \rmdefault {\mdseries@rm}
   \shapedefault
184   \SetMathAlphabet \mathit {normal} {TU} \rmdefault {\mdseries@rm} \it-
   default
185   \SetMathAlphabet \mathbf {normal} {TU} \rmdefault {\bfseries@rm}
   \shapedefault

```

```

186 \SetMathAlphabet \mathrm {bold}      {TU} \rmdefault {\bfseries@rm}
    \shapedefault
187 \SetMathAlphabet \mathit {bold}      {TU} \rmdefault {\bfseries@rm} \it-
    default
188 \SetMathAlphabet \mathbf {bold}      {TU} \rmdefault {\bfseries@rm}
    \shapedefault
189 \SetSymbolFont{operators}{normal} {TU} \rmdefault {\mdseries@rm} \shapedefault
190 \SetSymbolFont{operators}{bold}    {TU} \rmdefault {\bfseries@rm}
    \shapedefault
191 }
192 \cs_new_protected_nopar:Npn \__lfc_set_mathsf: {
193   \SetMathAlphabet \mathsf {normal} {TU} \sfdefault {\mdseries@sf}
    \shapedefault
194   \SetMathAlphabet \mathsf {bold}    {TU} \sfdefault {\bfseries@sf}
    \shapedefault
195 }
196 \cs_new_protected_nopar:Npn \__lfc_set_mathtt: {
197   \SetMathAlphabet \mathtt {normal} {TU} \ttdefault {\mdseries@tt}
    \shapedefault
198   \SetMathAlphabet \mathtt {bold}    {TU} \ttdefault {\bfseries@tt}
    \shapedefault
199 }

```

`\declare_lum_symbol_fonts:nn` (*fn.*) Maths setup function.

```

200 \cs_new_protected_nopar:Npn \__lfc_declare_lum_symbol_fonts:nn #1#2 {
201   \DeclareSymbolFont{lummain}{TU}{#1}{m}{n}
202   \SetSymbolFont{lummain}{bold}{TU}{#2}{b}{n}
203 }

```

Compatibility code for ltx-talk.

This should be equivalent to the default setup, modulo any differences between emulating the package and using `\setmathfont` for lua-unicode-math.

```

204 \hook_gput_code:nnn {class/ltx-talk/before} {.} {
205   \hook_gput_code:nnn {package/mathtools/after} {.} {
206     \RequirePackage{lua-unicode-math}
207     \fontconfig{
208       rm      = {newcomputermodernsans},
209       sf      = {newcomputermodernsans},
210       math    = {newcomputermodernsansmath},
211     }
212     \tagpdfsetup { math / mathml / luamml / load = true }
213   }
214 }

```

`\__lfc_set_fam_cfg:nn` (*fn.*) The Lua modules do the actual work.

```

\__lfc_set_fam_name:n (fn.)
  \__lfc_set_mathrm: (fn.) 215 \lua_load_module:n {lua-font-config}
  \__lfc_set_mathsf: (fn.) 216 \ExplSyntaxOff
  \__lfc_set_mathtt: (fn.)

```

`\__lfc_set_fam_cfg:nn` (*fn.*)

`\__lfc_set_fam_name:n` (*fn.*)

`\__lfc_set_mathrm:` (*fn.*)

`\__lfc_set_mathsf:` (*fn.*)

`\__lfc_set_mathtt:` (*fn.*)

Change History

v0.1

General: First public release. 14

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols	
<code>__lfc_configure_doc_families:</code> (fn.)	179, 215
<code>__lfc_debug_set_cats_scanner:n</code> (fn.)	165, 215
<code>__lfc_debug_set_false: (fn.)</code>	167, 215
<code>__lfc_debug_set_true: (fn.)</code>	168, 215
<code>__lfc_declare_lum_symbol_fonts:nn</code> (fn.)	200
<code>__lfc_set_fam_cfg:nn (fn.)</code>	66, 122, 124, 125, 126, 215
<code>__lfc_set_fam_name:n (fn.)</code>	120, 128, 215
<code>__lfc_set_mathrm: (fn.)</code>	182, 215
<code>__lfc_set_mathsf: (fn.)</code>	182, 215
<code>__lfc_set_mathtt: (fn.)</code>	182, 215
B	
<code>\bfseries@rm</code> . .	185, 186, 187, 188, 190
<code>\bfseries@sf</code>	194
<code>\bfseries@tt</code>	198
<code>\bool_if:NT</code>	178
<code>\bool_new:N</code>	49
<code>\bool_set_false:N</code>	176
<code>\bool_set_true:N</code>	56, 100, 113
C	
<code>categories (key)</code>	164
<code>\clist_map_inline:nn</code>	50
<code>command (key)</code>	5
<code>\cs_new_protected_nopar:Npn</code>	182, 192, 196, 200
<code>\cs_to_str:N</code>	122
D	
<code>debug (key)</code>	3, 88
<code>\DeclareSymbolFont</code>	201
E	
<code>\exp_args:Nne</code>	122
<code>\exp_args:NnV</code>	71, 150
<code>\exp_args:NV</code>	128
F	
<code>fam (key)</code>	3, 3
<code>family (key)</code>	88
<code>fea (key)</code>	4, 61, 119
<code>\fontconfig</code>	2, 175, 207
<code>force (key)</code>	4, 61, 119, 144
functions:	
<code>__lfc_configure_doc_families:</code>	179, 215
<code>__lfc_debug_set_cats_scanner:n</code>	165, 215
<code>__lfc_debug_set_false: .</code>	167, 215
<code>__lfc_debug_set_true: .</code>	168, 215
<code>__lfc_declare_lum_symbol_fonts:nn</code>	200
<code>__lfc_set_fam_cfg:nn</code>	66, 122, 124, 125, 126, 215
<code>__lfc_set_fam_name:n</code>	120, 128, 215
<code>__lfc_set_mathrm:</code>	182, 215
<code>__lfc_set_mathsf:</code>	182, 215
<code>__lfc_set_mathtt:</code>	182, 215
G	
<code>\g_msg_module_name_prop</code>	31
H	
<code>\hook_gput_code:nnn</code>	204, 205
I	
<code>\IfPackageLoadedF</code>	144
<code>\IfPackageLoadedTF</code>	105
<code>\itdefault</code>	184, 187
K	
keys:	
<code>categories</code>	164
<code>command</code>	5
<code>debug</code>	3, 88
<code>fam</code>	3, 3

family	88		
fea	4, 61, 119		
force	4, 61, 119, 144		
math	3, 3, 5, 61, 88		
maths	5, 61		
name	4, 61, 119, 144		
nomath	5, 61		
nomaths	5, 61		
rm	2, 2, 50		
scale	4, 61, 119, 144		
sf	2, 2, 50		
state	164		
tt	2, 2, 50		
\keys_define:nn	51, 61, 88, 119, 145, 164		
\keys_set:nn	63, 64, 65, 147, 148, 177		
\keys_set:nV	55, 93, 99, 112		
\keys_set_exclude_groups:nnnN	54, 98, 111		
\keys_set_groups:nnnN	91		
L			
\l_lfc_tmpa_bool (var.)	48, 56, 100, 113, 176, 178		
\l_lfc_tmpa_tl (var.)	48, 53, 54, 55, 90, 92, 93, 97, 98, 99, 110, 111, 112		
\l_keys_key_str	71, 128, 150		
\lfc@base	31		
\lua_load_module:n	215		
M			
math (key)	3, 3, 5, 61, 88		
\mathbf	185, 188		
\mathit	184, 187		
\mathrm	183, 186		
maths (key)	5, 61		
\mathsf	193, 194		
\mathtt	197, 198		
\mdseries@rm	183, 184, 189		
\mdseries@sf	193		
\mdseries@tt	197		
\msg_critical:nn	35		
\msg_error:nn	106		
\msg_info:nnn	108		
N			
name (key)	4, 61, 119, 144		
\NewDocumentCommand	175		
nomath (key)	5, 61		
nomaths (key)	5, 61		
P			
\prop_gput:NnV	31		
R			
rm (key)	2, 2, 50		
\rmdefault	183, 184, 185, 186, 187, 188, 189, 190		
S			
scale (key)	4, 61, 119, 144		
\SetMathAlphabet	183, 184, 185, 186, 187, 188, 193, 194, 197, 198		
\SetSymbolFont	189, 190, 202		
sf (key)	2, 2, 50		
\sfdefault	193, 194		
\shapedefault	183, 185, 186, 188, 189, 190, 193, 194, 197, 198		
state (key)	164		
\sys_if_engine luatex:F	35		
T			
\tagpdfsetup	212		
\tl_clear:N	53, 90, 97, 110		
\tl_new:N	48		
tt (key)	2, 2, 50		
\ttdefault	197, 198		
U			
\use:c	62, 71, 146, 150		
V			
variables:			
\l_lfc_tmpa_bool	48, 56, 100, 113, 176, 178		
\l_lfc_tmpa_tl	48, 53, 54, 55, 90, 92, 93, 97, 98, 99, 110, 111, 112		