Lateef - Developer information

Welcome font developers!

We welcome other developers who wish to get involved in supporting and enhancing these fonts or who want to modify them.

Permissions granted by the OFL

SIL's fonts are licensed according to the terms of the SIL Open Font License. The OFL allows the fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. For details see the OFL.txt and OFL-FAQ.txt files in the package.

Building the fonts from source code

Font sources are published in a Github project. The build process requires smith and project build parameters are set in the wscript.

Font sources are in the UFO3 format with font family structures defined using designspace. OpenType source code is stored in the .fea format in the UFO (features.fea) but is maintained in a separate file using the more efficient and powerful .feax format.

The fonts are built using a completely free and open source workflow using industry-standard tools (fonttools), a package of custom python scripts (pysilfont), and a build and packaging system (Smith). The whole toolchain is available as a Docker container.

Full instructions for setting up the tools and building SIL fonts are available on a dedicated web site: SIL Font Development Guide.

Building

The Lateef project can be built from source using smith. This is done via the sequence:

```
smith distclean
smith configure
smith build
smith alltests
```

Because of the complex kerning and collision avoidance logic, builds can take up to 15 minutes or longer, depending on hardware. If the complex kerning is *not* needed (such as for debugging other font logic), the `--quick` parameter can be supplied:

```
smith distclean
smith configure
smith build --quick
```

The resulting files will not have functional kerning or collision avoidance, but will be otherwise usable.

Adding characters

After base characters to the font, the following files will also need updating: - `glyph_data.csv` -- used to set glyph orders and psnames in the built font - `classes.xml` -- used to define classes used by OpenType. Note that some of the classes defined therein are noted to be "automatically generated" -- these will be updated (from glyph_data.csv) the next time `./preflight` is run. - `opentype/*.feax` -- modify as needed to add needed OpenType behavior - `tests/*.ftml` -- see below

Generated test files

After adding characters or additional behaviors to the font, test files should be created or enhanced to test the new behaviors. The test files: - `tests/AllChars-auto.ftml` - `tests/ALsorted-auto.ftml` - `tests/DaggerAlef-auto.ftml` - `tests/Diac1-auto.ftml` - `tests/Diac1-short-auto.ftml` - `tests/FeatLang-auto.ftml` - `tests/Kern-auto.ftml` - `tests/Kern-long-auto.ftml` - `tests/KernDigits-auto.ftml` - `tests/SubtendingMarks-auto.ftml` - `tests/Yehbarree-auto.ftml`

are generated automatically using `tools/bin/genftmlfiles.sh`.

`tools/ftml.xsl` can be used to view ftml documents directly in Firefox (which supports both Graphite and OpenType rendering).

About ftml tests

After a successful build, the results/ folder will contain, along with the built ttf and woff fonts, a number of test files in an xml-based format called FTML. Examples are AllChars-auto.ftml, DiacTest1-auto.ftml. There is an ftml.xsl file that can be used to view these ftml documents directly in Firefox.

However, in order for Firefox to access the .xsl file, you need to relax its "strict URI" policy by going to about:config and setting security.fileuri.strict_origin_policy to false.

Once you have this setting in effect, you can load the FTML documents directly into Firefox and see the built font rendered.

Contributing to the project

We warmly welcome contributions to the fonts, such as new glyphs, enhanced smart font code, or bug fixes. The brief overview of contributing changes is a good place to begin. The next step is to contact us by responding to an existing issue or creating an issue in the Github repository and expressing your interest. We can then work together to plan and integrate your contributions.

To enable us to accept contributions in a way that honors your contribution and respects your copyright while preserving long-term flexibility for open source licensing, you would also need to agree to the **SIL International Contributor License Agreement for Font Software (v1.0)** prior to sending us your contribution. To read more about this requirement and find out how to submit the required form, please visit the CLA information page.