C:\Users\david/src\Ligaturizer\.gitignore

Created by https://www.gitignore.io/api/font,linux,macos,python,fontforge,pycharm+all,sublimetext,visuals

FontForge

.sfd-

*.sfd~

No Binaries

*.ttc

*.pfb

*.pfa

*.pt3

*.suit

Linux

*~

temporary files which can be created if a process still has a handle open of a deleted file

.fuse_hidden*

KDE directory preferences

.directory

Linux trash folder which might appear on any partition or disk

.Trash-*

```
# .nfs files are created when an open file is removed but is still being accessed
.nfs*

### macOS ###
*.DS_Store
.AppleDouble
.LSOverride

# Icon must end with two \r
Icon

# Thumbnails
._*

# Files that might appear in the root of a volume
.DocumentRevisions-V100
.fseventsd
.Spotlight-V100
.TemporaryItems
.Trashes
.VolumeIcon.icns
.com.apple.timemachine.donotpresent

# Directories potentially created on remote AFP share
.AppleDB
.AppleDesktop
```

Network Trash Folder

Temporary Items

.apdisk

PyCharm+all

Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio and W

Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839

User-specific stuff:

.idea/**/workspace.xml

.idea/**/tasks.xml

.idea/dictionaries

Sensitive or high-churn files:

.idea/**/dataSources/

.idea/**/dataSources.ids

.idea/**/dataSources.xml

.idea/**/dataSources.local.xml

.idea/**/sqlDataSources.xml

.idea/**/dynamic.xml

.idea/**/uiDesigner.xml

Gradle:

.idea/**/gradle.xml

.idea/**/libraries

CMake

```
cmake-build-debug/

# Mongo Explorer plugin:
.idea/**/mongoSettings.xml

## File-based project format:
*.iws

## Plugin-specific files:

# IntelliJ
/out/

# mpeltonen/sbt-idea plugin
.idea_modules/

# JIRA plugin
atlassian-ide-plugin.xml

# Cursive Clojure plugin
.idea/replstate.xml

# Crashlytics plugin (for Android Studio and IntelliJ)
com_crashlytics_export_strings.xml
crashlytics.properties
crashlytics-build.properties
fabric.properties
```

```
### PyCharm+all Patch ###
# Ignores the whole idea folder
# See https://github.com/joeblau/gitignore.io/issues/186 and https://github.com/joeblau/gitignore.io/issues/3

.idea/

### Python ###
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
env/
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
```

```
parts/
sdist/
var/
wheels/
*.egg-info/
.installed.cfg
*.egg

# PyInstaller
#  Usually these files are written by a python script from a template
#  before PyInstaller builds the exe, so as to inject date/other infos into it.
*.manifest
*.spec

# Installer logs
pip-log.txt
pip-delete-this-directory.txt

# Unit test / coverage reports
htmlcov/
.tox/
.coverage
.coverage.*
.cache
nosetests.xml
coverage.xml
*,cover
```

```
.hypothesis/

# Translations
*.mo
*.pot

# Django stuff:
*.log
local_settings.py

# Flask stuff:
instance/
.webassets-cache

# Scrapy stuff:
.scrapy

# Sphinx documentation
docs/_build/

# PyBuilder
target/

# Jupyter Notebook
.ipynb_checkpoints

# pyenv
```

```
.python-version

# celery beat schedule file
celerybeat-schedule

# SageMath parsed files
*.sage.py

# dotenv
.env

# virtualenv
.venv
venv/
ENV/

# Spyder project settings
.spyderproject
.spyproject

# Rope project settings
.ropeproject

# mkdocs documentation
/site

### SublimeText ###
```

```
# cache files for sublime text
*.tmlanguage.cache
*.tmPreferences.cache
*.stTheme.cache

# workspace files are user-specific
*.sublime-workspace

# project files should be checked into the repository, unless a significant
# proportion of contributors will probably not be using SublimeText
# *.sublime-project

# sftp configuration file
sftp-config.json

# Package control specific files
Package Control.last-run
Package Control.ca-list
Package Control.ca-bundle
Package Control.system-ca-bundle
Package Control.cache/
Package Control.ca-certs/
Package Control.merged-ca-bundle
Package Control.user-ca-bundle
oscrypto-ca-bundle.crt
bh_unicode_properties.cache
```

```
# Sublime-github package stores a github token in this file
# https://packagecontrol.io/packages/sublime-github
GitHub.sublime-settings

### VisualStudioCode ###
.vscode/*
!.vscode/settings.json
!.vscode/tasks.json
!.vscode/launch.json
!.vscode/extensions.json
.history

# End of https://www.gitignore.io/api/font,linux,macos,python,fontforge,pycharm+all,sublimetext,visualstudi
```

C:\Users\david/src\Ligaturizer\.gitmodules

```
[submodule "fira"]
 path = fonts/fira
 url = https://github.com/tonsky/FiraCode.git
 shallow = true
[submodule "codeface"]
 path = fonts/codeface
 url = https://github.com/chrissimpkins/codeface.git
 shallow = true
[submodule "plex"]
 path = fonts/plex
 url = https://github.com/IBM/plex.git
 shallow = true
[submodule "fonts/spacemono"]
 path = fonts/spacemono
 url = https://github.com/googlefonts/spacemono.git
[submodule "fonts/Montserrat"]
 path = fonts/Montserrat
 url = https://github.com/JulietaUla/Montserrat.git
```

C:\Users\david/src\Ligaturizer\build.py

```
#!/usr/bin/env python
#
# Rebuild script for ligaturized fonts.
# Uses ligaturize.py to do the heavy lifting; this file basically just contains
# the mapping from input font paths to output fonts.

#### User configurable settings ####

# For the prefixed_fonts below, what word do we stick in front of the font name?
LIGATURIZED_FONT_NAME_PREFIX = "Liga"

# Should we copy some individual punctuations characters like &, ~, and <>,
# as well as ligatures? The full list is in ligatures.py.
# You can also override this (and OUTPUT_DIR) automatically by passing
# --copy-character-glyphs on the command line.
COPY_CHARACTER_GLYPHS = False

# If copying individual characters, how different in width (relative to the font
# we're ligaturizing) should they be before we attempt to width-correct them?
# The default (0.1) means to width-correct if they're +/- 10%. Values >1.0
# effectively disable this feature.
SCALE_CHARACTER_GLYPHS_THRESHOLD = 0.1

# Where to put the generated fonts.
OUTPUT_DIR = "fonts/output/"
```

```
#### Fonts that should be prefixed with "Liga" when ligaturized. ####
# Don't put fonts licensed under UFL here, and don't put fonts licensed under
# SIL OFL here either unless they haven't specified a Reserved Font Name.

prefixed_fonts = [
    # Apache 2.0 license
    "fonts/codeface/fonts/cousine/*.ttf",
    "fonts/codeface/fonts/droid-sans-mono/*.ttf",
    "fonts/codeface/fonts/meslo/*.ttf",
    "fonts/codeface/fonts/roboto-mono/*.ttf",
    # MIT license
    "fonts/codeface/fonts/dejavu-sans-mono/*.ttf",
    "fonts/codeface/fonts/hack/*.ttf",
    # SIL OFL with no Reserved Font Name
    "fonts/codeface/fonts/edlo/*.ttf",
    "fonts/codeface/fonts/inconsolata/*.ttf",
    "fonts/spacemono/fonts/*.ttf",
    "fonts/Montserrat/fonts/otf/*.otf",
    "fonts/Montserrat/fonts/ttf/*.ttf",
    "fonts/SauceCodePro**",
]

#### Fonts that need to be renamed. ####
# These are fonts that either have name collisions with the prefixed_fonts
# above, or are released under licenses that permit modification only if we
# change the name of the modified fonts.
```

```
renamed_fonts = {
    # This doesn't have a reserved name, but if we don't rename it it'll collide
    # with its sibling Fantasque Sans Mono Normal, listed above.
    "fonts/FantasqueSansMono-Normal/*.otf": "Liga Fantasque Sans Mono",
    "fonts/FantasqueSansMono-Normal/*.ttf": "Liga Fantasque Sans Mono",
    "fonts/FantasqueSansMono-NoLoopK/*.otf": "Liga Fantasque Sans Mono NoLoopK",
    "fonts/FantasqueSansMono-NoLoopK/*.ttf": "Liga Fantasque Sans Mono NoLoopK",
    "fonts/FantasqueSansMono-LargeLineHeight/*.otf": "Liga Fantasque Sans Mono LargeLineHeight",
    "fonts/FantasqueSansMono-LargeLineHeight/*.ttf": "Liga Fantasque Sans Mono LargeLineHeight",
    "fonts/FantasqueSansMono-LargeLineHeight-NoLoopK/*.otf": "Liga Fantasque Sans Mono LargeLineHe
    "fonts/FantasqueSansMono-LargeLineHeight-NoLoopK/*.ttf": "Liga Fantasque Sans Mono LargeLineHei
    # SIL OFL with reserved name
    "fonts/codeface/fonts/anonymous-pro/*.ttf": "Liganymous",
    "fonts/plex/IBM-Plex-Mono/fonts/complete/ttf/*.ttf": "Ligalex Mono",
    "fonts/codeface/fonts/oxygen-mono/*.otf": "Liga O2 Mono",
    "fonts/codeface/fonts/source-code-pro/*.ttf": "LigaSrc Pro",
    "fonts/SourceCodeVariable*": "LigaSrc Variable",
    "fonts/Hermit/*.otf": "Ligamit",
    # UFL
    "fonts/codeface/fonts/ubuntu-mono/*.ttf": "Ubuntu Mono Ligaturized",
}

#### Fonts we can't ligaturize. ####
# Fonts that we can't ligaturize because their licences do not permit derivative
# works of any kind.
# Individual users may still be able to make ligaturized versions for personal
# use, but we can't check them into the repo or include them in releases.
```

```
# prefixed_fonts += [
#   'CamingoCode*',
#   'SFMono*',
# ]

#### No user serviceable parts below this line. ####

import sys
from glob import glob
from ligaturize import ligaturize_font

if "--copy-character-glyphs" in sys.argv:
    COPY_CHARACTER_GLYPHS = True
    OUTPUT_DIR = "fonts/output-with-characters"



for pattern in prefixed_fonts:
    files = glob(pattern)
    if not files:
        print("Error: pattern '%s' didn't match any files." % pattern)
        sys.exit(1)
    for input_file in files:
        ligaturize_font(
            input_file,
            ligature_font_file=None,
            output_dir=OUTPUT_DIR,
```

```
            prefix=LIGATURIZED_FONT_NAME_PREFIX,
            output_name=None,
            copy_character_glyphs=COPY_CHARACTER_GLYPHS,
            scale_character_glyphs_threshold=SCALE_CHARACTER_GLYPHS_THRESHOLD,
        )


for pattern, name in renamed_fonts.items():
    files = glob(pattern)
    if not files:
        print("Error: pattern '%s' didn't match any files." % pattern)
        sys.exit(1)
    for input_file in files:
        ligaturize_font(
            input_file,
            ligature_font_file=None,
            output_dir=OUTPUT_DIR,
            prefix=None,
            output_name=name,
            copy_character_glyphs=COPY_CHARACTER_GLYPHS,
            scale_character_glyphs_threshold=SCALE_CHARACTER_GLYPHS_THRESHOLD,
        )
```

C:\Users\david/src\Ligaturizer\char_dict.py

```
char_dict = {
    'ampersand': '&',
    'asciicircum': '^',
    'asciitilde': '~',
    'asterisk': '*',
    'backslash': '\\',
    'bar': '|',
    'colon': ':',
    'equal': '=',
    'exclam': '!',
    'greater': '>',
    'hyphen': '-',
    'less': '<',
    'numbersign': '#',
    'percent': '%',
    'period': '.',
    'plus': '+',
    'question': '?',
    'semicolon': ';',
    'slash': '/',
    'underscore': '_',
    'at': '@',
    'braceleft': '{',
    'braceright': '}',
    'bracketleft': '[',
    'bracketright': ']',
```

```
    'dollar': '$',
    'parenleft': '(',
    'parenright': ')',
    'w': 'w',
}
```

C:\Users\david/src\Ligaturizer\LICENSE

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new

free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic

pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this

License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the

work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all

the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your

rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention

is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other

parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the

written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied

by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately

under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions;

the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently

reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could

give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will

be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the program's name and a brief idea of what it does.>
> Copyright (C) <year> <name of author>
>
> This program is free software: you can redistribute it and/or modify
> it under the terms of the GNU General Public License as published by
> the Free Software Foundation, either version 3 of the License, or
> (at your option) any later version.
>
> This program is distributed in the hope that it will be useful,
> but WITHOUT ANY WARRANTY; without even the implied warranty of
> MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
> GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License

    along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

    <program> Copyright (C) <year> <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License. But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.

```
C:\Users\david/src\Ligaturizer\ligatures.py
## This is the master list of ligatures that ligaturize.py will attempt to copy
## from Fira Code to your output font. Ligatures that aren't present in the
## version of Fira Code you're using will be skipped.
## To disable ligatures, simply comment them out in this file.
ligatures = [
    {
        ## These are all the punctuation characters used in Fira Code ligatures.
        ## Use the `--copy-character-glyphs` option to copy these into the output
        ## font along with the ligatures themselves.
        'chars': [
            ## These characters generally look good in most fonts and are
            ## enabled by default if you use `--copy-character-glyphs`.
            'ampersand', 'asciicircum', 'asciitilde', 'asterisk',
            'backslash', 'bar',
            'colon', 'equal', 'exclam', 'greater', 'hyphen',
            'less', 'numbersign', 'percent', 'period', 'plus',
            'question', 'semicolon', 'slash', 'underscore',

            ## These characters are also used by the ligatures, but are likely
            ## to look more out of place when spliced into another font.
            # 'at', 'braceleft', 'braceright', 'bracketleft', 'bracketright',
            # 'dollar', 'parenleft', 'parenright', 'underscore', 'w'
        ],
        'firacode_ligature_name': None,
    },
    ## These are traditional (i.e. present in most variable-width fonts)
```

```
## aesthetic ligatures. They are commented out here so that they don't
## overwrite similar ligatures present in the destination font.
# {   # Fl
#     'chars': ['F', 'l'],
#     'firacode_ligature_name': 'F_l.liga',
# },
# {   # Tl
#     'chars': ['T', 'l'],
#     'firacode_ligature_name': 'T_l.liga',
# },
# {   # fi
#     'chars': ['f', 'i'],
#     'firacode_ligature_name': 'f_i.liga',
# },
# {   # fj
#     'chars': ['f', 'j'],
#     'firacode_ligature_name': 'f_j.liga',
# },
# {   # fl
#     'chars': ['f', 'l'],
#     'firacode_ligature_name': 'f_l.liga',
# },
# {   # ft
#     'chars': ['f', 't'],
#     'firacode_ligature_name': 'f_t.liga',
# },
## Programming ligatures begin here.
```

```
{   # &&
    'chars': ['ampersand', 'ampersand'],
    'firacode_ligature_name': 'ampersand_ampersand.liga',
},
{   # ^=
    'chars': ['asciicircum', 'equal'],
    'firacode_ligature_name': 'asciicircum_equal.liga',
},
{   # ~~
    'chars': ['asciitilde', 'asciitilde'],
    'firacode_ligature_name': 'asciitilde_asciitilde.liga',
},
{   # ~~>
    'chars': ['asciitilde', 'asciitilde', 'greater'],
    'firacode_ligature_name': 'asciitilde_asciitilde_greater.liga',
},
{   # ~@
    'chars': ['asciitilde', 'at'],
    'firacode_ligature_name': 'asciitilde_at.liga',
},
{   # ~=
    'chars': ['asciitilde', 'equal'],
    'firacode_ligature_name': 'asciitilde_equal.liga',
},
{   # ~>
    'chars': ['asciitilde', 'greater'],
    'firacode_ligature_name': 'asciitilde_greater.liga',
```

```
},
{   # ~-
    'chars': ['asciitilde', 'hyphen'],
    'firacode_ligature_name': 'asciitilde_hyphen.liga',
},
{   # **
    'chars': ['asterisk', 'asterisk'],
    'firacode_ligature_name': 'asterisk_asterisk.liga',
},
{   # ***
    'chars': ['asterisk', 'asterisk', 'asterisk'],
    'firacode_ligature_name': 'asterisk_asterisk_asterisk.liga',
},
{   # *>
    'chars': ['asterisk', 'greater'],
    'firacode_ligature_name': 'asterisk_greater.liga',
},
{   # */
    'chars': ['asterisk', 'slash'],
    'firacode_ligature_name': 'asterisk_slash.liga',
},
{   # \/
    'chars': ['backslash', 'slash'],
    'firacode_ligature_name': 'backslash_slash.liga',
},
{   # ||
    'chars': ['bar', 'bar'],
```

```
    'firacode_ligature_name': 'bar_bar.liga',
},
{   # |||>
    'chars': ['bar', 'bar', 'bar', 'greater'],
    'firacode_ligature_name': 'bar_bar_bar_greater.liga',
},
{   # ||=
    'chars': ['bar', 'bar', 'equal'],
    'firacode_ligature_name': 'bar_bar_equal.liga',
},
{   # ||>
    'chars': ['bar', 'bar', 'greater'],
    'firacode_ligature_name': 'bar_bar_greater.liga',
},
{   # ||-
    'chars': ['bar', 'bar', 'hyphen'],
    'firacode_ligature_name': 'bar_bar_hyphen.liga',
},
{   # |}
    'chars': ['bar', 'braceright'],
    'firacode_ligature_name': 'bar_braceright.liga',
},
{   # |]
    'chars': ['bar', 'bracketright'],
    'firacode_ligature_name': 'bar_bracketright.liga',
},
{   # |=
```

```
    'chars': ['bar', 'equal'],
    'firacode_ligature_name': 'bar_equal.liga',
},
{   # |=>
    'chars': ['bar', 'equal', 'greater'],
    'firacode_ligature_name': 'bar_equal_greater.liga',
},
{   # |>
    'chars': ['bar', 'greater'],
    'firacode_ligature_name': 'bar_greater.liga',
},
{   # |-
    'chars': ['bar', 'hyphen'],
    'firacode_ligature_name': 'bar_hyphen.liga',
},
{   # |->
    'chars': ['bar', 'hyphen', 'greater'],
    'firacode_ligature_name': 'bar_hyphen_greater.liga',
},
{   # {|
    'chars': ['braceleft', 'bar'],
    'firacode_ligature_name': 'braceleft_bar.liga',
},
{   # [|
    'chars': ['bracketleft', 'bar'],
    'firacode_ligature_name': 'bracketleft_bar.liga',
},
```

```
{   # ]#
    'chars': ['bracketright', 'numbersign'],
    'firacode_ligature_name': 'bracketright_numbersign.liga',
},
{   # ::
    'chars': ['colon', 'colon'],
    'firacode_ligature_name': 'colon_colon.liga',
},
{   # :::
    'chars': ['colon', 'colon', 'colon'],
    'firacode_ligature_name': 'colon_colon_colon.liga',
},
{   # ::=
    'chars': ['colon', 'colon', 'equal'],
    'firacode_ligature_name': 'colon_colon_equal.liga',
},
{   # :=
    'chars': ['colon', 'equal'],
    'firacode_ligature_name': 'colon_equal.liga',
},
{   # :>
    'chars': ['colon', 'greater'],
    'firacode_ligature_name': 'colon_greater.liga',
},
{   # :<
    'chars': ['colon', 'less'],
    'firacode_ligature_name': 'colon_less.liga',
```

```
},
{   # $>
    'chars': ['dollar', 'greater'],
    'firacode_ligature_name': 'dollar_greater.liga',
},
{   # =:=
    'chars': ['equal', 'colon', 'equal'],
    'firacode_ligature_name': 'equal_colon_equal.liga',
},
{   # ==
    'chars': ['equal', 'equal'],
    'firacode_ligature_name': 'equal_equal.liga',
},
{   # ===
    'chars': ['equal', 'equal', 'equal'],
    'firacode_ligature_name': 'equal_equal_equal.liga',
},
{   # ==>
    'chars': ['equal', 'equal', 'greater'],
    'firacode_ligature_name': 'equal_equal_greater.liga',
},
{   # =!=
    'chars': ['equal', 'exclam', 'equal'],
    'firacode_ligature_name': 'equal_exclam_equal.liga',
},
{   # =>
    'chars': ['equal', 'greater'],
```

```
    'firacode_ligature_name': 'equal_greater.liga',
},
{   # =>>
    'chars': ['equal', 'greater', 'greater'],
    'firacode_ligature_name': 'equal_greater_greater.liga',
},
{   # =<<
    'chars': ['equal', 'less', 'less'],
    'firacode_ligature_name': 'equal_less_less.liga',
},
{   # =/=
    'chars': ['equal', 'slash', 'equal'],
    'firacode_ligature_name': 'equal_slash_equal.liga',
},
{   # !=
    'chars': ['exclam', 'equal'],
    'firacode_ligature_name': 'exclam_equal.liga',
},
{   # !==
    'chars': ['exclam', 'equal', 'equal'],
    'firacode_ligature_name': 'exclam_equal_equal.liga',
},
{   # !!
    'chars': ['exclam', 'exclam'],
    'firacode_ligature_name': 'exclam_exclam.liga',
},
{   # !!.
```

```
    'chars': ['exclam', 'exclam', 'period'],
    'firacode_ligature_name': 'exclam_exclam_period.liga',
},
{   # >:
    'chars': ['greater', 'colon'],
    'firacode_ligature_name': 'greater_colon.liga',
},
{   # >=
    'chars': ['greater', 'equal'],
    'firacode_ligature_name': 'greater_equal.liga',
},
{   # >=>
    'chars': ['greater', 'equal', 'greater'],
    'firacode_ligature_name': 'greater_equal_greater.liga',
},
{   # >>
    'chars': ['greater', 'greater'],
    'firacode_ligature_name': 'greater_greater.liga',
},
{   # >>=
    'chars': ['greater', 'greater', 'equal'],
    'firacode_ligature_name': 'greater_greater_equal.liga',
},
{   # >>>
    'chars': ['greater', 'greater', 'greater'],
    'firacode_ligature_name': 'greater_greater_greater.liga',
},
```

```
{   # >>-
    'chars': ['greater', 'greater', 'hyphen'],
    'firacode_ligature_name': 'greater_greater_hyphen.liga',
},
{   # >-
    'chars': ['greater', 'hyphen'],
    'firacode_ligature_name': 'greater_hyphen.liga',
},
{   # >->
    'chars': ['greater', 'hyphen', 'greater'],
    'firacode_ligature_name': 'greater_hyphen_greater.liga',
},
{   # -~
    'chars': ['hyphen', 'asciitilde'],
    'firacode_ligature_name': 'hyphen_asciitilde.liga',
},
{   # -|
    'chars': ['hyphen', 'bar'],
    'firacode_ligature_name': 'hyphen_bar.liga',
},
{   # ->
    'chars': ['hyphen', 'greater'],
    'firacode_ligature_name': 'hyphen_greater.liga',
},
{   # ->>
    'chars': ['hyphen', 'greater', 'greater'],
    'firacode_ligature_name': 'hyphen_greater_greater.liga',
```

```
},
{   # --
    'chars': ['hyphen', 'hyphen'],
    'firacode_ligature_name': 'hyphen_hyphen.liga',
},
{   # -->
    'chars': ['hyphen', 'hyphen', 'greater'],
    'firacode_ligature_name': 'hyphen_hyphen_greater.liga',
},
{   # ---
    'chars': ['hyphen', 'hyphen', 'hyphen'],
    'firacode_ligature_name': 'hyphen_hyphen_hyphen.liga',
},
{   # -<
    'chars': ['hyphen', 'less'],
    'firacode_ligature_name': 'hyphen_less.liga',
},
{   # -<<
    'chars': ['hyphen', 'less', 'less'],
    'firacode_ligature_name': 'hyphen_less_less.liga',
},
{   # <~
    'chars': ['less', 'asciitilde'],
    'firacode_ligature_name': 'less_asciitilde.liga',
},
{   # <~~
    'chars': ['less', 'asciitilde', 'asciitilde'],
```

```
    'firacode_ligature_name': 'less_asciitilde_asciitilde.liga',
},
{   # <~>
    'chars': ['less', 'asciitilde', 'greater'],
    'firacode_ligature_name': 'less_asciitilde_greater.liga',
},
{   # <*
    'chars': ['less', 'asterisk'],
    'firacode_ligature_name': 'less_asterisk.liga',
},
{   # <*>
    'chars': ['less', 'asterisk', 'greater'],
    'firacode_ligature_name': 'less_asterisk_greater.liga',
},
{   # <|
    'chars': ['less', 'bar'],
    'firacode_ligature_name': 'less_bar.liga',
},
{   # <||
    'chars': ['less', 'bar', 'bar'],
    'firacode_ligature_name': 'less_bar_bar.liga',
},
{   # <|||
    'chars': ['less', 'bar', 'bar', 'bar'],
    'firacode_ligature_name': 'less_bar_bar_bar.liga',
},
{   # <|>
```

```
    'chars': ['less', 'bar', 'greater'],
    'firacode_ligature_name': 'less_bar_greater.liga',
},
{   # <:
    'chars': ['less', 'colon'],
    'firacode_ligature_name': 'less_colon.liga',
},
{   # <$
    'chars': ['less', 'dollar'],
    'firacode_ligature_name': 'less_dollar.liga',
},
{   # <$>
    'chars': ['less', 'dollar', 'greater'],
    'firacode_ligature_name': 'less_dollar_greater.liga',
},
{   # <=
    'chars': ['less', 'equal'],
    'firacode_ligature_name': 'less_equal.liga',
},
{   # <=|
    'chars': ['less', 'equal', 'bar'],
    'firacode_ligature_name': 'less_equal_bar.liga',
},
{   # <==
    'chars': ['less', 'equal', 'equal'],
    'firacode_ligature_name': 'less_equal_equal.liga',
},
```

```
{   # <==>
    'chars': ['less', 'equal', 'equal', 'greater'],
    'firacode_ligature_name': 'less_equal_equal_greater.liga',
},
{   # <=>
    'chars': ['less', 'equal', 'greater'],
    'firacode_ligature_name': 'less_equal_greater.liga',
},
{   # <=<
    'chars': ['less', 'equal', 'less'],
    'firacode_ligature_name': 'less_equal_less.liga',
},
{   # <!--
    'chars': ['less', 'exclam', 'hyphen', 'hyphen'],
    'firacode_ligature_name': 'less_exclam_hyphen_hyphen.liga',
},
{   # <>
    'chars': ['less', 'greater'],
    'firacode_ligature_name': 'less_greater.liga',
},
{   # <-
    'chars': ['less', 'hyphen'],
    'firacode_ligature_name': 'less_hyphen.liga',
},
{   # <-|
    'chars': ['less', 'hyphen', 'bar'],
    'firacode_ligature_name': 'less_hyphen_bar.liga',
```

```
},
{   # <->
    'chars': ['less', 'hyphen', 'greater'],
    'firacode_ligature_name': 'less_hyphen_greater.liga',
},
{   # <--
    'chars': ['less', 'hyphen', 'hyphen'],
    'firacode_ligature_name': 'less_hyphen_hyphen.liga',
},
{   # <-<
    'chars': ['less', 'hyphen', 'less'],
    'firacode_ligature_name': 'less_hyphen_less.liga',
},
{   # <<
    'chars': ['less', 'less'],
    'firacode_ligature_name': 'less_less.liga',
},
{   # <<=
    'chars': ['less', 'less', 'equal'],
    'firacode_ligature_name': 'less_less_equal.liga',
},
{   # <<-
    'chars': ['less', 'less', 'hyphen'],
    'firacode_ligature_name': 'less_less_hyphen.liga',
},
{   # <<->>
    'chars': ['less', 'less', 'hyphen', 'greater', 'greater'],
```

```
    'firacode_ligature_name': 'less_less_hyphen_greater_greater.liga',
},
{   # <<<
    'chars': ['less', 'less', 'less'],
    'firacode_ligature_name': 'less_less_less.liga',
},
{   # <+
    'chars': ['less', 'plus'],
    'firacode_ligature_name': 'less_plus.liga',
},
{   # <+>
    'chars': ['less', 'plus', 'greater'],
    'firacode_ligature_name': 'less_plus_greater.liga',
},
{   # </
    'chars': ['less', 'slash'],
    'firacode_ligature_name': 'less_slash.liga',
},
{   # </>
    'chars': ['less', 'slash', 'greater'],
    'firacode_ligature_name': 'less_slash_greater.liga',
},
{   # #{
    'chars': ['numbersign', 'braceleft'],
    'firacode_ligature_name': 'numbersign_braceleft.liga',
},
{   # #[
```

```
    'chars': ['numbersign', 'bracketleft'],
    'firacode_ligature_name': 'numbersign_bracketleft.liga',
},
{   # #:
    'chars': ['numbersign', 'colon'],
    'firacode_ligature_name': 'numbersign_colon.liga',
},
{   # #=
    'chars': ['numbersign', 'equal'],
    'firacode_ligature_name': 'numbersign_equal.liga',
},
{   # #!
    'chars': ['numbersign', 'exclam'],
    'firacode_ligature_name': 'numbersign_exclam.liga',
},
{   # ##
    'chars': ['numbersign', 'numbersign'],
    'firacode_ligature_name': 'numbersign_numbersign.liga',
},
{   # ###
    'chars': ['numbersign', 'numbersign', 'numbersign'],
    'firacode_ligature_name': 'numbersign_numbersign_numbersign.liga',
},
{   # ####
    'chars': ['numbersign', 'numbersign', 'numbersign', 'numbersign'],
    'firacode_ligature_name': 'numbersign_numbersign_numbersign_numbersign.liga',
},
```

```
{   # #(
    'chars': ['numbersign', 'parenleft'],
    'firacode_ligature_name': 'numbersign_parenleft.liga',
},
{   # #?
    'chars': ['numbersign', 'question'],
    'firacode_ligature_name': 'numbersign_question.liga',
},
{   # #_
    'chars': ['numbersign', 'underscore'],
    'firacode_ligature_name': 'numbersign_underscore.liga',
},
{   # #_(
    'chars': ['numbersign', 'underscore', 'parenleft'],
    'firacode_ligature_name': 'numbersign_underscore_parenleft.liga',
},
{   # %%
    'chars': ['percent', 'percent'],
    'firacode_ligature_name': 'percent_percent.liga',
},
{   # .=
    'chars': ['period', 'equal'],
    'firacode_ligature_name': 'period_equal.liga',
},
{   # .-
    'chars': ['period', 'hyphen'],
    'firacode_ligature_name': 'period_hyphen.liga',
```

```
},
{   # ..
    'chars': ['period', 'period'],
    'firacode_ligature_name': 'period_period.liga',
},
{   # ..=
    'chars': ['period', 'period', 'equal'],
    'firacode_ligature_name': 'period_period_equal.liga',
},
{   # ..<
    'chars': ['period', 'period', 'less'],
    'firacode_ligature_name': 'period_period_less.liga',
},
{   # ...
    'chars': ['period', 'period', 'period'],
    'firacode_ligature_name': 'period_period_period.liga',
},
{   # .?
    'chars': ['period', 'question'],
    'firacode_ligature_name': 'period_question.liga',
},
{   # +>
    'chars': ['plus', 'greater'],
    'firacode_ligature_name': 'plus_greater.liga',
},
{   # ++
    'chars': ['plus', 'plus'],
```

```
    'firacode_ligature_name': 'plus_plus.liga',
},
{   # +++
    'chars': ['plus', 'plus', 'plus'],
    'firacode_ligature_name': 'plus_plus_plus.liga',
},
{   # ?:
    'chars': ['question', 'colon'],
    'firacode_ligature_name': 'question_colon.liga',
},
{   # ?=
    'chars': ['question', 'equal'],
    'firacode_ligature_name': 'question_equal.liga',
},
{   # ?.
    'chars': ['question', 'period'],
    'firacode_ligature_name': 'question_period.liga',
},
{   # ??
    'chars': ['question', 'question'],
    'firacode_ligature_name': 'question_question.liga',
},
{   # ;;
    'chars': ['semicolon', 'semicolon'],
    'firacode_ligature_name': 'semicolon_semicolon.liga',
},
{   # /*
```

```
    'chars': ['slash', 'asterisk'],
    'firacode_ligature_name': 'slash_asterisk.liga',
},
{   # /\
    'chars': ['slash', 'backslash'],
    'firacode_ligature_name': 'slash_backslash.liga',
},
{   # /=
    'chars': ['slash', 'equal'],
    'firacode_ligature_name': 'slash_equal.liga',
},
{   # /==
    'chars': ['slash', 'equal', 'equal'],
    'firacode_ligature_name': 'slash_equal_equal.liga',
},
{   # />
    'chars': ['slash', 'greater'],
    'firacode_ligature_name': 'slash_greater.liga',
},
{   # //
    'chars': ['slash', 'slash'],
    'firacode_ligature_name': 'slash_slash.liga',
},
{   # ///
    'chars': ['slash', 'slash', 'slash'],
    'firacode_ligature_name': 'slash_slash_slash.liga',
},
```

```
    {   # _|_
        'chars': ['underscore', 'bar', 'underscore'],
        'firacode_ligature_name': 'underscore_bar_underscore.liga',
    },
    {   # __
        'chars': ['underscore', 'underscore'],
        'firacode_ligature_name': 'underscore_underscore.liga',
    },
    {   # www
        'chars': ['w', 'w', 'w'],
        'firacode_ligature_name': 'w_w_w.liga',
    },
]
```

C:\Users\david/src\Ligaturizer\ligaturize.py

```
#!/usr/bin/env python
#
# usage: fontforge -lang=py ligaturize.py [options]
# Run with --help for detailed options, or use the `build.py` script to
# process lots of fonts at once.
#
# See ligatures.py for a list of all the ligatures (and, optionally, individual
# characters) that will be copied.

import fontforge
import psMat
import os
from os import path
import sys

from ligatures import ligatures
from char_dict import char_dict

# Constants
COPYRIGHT = '''
Programming ligatures added by Ilya Skriblovsky from FiraCode
FiraCode Copyright (c) 2015 by Nikita Prokopov'''

def get_ligature_source(fontname):
    # Become case-insensitive
    fontname = fontname.lower()
```

```
    for weight in ['Bold', 'Retina', 'Medium', 'Regular', 'Light']:
        if fontname.endswith('-' + weight.lower()):
            # Exact match for one of the Fira Code weights
            return 'fonts/fira/distr/otf/FiraCode-%s.otf' % weight

    # No exact match. Guess that we want 'Bold' if the font name has 'bold' or
    # 'heavy' in it, and 'Regular' otherwise.
    if 'bold' in fontname or 'heavy' in fontname:
        return 'fonts/fira/distr/otf/FiraCode-Bold.otf'
    return 'fonts/fira/distr/otf/FiraCode-Regular.otf'

class LigatureCreator(object):

    def __init__(self, font, firacode,
                 scale_character_glyphs_threshold,
                 copy_character_glyphs):
        self.font = font
        self.firacode = firacode
        self.scale_character_glyphs_threshold = scale_character_glyphs_threshold
        self.should_copy_character_glyphs = copy_character_glyphs
        self._lig_counter = 0

        # Scale firacode to correct em height.
        self.firacode.em = self.font.em
        self.emwidth = self.font[ord('m')].width

    def copy_ligature_from_source(self, ligature_name):
```

```
    try:
        self.firacode.selection.none()
        self.firacode.selection.select(ligature_name)
        self.firacode.copy()
        return True
    except ValueError:
        return False


def correct_character_width(self, glyph):
    """Width-correct copied individual characters (not ligatures!).

    This will correct the horizontal advance of characters to match the em
    width of the output font, and (depending on the width of the glyph, the
    em width of the output font, and the value of the command line option
    --scale-character-glyphs-threshold) optionally horizontally scale it.

    Glyphs that are not horizontally scaled, but which still need horizontal
    advance correction, will be centered instead.
    """

    if glyph.width == self.emwidth:
        # No correction needed.
        return

    widthdelta = float(abs(glyph.width - self.emwidth)) / self.emwidth
    if widthdelta >= self.scale_character_glyphs_threshold:
        # Character is too wide/narrow compared to output font; scale it.
```

```
            scale = float(self.emwidth) / glyph.width
            glyph.transform(psMat.scale(scale, 1.0))
        else:
            # Do not scale; just center copied characters in their hbox.
            # Fix horizontal advance first, to recalculate the bearings.
            glyph.width = self.emwidth
            # Correct bearings to center the glyph.
            glyph.left_side_bearing = (glyph.left_side_bearing + glyph.right_side_bearing) / 2
            glyph.right_side_bearing = glyph.left_side_bearing

        # Final adjustment of horizontal advance to correct for rounding
        # errors when scaling/centering -- otherwise small errors can result
        # in visible misalignment near the end of long lines.
        glyph.width = self.emwidth


    def copy_character_glyphs(self, chars):
        """Copy individual (non-ligature) characters from the ligature font."""
        if not self.should_copy_character_glyphs:
            return
        print("    ...copying %d character glyphs..." % (len(chars)))

        for char in chars:
            self.firacode.selection.none()
            self.firacode.selection.select(char)
            self.firacode.copy()
            self.font.selection.none()
```

```
            self.font.selection.select(char)
            self.font.paste()
            self.correct_character_width(self.font[ord(char_dict[char])])

    def correct_ligature_width(self, glyph):
        """Correct the horizontal advance and scale of a ligature."""

        if glyph.width == self.emwidth:
            return

        # TODO: some kind of threshold here, similar to the character glyph
        # scale threshold? The largest ligature uses 0.956 of its hbox, so if
        # the target font is within 4% of the source font size, we don't need to
        # resize -- but we may want to adjust the bearings. And we can't just
        # center it, because ligatures are characterized by very large negative
        # left bearings -- they advance 1em, but draw from (-(n-1))em to +1em.
        scale = float(self.emwidth) / glyph.width
        glyph.transform(psMat.scale(scale, 1.0))
        glyph.width = self.emwidth

    def add_ligature(self, input_chars, firacode_ligature_name):
        if firacode_ligature_name is None:
            # No ligature name -- we're just copying a bunch of individual characters.
            self.copy_character_glyphs(input_chars)
            return

        if not self.copy_ligature_from_source(firacode_ligature_name):
```

```
    # Ligature not in source font.
    return

self._lig_counter += 1
ligature_name = 'lig.{}'.format(self._lig_counter)

self.font.createChar(-1, ligature_name)
self.font.selection.none()
self.font.selection.select(ligature_name)
self.font.paste()
self.correct_ligature_width(self.font[ligature_name])

self.font.selection.none()
self.font.selection.select('space')
self.font.copy()

lookup_name = lambda i: 'lookup.{}.{}'.format(self._lig_counter, i)
lookup_sub_name = lambda i: 'lookup.sub.{}.{}'.format(self._lig_counter, i)
cr_name = lambda i: 'CR.{}.{}'.format(self._lig_counter, i)

for i, char in enumerate(input_chars):
    self.font.addLookup(lookup_name(i), 'gsub_single', (), ())
    self.font.addLookupSubtable(lookup_name(i), lookup_sub_name(i))

    if char not in self.font:
        # We assume here that this is because char is a single letter
        # (e.g. 'w') rather than a character name, and the font we're
```

```
        # editing doesn't have glyphnames for letters.
        self.font[ord(char_dict[char])].glyphname = char

    if i < len(input_chars) - 1:
        self.font.createChar(-1, cr_name(i))
        self.font.selection.none()
        self.font.selection.select(cr_name(i))
        self.font.paste()

        self.font[char].addPosSub(lookup_sub_name(i), cr_name(i))
    else:
        self.font[char].addPosSub(lookup_sub_name(i), ligature_name)

calt_lookup_name = 'calt.{}'.format(self._lig_counter)
self.font.addLookup(calt_lookup_name, 'gsub_contextchain', (),
    (('calt', (('DFLT', ('dflt',)),
                ('arab', ('dflt',)),
                ('armn', ('dflt',)),
                ('cyrl', ('SRB ', 'dflt')),
                ('geor', ('dflt',)),
                ('grek', ('dflt',)),
                ('lao ', ('dflt',)),
                ('latn', ('CAT ', 'ESP ', 'GAL ', 'ISM ', 'KSM ', 'LSM ', 'MOL ', 'NSM ', 'ROM ', 'SKS ', 'SSM ', 'dfl
                ('math', ('dflt',)),
                ('thai', ('dflt',)))),)),
#print('CALT %s (%s)' % (calt_lookup_name, firacode_ligature_name))
for i, char in enumerate(input_chars):
```

```
            self.add_calt(calt_lookup_name, 'calt.{}.{}'.format(self._lig_counter, i),
                '{prev} | {cur} @<{lookup}> | {next}',
                prev = ' '.join(cr_name(j) for j in range(i)),
                cur = char,
                lookup = lookup_name(i),
                next = ' '.join(input_chars[i+1:]))

        # Add ignore rules
        self.add_calt(calt_lookup_name, 'calt.{}.{}'.format(self._lig_counter, i+1),
            '| {first} | {rest} {last}',
            first = input_chars[0],
            rest = ' '.join(input_chars[1:]),
            last = input_chars[-1])
        self.add_calt(calt_lookup_name, 'calt.{}.{}'.format(self._lig_counter, i+2),
            '{first} | {first} | {rest}',
            first = input_chars[0],
            rest = ' '.join(input_chars[1:]))

    def add_calt(self, calt_name, subtable_name, spec, **kwargs):
        spec = spec.format(**kwargs)
        #print('    %s: %s ' % (subtable_name, spec))
        self.font.addContextualSubtable(calt_name, subtable_name, 'glyph', spec)


def replace_sfnt(font, key, value):
    font.sfnt_names = tuple(
        (row[0], key, value)
```

```
        if row[1] == key
        else row
        for row in font.sfnt_names
    )

def update_font_metadata(font, new_name):
    # Figure out the input font's real name (i.e. without a hyphenated suffix)
    # and hyphenated suffix (if present)
    old_name = font.familyname
    try:
        suffix = font.fontname.split('-')[1]
    except IndexError:
        suffix = None

    # Replace the old name with the new name whether or not a suffix was present.
    # If a suffix was present, append it accordingly.
    font.familyname = new_name
    if suffix:
        font.fullname = "%s %s" % (new_name, suffix)
        font.fontname = "%s-%s" % (new_name.replace(' ', ''), suffix)
    else:
        font.fullname = new_name
        font.fontname = new_name.replace(' ', '')

    print("Ligaturizing font %s (%s) as '%s'" % (
        path.basename(font.path), old_name, new_name))
```

```
    font.copyright = (font.copyright or '') + COPYRIGHT
    replace_sfnt(font, 'UniqueID', '%s; Ligaturized' % font.fullname)
    replace_sfnt(font, 'Preferred Family', new_name)
    replace_sfnt(font, 'Compatible Full', new_name)
    replace_sfnt(font, 'Family', new_name)
    replace_sfnt(font, 'WWS Family', new_name)

def ligaturize_font(input_font_file, output_dir, ligature_font_file,
                    output_name, prefix, **kwargs):
    font = fontforge.open(input_font_file)

    if not ligature_font_file:
        ligature_font_file = get_ligature_source(font.fontname)

    if output_name:
        name = output_name
    else:
        name = font.familyname
    if prefix:
        name = "%s %s" % (prefix, name)

    update_font_metadata(font, name)

    print('    ...using ligatures from %s' % ligature_font_file)
    firacode = fontforge.open(ligature_font_file)

    creator = LigatureCreator(font, firacode, **kwargs)
```

```
    ligature_length = lambda lig: len(lig['chars'])
    for lig_spec in sorted(ligatures, key = ligature_length):
        try:
            creator.add_ligature(lig_spec['chars'], lig_spec['firacode_ligature_name'])
        except Exception as e:
            print('Exception while adding ligature: {}'.format(lig_spec))
            raise

    # Work around a bug in Fontforge where the underline height is subtracted from
    # the underline width when you call generate().
    font.upos += font.uwidth

    # Generate font type (TTF or OTF) corresponding to input font extension
    # (defaults to TTF)
    if input_font_file[-4:].lower() == '.otf':
        output_font_type = '.otf'
    else:
        output_font_type = '.ttf'

    # Generate font & move to output directory
    output_font_file = path.join(output_dir, font.fontname + output_font_type)
    print("    ...saving to '%s' (%s)" % (output_font_file, font.fullname))
    font.generate(output_font_file)


def parse_args():
    from argparse import ArgumentParser
```

```
parser = ArgumentParser()
parser.add_argument("input_font_file",
    help="The TTF or OTF font to add ligatures to.")
parser.add_argument("--output-dir",
    help="The directory to save the ligaturized font in. The actual filename"
        " will be automatically generated based on the input font name and"
        " the --prefix and --output-name flags.")
parser.add_argument("--ligature-font-file",
    type=str, default='', metavar='PATH',
    help="The file to copy ligatures from. If unspecified, ligaturize will"
        " attempt to pick a suitable one from fonts/fira/distr/otf/ based on the input"
        " font's weight.")
parser.add_argument("--copy-character-glyphs",
    default=False, action='store_true',
    help="Copy glyphs for (some) individual characters from the ligature"
        " font as well. This will result in punctuation that matches the"
        " ligatures more closely, but may not fit in as well with the rest"
        " of the font.")
parser.add_argument("--scale-character-glyphs-threshold",
    type=float, default=0.1, metavar='THRESHOLD',
    help="When copying character glyphs, if they differ in width from the"
        " width of the input font by at least this much, scale them"
        " horizontally to match the input font even if this noticeably"
        " changes their aspect ratio. The default (0.1) means to scale if"
        " they are at least 10%% wider or narrower. A value of 0 will scale"
        " all copied character glyphs; a value of 2 effectively disables"
        " character glyph scaling.")
```

```
    parser.add_argument("--prefix",
        type=str, default="Liga",
        help="String to prefix the name of the generated font with.")
    parser.add_argument("--output-name",
        type=str, default="",
        help="Name of the generated font. Completely replaces the original.")
    return parser.parse_args()


def main():
    ligaturize_font(**vars(parse_args()))


if __name__ == '__main__':
    main()
```

C:\Users\david/src\Ligaturizer\Makefile

```
# To build with different settings (e.g. turn on character glyph copying),
# edit build.py and then "make".

default: without-characters

all: without-characters with-characters

clean:
 rm -rf fonts/output/* fonts/output-with-characters/* Ligaturized*.zip

release: clean all pack

pack:
 zip -r -9 -j LigaturizedFonts.zip fonts/output/
 zip -r -9 -j LigaturizedFontsWithCharacters.zip fonts/output-with-characters/

without-characters:
 fontforge -lang=py -script build.py 2>&1 \
 | grep -Fv 'This contextual rule applies no lookups.' \
 | grep -Fv 'Bad device table'

with-characters:
 fontforge -lang=py -script build.py --copy-character-glyphs 2>&1 \
 | grep -Fv 'This contextual rule applies no lookups.' \
 | grep -Fv 'Bad device table'
```

```
ligature-list:
 luajit name2dict.lua < fonts/fira/FiraCode.glyphs


testpattern:
 grep -F "{   #" ligatures.py \
  | grep -v absent \
  | cut -d'#' -f2 \
  | tr -d ' ' \
  | egrep '.' \
  | sed -E 's,\\,\\\\,g' \
  | xargs printf '| %6s %6s %6s %6s %6s %6s %6s %6s |\n'


.PHONY: testpattern
```

C:\Users\david/src\Ligaturizer\README.md

Ligaturizer

Add ligatures to any coding font!

This script copies the ligatures (glyphs and rendering information) from [Fira Code](https://github.com/tonsl

This repo contains a [Fontforge python script](ligaturize.py) that you can use to add the Fira Code ligatures

Pre-ligaturized versions are available under [releases](https://github.com/ToxicFrog/Ligaturizer/releases).

Here's a couple examples of the fonts generated: SF Mono & Menlo with ligatures (note the `!=` and `->`):

Requirements

This Repo: You'll need the repo and its submodules, so `git clone` with `--recurse-submodules`.

Using the Fonts: See the [FiraCode README](https://github.com/tonsky/FiraCode) for a list of supporte

Script: This script requires FontForge python bindings. For Debian/Ubuntu they are available in `python

Using the Script

Automatic

Use automatic mode to easily convert 1 or more font(s).

1. Put the font(s) you want into `fonts/`.
1. Edit `ligatures.py` to disable any ligatures you don't want, and/or enable any (non-ligature) characters y
1. Edit `build.py` to add your new font(s) to the `prefixed_fonts` list. It supports globbing, so if (e.g.) you wa
1. Run `make`.
1. Retrieve the ligaturized fonts from `fonts/output/`.
1. The output fonts will be renamed with the prefix "Liga".

Manual

1. Move/copy the font you want to ligaturize into `fonts/` (or somewhere else convenient).
1. Edit `ligatures.py` to disable any ligatures you don't want.
1. Run the script:

```
$ fontforge -lang py -script ligaturize.py path/to/input/font.ttf
    --output-dir=path/to/output/dir/ \
    --output-name='Name of Ligaturized Font'
```

e.g.

```
$ fontforge -lang py -script ligaturize.py fonts/Cousine-Regular.ttf
    --output-dir='fonts/output/' \
    --output-name='Ligaturized Cousine'
```

Which will produce `fonts/output/LigaturizedCousine-Regular.ttf`.

The font weight will be inherited from the original file; the font name will be replaced with whatever you spe

`ligatures.py` supports some additional command line options to (e.g.) change which font ligatures are copi

Misc.

Credit

This script was originally written by [IlyaSkriblovsky](https://github.com/IlyaSkriblovsky) for adding ligatures

Contributions

Contributions always welcome! Please submit a Pull Request, or create an Issue if you have an idea for a f

Related Projects

For more awesome programming fonts with ligatures, check out:

1. [FiraCode](https://github.com/tonsky/FiraCode)
2. [Hasklig](https://github.com/i-tu/Hasklig)

C:\Users\david/src\Ligaturizer\stat.py

```
#!/usr/bin/env python
#
# usage: fontforge -lang=py ligaturize.py <input file> <output file> [ligature file]
#
# It will copy input to output, updating the embedded font name and splicing
# in the ligatures from FiraCode-Medium.otf (which must be in $PWD). If the
# ligature file is not specified, it will try to guess an appropriate Fira Code
# OTF based on the name of the output file.
#
# See ligatures.py for a list of all the ligatures that will be copied.

import fontforge
import os
from os import path
import sys

input_font_path = sys.argv[1]
font = fontforge.open(input_font_path)

for field in dir(font):
  if field.startswith('__'):
    continue
  value = getattr(font, field)
  if 'built-in' in str(value):
    continue
  print("%s = %s" % (field, value))
```

```
#print "emwidth = %d" % (font['m'].width)

print("emwidth = %d" % (font[ord('m')].width))

print("aspect = %f" % (float(font[ord('m')].width)/font.em))


#print "font: %d %d %d " % (font.upos, font.capHeight, font.xHeight)
```

C:\Users\david/src\Ligaturizer\testpattern

~!@#$%^&*()_+ `1234567890-=

QWERTYUIOP{}| qwertyuiop[]\

ASDFGHJKL:" asdfghjkl;'

ZXCVBNM< >? zxcvbnm,./

regenerate with 'make testpattern >> testpattern'

| { { { { { { && ^= |

| ~~ ~~> ~@ ~= ~> ~- ** *** |

| *> */ / || |||> ||= ||> ||- |

| |} |] |= |=> |> |- |-> {| |

| [|] :: ::: ::= := :> :< |

| $> =:= == === ==> =!= => =>> |

| =<< =/= != !== !! !!. >: >= |

| >=> >> >>= >>> >>- >- >-> -~ |

| -| -> ->> -- --> --- -< -<< |

| <~ <~~ <~> <* <*> <| <|| <||| |

| <|> <: <$ <$> <= <=| <== <==> |

| <=> <=< <!-- <> <- <-| <-> <-- |

| <-< << <<= <<- <<->> <<< <+ <+> |

| </ </> %% .= .- .. ..= ..< |

| ... .? +> ++ +++ ?: ?= ?. |

| ?? ;; /* //= /== /> // /// |

| _|_ __ www |

C:\Users\david/src\Ligaturizer\.git\config

```
[core]
 repositoryformatversion = 0
 filemode = false
 bare = false
 logallrefupdates = true
 symlinks = false
 ignorecase = true
[remote "origin"]
 url = git@github.com:ToxicFrog/Ligaturizer.git
 fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
 remote = origin
 merge = refs/heads/master
```

C:\Users\david/src\Ligaturizer\.git\description

Unnamed repository; edit this file 'description' to name the repository.

C:\Users\david/src\Ligaturizer\.git\FETCH_HEAD

c4065187a544a8fab40826fc91db1c6180a2d342 branch 'master' of github.com:ToxicFrog/Ligaturizer

C:\Users\david/src\Ligaturizer\.git\HEAD

ref: refs/heads/master

C:\Users\david/src\Ligaturizer\.git\index

C:\Users\david/src\Ligaturizer\.git\packed-refs

pack-refs with: peeled fully-peeled sorted

c4065187a544a8fab40826fc91db1c6180a2d342 refs/remotes/origin/master

01b7c2bb014acfe733100596cc2bcf8ff64fafa5 refs/tags/v1

5389c7c2b33f515b9de6a220894c0203662d86f6 refs/tags/v2

ff89b86078871c24b4c48a7420ed87580aae51df refs/tags/v3

6d3070ae1c6b1a6ec196db486be813d713c7ce9c refs/tags/v4

c4065187a544a8fab40826fc91db1c6180a2d342 refs/tags/v5

C:\Users\david/src\Ligaturizer\.git\hooks\applypatch-msg.sample

```
#!/bin/sh
#
# An example hook script to check the commit log message taken by
# applypatch from an e-mail message.
#
# The hook should exit with non-zero status after issuing an
# appropriate message if it wants to stop the commit.  The hook is
# allowed to edit the commit message file.
#
# To enable this hook, rename this file to "applypatch-msg".

. git-sh-setup
commitmsg="$(git rev-parse --git-path hooks/commit-msg)"
test -x "$commitmsg" && exec "$commitmsg" ${1+"$@"}
:
```

C:\Users\david/src\Ligaturizer\.git\hooks\commit-msg.sample

```
#!/bin/sh
#
# An example hook script to check the commit log message.
# Called by "git commit" with one argument, the name of the file
# that has the commit message.  The hook should exit with non-zero
# status after issuing an appropriate message if it wants to stop the
# commit.  The hook is allowed to edit the commit message file.
#
# To enable this hook, rename this file to "commit-msg".


# Uncomment the below to add a Signed-off-by line to the message.
# Doing this in a hook is a bad idea in general, but the prepare-commit-msg
# hook is more suited to it.
#
# SOB=$(git var GIT_AUTHOR_IDENT | sed -n 's/^\(.*>\).*$/Signed-off-by: \1/p')
# grep -qs "^$SOB" "$1" || echo "$SOB" >> "$1"


# This example catches duplicate Signed-off-by lines.


test "" = "$(grep '^Signed-off-by: ' "$1" |
  sort | uniq -c | sed -e '/^[  ]*1[  ]/d')" || {
 echo >&2 Duplicate Signed-off-by lines.
 exit 1
}
```

C:\Users\david/src\Ligaturizer\.git\hooks\fsmonitor-watchman.sample

```
#!/usr/bin/perl

use strict;
use warnings;
use IPC::Open2;

# An example hook script to integrate Watchman
# (https://facebook.github.io/watchman/) with git to speed up detecting
# new and modified files.
#
# The hook is passed a version (currently 2) and last update token
# formatted as a string and outputs to stdout a new update token and
# all files that have been modified since the update token. Paths must
# be relative to the root of the working tree and separated by a single NUL.
#
# To enable this hook, rename this file to "query-watchman" and set
# 'git config core.fsmonitor .git/hooks/query-watchman'
#
my ($version, $last_update_token) = @ARGV;

# Uncomment for debugging
# print STDERR "$0 $version $last_update_token\n";

# Check the hook interface version
if ($version ne 2) {
 die "Unsupported query-fsmonitor hook version '$version'.\n" .
```

```
    "Falling back to scanning...\n";
}

my $git_work_tree = get_working_dir();

my $retry = 1;

my $json_pkg;
eval {
 require JSON::XS;
 $json_pkg = "JSON::XS";
 1;
} or do {
 require JSON::PP;
 $json_pkg = "JSON::PP";
};

launch_watchman();

sub launch_watchman {
 my $o = watchman_query();
 if (is_work_tree_watched($o)) {
  output_result($o->{clock}, @{$o->{files}});
 }
}

sub output_result {
```

```
 my ($clockid, @files) = @_;

 # Uncomment for debugging watchman output
 # open (my $fh, ">", ".git/watchman-output.out");
 # binmode $fh, ":utf8";
 # print $fh "$clockid\n@files\n";
 # close $fh;

 binmode STDOUT, ":utf8";
 print $clockid;
 print "\0";
 local $, = "\0";
 print @files;
}

sub watchman_clock {
 my $response = qx/watchman clock "$git_work_tree"/;
 die "Failed to get clock id on '$git_work_tree'.\n" .
  "Falling back to scanning...\n" if $? != 0;

 return $json_pkg->new->utf8->decode($response);
}

sub watchman_query {
 my $pid = open2(\*CHLD_OUT, \*CHLD_IN, 'watchman -j --no-pretty')
 or die "open2() failed: $!\n" .
 "Falling back to scanning...\n";
```

```
# In the query expression below we're asking for names of files that
# changed since $last_update_token but not from the .git folder.
#
# To accomplish this, we're using the "since" generator to use the
# recency index to select candidate nodes and "fields" to limit the
# output to file names only. Then we're using the "expression" term to
# further constrain the results.
my $last_update_line = "";
if (substr($last_update_token, 0, 1) eq "c") {
 $last_update_token = "\"$last_update_token\"";
 $last_update_line = qq[\n"since": $last_update_token,];
}
my $query = <<" END";
 ["query", "$git_work_tree", {$last_update_line
  "fields": ["name"],
  "expression": ["not", ["dirname", ".git"]]
 }]
END

# Uncomment for debugging the watchman query
# open (my $fh, ">", ".git/watchman-query.json");
# print $fh $query;
# close $fh;

print CHLD_IN $query;
close CHLD_IN;
```

```
 my $response = do {local $/; <CHLD_OUT>};

 # Uncomment for debugging the watch response
 # open ($fh, ">", ".git/watchman-response.json");
 # print $fh $response;
 # close $fh;

 die "Watchman: command returned no output.\n" .
 "Falling back to scanning...\n" if $response eq "";
 die "Watchman: command returned invalid output: $response\n" .
 "Falling back to scanning...\n" unless $response =~ /^\{/;

 return $json_pkg->new->utf8->decode($response);
}

sub is_work_tree_watched {
 my ($output) = @_;
 my $error = $output->{error};
 if ($retry > 0 and $error and $error =~ m/unable to resolve root .* directory (.*) is not watched/) {
  $retry--;
  my $response = qx/watchman watch "$git_work_tree"/;
  die "Failed to make watchman watch '$git_work_tree'.\n" .
      "Falling back to scanning...\n" if $? != 0;
  $output = $json_pkg->new->utf8->decode($response);
  $error = $output->{error};
  die "Watchman: $error.\n" .
  "Falling back to scanning...\n" if $error;
```

```
 # Uncomment for debugging watchman output
 # open (my $fh, ">", ".git/watchman-output.out");
 # close $fh;

 # Watchman will always return all files on the first query so
 # return the fast "everything is dirty" flag to git and do the
 # Watchman query just to get it over with now so we won't pay
 # the cost in git to look up each individual file.
 my $o = watchman_clock();
 $error = $output->{error};

 die "Watchman: $error.\n" .
 "Falling back to scanning...\n" if $error;

 output_result($o->{clock}, ("/"));
 $last_update_token = $o->{clock};

 eval { launch_watchman() };
 return 0;
}

die "Watchman: $error.\n" .
"Falling back to scanning...\n" if $error;

return 1;
}
```

```
sub get_working_dir {
 my $working_dir;
 if ($^O =~ 'msys' || $^O =~ 'cygwin') {
  $working_dir = Win32::GetCwd();
  $working_dir =~ tr/\\/\/;
 } else {
  require Cwd;
  $working_dir = Cwd::cwd();
 }

 return $working_dir;
}
```

C:\Users\david/src\Ligaturizer\.git\hooks\post-update.sample

```
#!/bin/sh
#
# An example hook script to prepare a packed repository for use over
# dumb transports.
#
# To enable this hook, rename this file to "post-update".

exec git update-server-info
```

C:\Users\david/src\Ligaturizer\.git\hooks\pre-applypatch.sample

```
#!/bin/sh
#
# An example hook script to verify what is about to be committed
# by applypatch from an e-mail message.
#
# The hook should exit with non-zero status after issuing an
# appropriate message if it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-applypatch".

. git-sh-setup
precommit="$(git rev-parse --git-path hooks/pre-commit)"
test -x "$precommit" && exec "$precommit" ${1+"$@"}
:
```

C:\Users\david/src\Ligaturizer\.git\hooks\pre-commit.sample

```
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git commit" with no arguments.  The hook should
# exit with non-zero status after issuing an appropriate message if
# it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-commit".

if git rev-parse --verify HEAD >/dev/null 2>&1
then
 against=HEAD
else
 # Initial commit: diff against an empty tree object
 against=$(git hash-object -t tree /dev/null)
fi

# If you want to allow non-ASCII filenames set this variable to true.
allownonascii=$(git config --type=bool hooks.allownonascii)

# Redirect output to stderr.
exec 1>&2

# Cross platform projects tend to avoid non-ASCII filenames; prevent
# them from being added to the repository. We exploit the fact that the
# printable range starts at the space character and ends with tilde.
```

```
if [ "$allownonascii" != "true" ] &&
 # Note that the use of brackets around a tr range is ok here, (it's
 # even required, for portability to Solaris 10's /usr/bin/tr), since
 # the square bracket bytes happen to fall in the designated range.
 test $(git diff --cached --name-only --diff-filter=A -z $against |
   LC_ALL=C tr -d '[ -~]\0' | wc -c) != 0
then
 cat <<\EOF
Error: Attempt to add a non-ASCII file name.

This can cause problems if you want to work with people on other platforms.

To be portable it is advisable to rename the file.

If you know what you are doing you can disable this check using:

  git config hooks.allownonascii true
EOF
 exit 1
fi

# If there are whitespace errors, print the offending file names and fail.
exec git diff-index --check --cached $against --
```

C:\Users\david/src\Ligaturizer\.git\hooks\pre-merge-commit.sample

```
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git merge" with no arguments.  The hook should
# exit with non-zero status after issuing an appropriate message to
# stderr if it wants to stop the merge commit.
#
# To enable this hook, rename this file to "pre-merge-commit".

. git-sh-setup
test -x "$GIT_DIR/hooks/pre-commit" &&
        exec "$GIT_DIR/hooks/pre-commit"
:
```

C:\Users\david/src\Ligaturizer\.git\hooks\pre-push.sample

```
#!/bin/sh

# An example hook script to verify what is about to be pushed.  Called by "git
# push" after it has checked the remote status, but before anything has been
# pushed.  If this script exits with a non-zero status nothing will be pushed.
#
# This hook is called with the following parameters:
#
# $1 -- Name of the remote to which the push is being done
# $2 -- URL to which the push is being done
#
# If pushing without using a named remote those arguments will be equal.
#
# Information about the commits which are being pushed is supplied as lines to
# the standard input in the form:
#
#   <local ref> <local oid> <remote ref> <remote oid>
#
# This sample shows how to prevent push of commits where the log message starts
# with "WIP" (work in progress).

remote="$1"
url="$2"

zero=$(git hash-object --stdin </dev/null | tr '[0-9a-f]' '0')
```

```
while read local_ref local_oid remote_ref remote_oid
do
 if test "$local_oid" = "$zero"
 then
  # Handle delete
  :
 else
  if test "$remote_oid" = "$zero"
  then
   # New branch, examine all commits
   range="$local_oid"
  else
   # Update to existing branch, examine new commits
   range="$remote_oid..$local_oid"
  fi

  # Check for WIP commit
  commit=$(git rev-list -n 1 --grep '^WIP' "$range")
  if test -n "$commit"
  then
   echo >&2 "Found WIP commit in $local_ref, not pushing"
   exit 1
  fi
 fi
done

exit 0
```

C:\Users\david/src\Ligaturizer\.git\hooks\pre-rebase.sample

```
#!/bin/sh
#
# Copyright (c) 2006, 2008 Junio C Hamano
#
# The "pre-rebase" hook is run just before "git rebase" starts doing
# its job, and can prevent the command from running by exiting with
# non-zero status.
#
# The hook is called with the following parameters:
#
# $1 -- the upstream the series was forked from.
# $2 -- the branch being rebased (or empty when rebasing the current branch).
#
# This sample shows how to prevent topic branches that are already
# merged to 'next' branch from getting rebased, because allowing it
# would result in rebasing already published history.

publish=next
basebranch="$1"
if test "$#" = 2
then
 topic="refs/heads/$2"
else
 topic=`git symbolic-ref HEAD` ||
 exit 0 ;# we do not interrupt rebasing detached HEAD
fi
```

```
case "$topic" in
refs/heads/??/*)
 ;;
*)
 exit 0 ;# we do not interrupt others.
 ;;
esac

# Now we are dealing with a topic branch being rebased
# on top of master.  Is it OK to rebase it?

# Does the topic really exist?
git show-ref -q "$topic" || {
 echo >&2 "No such branch $topic"
 exit 1
}

# Is topic fully merged to master?
not_in_master=`git rev-list --pretty=oneline ^master "$topic"`
if test -z "$not_in_master"
then
 echo >&2 "$topic is fully merged to master; better remove it."
 exit 1 ;# we could allow it, but there is no point.
fi

# Is topic ever merged to next?  If so you should not be rebasing it.
```

```
only_next_1=`git rev-list ^master "^$topic" ${publish} | sort`
only_next_2=`git rev-list ^master           ${publish} | sort`
if test "$only_next_1" = "$only_next_2"
then
 not_in_topic=`git rev-list "^$topic" master`
 if test -z "$not_in_topic"
 then
  echo >&2 "$topic is already up to date with master"
  exit 1 ;# we could allow it, but there is no point.
 else
  exit 0
 fi
else
 not_in_next=`git rev-list --pretty=oneline ^${publish} "$topic"`
 /usr/bin/perl -e '
  my $topic = $ARGV[0];
  my $msg = "* $topic has commits already merged to public branch:\n";
  my (%not_in_next) = map {
   /^([0-9a-f]+) /;
   ($1 => 1);
  } split(/\n/, $ARGV[1]);
  for my $elem (map {
    /^([0-9a-f]+) (.*)$/;
    [$1 => $2];
   } split(/\n/, $ARGV[2])) {
   if (!exists $not_in_next{$elem->[0]}) {
    if ($msg) {
```

```
     print STDERR $msg;
     undef $msg;
    }
    print STDERR " $elem->[1]\n";
   }
  }
 ' "$topic" "$not_in_next" "$not_in_master"
 exit 1
fi

<<\DOC_END

This sample hook safeguards topic branches that have been
published from being rewound.

The workflow assumed here is:

 * Once a topic branch forks from "master", "master" is never
   merged into it again (either directly or indirectly).

 * Once a topic branch is fully cooked and merged into "master",
   it is deleted.  If you need to build on top of it to correct
   earlier mistakes, a new topic branch is created by forking at
   the tip of the "master".  This is not strictly necessary, but
   it makes it easier to keep your history simple.

 * Whenever you need to test or publish your changes to topic
```

branches, merge them into "next" branch.

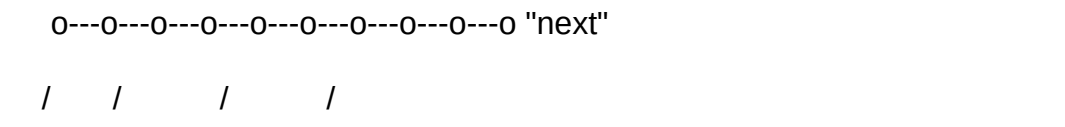
The script, being an example, hardcodes the publish branch name
to be "next", but it is trivial to make it configurable via
$GIT_DIR/config mechanism.

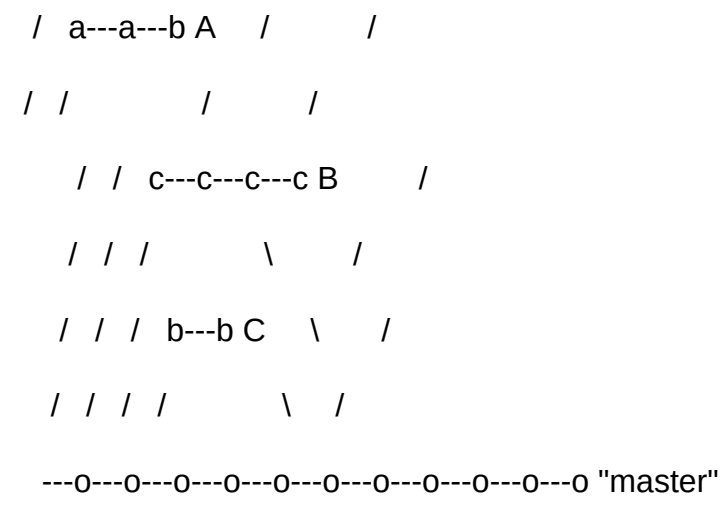
With this workflow, you would want to know:

(1) ... if a topic branch has ever been merged to "next". Young
topic branches can have stupid mistakes you would rather
clean up before publishing, and things that have not been
merged into other branches can be easily rebased without
affecting other people. But once it is published, you would
not want to rewind it.

(2) ... if a topic branch has been fully merged to "master".
Then you can delete it. More importantly, you should not
build on top of it -- other people may already want to
change things related to the topic as patches against your
"master", so if you need further changes, it is better to
fork the topic (perhaps with the same name) afresh from the
tip of "master".

Let's look at this example:





A, B and C are topic branches.

* A has one fix since it was merged up to "next".

* B has finished. It has been fully merged up to "master" and "next",
  and is ready to be deleted.

* C has not merged to "next" at all.

We would want to allow C to be rebased, refuse A, and encourage
B to be deleted.

To compute (1):

```
git rev-list ^master ^topic next
git rev-list ^master        next
```

if these match, topic has not merged in next at all.

To compute (2):

git rev-list master..topic

if this is empty, it is fully merged to "master".

DOC_END

C:\Users\david/src\Ligaturizer\.git\hooks\pre-receive.sample

```
#!/bin/sh
#
# An example hook script to make use of push options.
# The example simply echoes all push options that start with 'echoback='
# and rejects all pushes when the "reject" push option is used.
#
# To enable this hook, rename this file to "pre-receive".

if test -n "$GIT_PUSH_OPTION_COUNT"
then
 i=0
 while test "$i" -lt "$GIT_PUSH_OPTION_COUNT"
 do
  eval "value=\$GIT_PUSH_OPTION_$i"
  case "$value" in
  echoback=*)
   echo "echo from the pre-receive-hook: ${value#*=}" >&2
   ;;
  reject)
   exit 1
  esac
  i=$((i + 1))
 done
fi
```

C:\Users\david/src\Ligaturizer\.git\hooks\prepare-commit-msg.sample

```
#!/bin/sh
#
# An example hook script to prepare the commit log message.
# Called by "git commit" with the name of the file that has the
# commit message, followed by the description of the commit
# message's source.  The hook's purpose is to edit the commit
# message file.  If the hook fails with a non-zero status,
# the commit is aborted.
#
# To enable this hook, rename this file to "prepare-commit-msg".

# This hook includes three examples. The first one removes the
# "# Please enter the commit message..." help message.
#
# The second includes the output of "git diff --name-status -r"
# into the message, just before the "git status" output.  It is
# commented because it doesn't cope with --amend or with squashed
# commits.
#
# The third example adds a Signed-off-by line to the message, that can
# still be edited.  This is rarely a good idea.

COMMIT_MSG_FILE=$1
COMMIT_SOURCE=$2
SHA1=$3
```

```
/usr/bin/perl -i.bak -ne 'print unless(m/^. Please enter the commit message/..m/^#$/)' "$COMMIT_MSG_FIL


# case "$COMMIT_SOURCE,$SHA1" in
#  ,|template,)
#    /usr/bin/perl -i.bak -pe '
#       print "\n" . `git diff --cached --name-status -r`
#   if /^#/ && $first++ == 0' "$COMMIT_MSG_FILE" ;;
#  *) ;;
# esac


# SOB=$(git var GIT_COMMITTER_IDENT | sed -n 's/^\(.*>\).*$/Signed-off-by: \1/p')
# git interpret-trailers --in-place --trailer "$SOB" "$COMMIT_MSG_FILE"
# if test -z "$COMMIT_SOURCE"
# then
#   /usr/bin/perl -i.bak -pe 'print "\n" if !$first_line++' "$COMMIT_MSG_FILE"
# fi
```

C:\Users\david/src\Ligaturizer\.git\hooks\push-to-checkout.sample

```
#!/bin/sh

# An example hook script to update a checked-out tree on a git push.
#
# This hook is invoked by git-receive-pack(1) when it reacts to git
# push and updates reference(s) in its repository, and when the push
# tries to update the branch that is currently checked out and the
# receive.denyCurrentBranch configuration variable is set to
# updateInstead.
#
# By default, such a push is refused if the working tree and the index
# of the remote repository has any difference from the currently
# checked out commit; when both the working tree and the index match
# the current commit, they are updated to match the newly pushed tip
# of the branch. This hook is to be used to override the default
# behaviour; however the code below reimplements the default behaviour
# as a starting point for convenient modification.
#
# The hook receives the commit with which the tip of the current
# branch is going to be updated:
commit=$1

# It can exit with a non-zero status to refuse the push (when it does
# so, it must not modify the index or the working tree).
die () {
 echo >&2 "$*"
```

```
 exit 1
}

# Or it can make any necessary changes to the working tree and to the
# index to bring them to the desired state when the tip of the current
# branch is updated to the new commit, and exit with a zero status.
#
# For example, the hook can simply run git read-tree -u -m HEAD "$1"
# in order to emulate git fetch that is run in the reverse direction
# with git push, as the two-tree form of git read-tree -u -m is
# essentially the same as git switch or git checkout that switches
# branches while keeping the local changes in the working tree that do
# not interfere with the difference between the branches.

# The below is a more-or-less exact translation to shell of the C code
# for the default behaviour for git's push-to-checkout hook defined in
# the push_to_deploy() function in builtin/receive-pack.c.
#
# Note that the hook will be executed from the repository directory,
# not from the working tree, so if you want to perform operations on
# the working tree, you will have to adapt your code accordingly, e.g.
# by adding "cd .." or using relative paths.

if ! git update-index -q --ignore-submodules --refresh
then
 die "Up-to-date check failed"
fi
```

```
if ! git diff-files --quiet --ignore-submodules --
then
 die "Working directory has unstaged changes"
fi

# This is a rough translation of:
#
#   head_has_history() ? "HEAD" : EMPTY_TREE_SHA1_HEX
if git cat-file -e HEAD 2>/dev/null
then
 head=HEAD
else
 head=$(git hash-object -t tree --stdin </dev/null)
fi

if ! git diff-index --quiet --cached --ignore-submodules $head --
then
 die "Working directory has staged changes"
fi

if ! git read-tree -u -m "$commit"
then
 die "Could not update working tree to new HEAD"
fi
```

C:\Users\david/src\Ligaturizer\.git\hooks\sendemail-validate.sample

```
#!/bin/sh

# An example hook script to validate a patch (and/or patch series) before
# sending it via email.
#
# The hook should exit with non-zero status after issuing an appropriate
# message if it wants to prevent the email(s) from being sent.
#
# To enable this hook, rename this file to "sendemail-validate".
#
# By default, it will only check that the patch(es) can be applied on top of
# the default upstream branch without conflicts in a secondary worktree. After
# validation (successful or not) of the last patch of a series, the worktree
# will be deleted.
#
# The following config variables can be set to change the default remote and
# remote ref that are used to apply the patches against:
#
#   sendemail.validateRemote (default: origin)
#   sendemail.validateRemoteRef (default: HEAD)
#
# Replace the TODO placeholders with appropriate checks according to your
# needs.

validate_cover_letter () {
 file="$1"
```

```
 # TODO: Replace with appropriate checks (e.g. spell checking).
 true
}

validate_patch () {
 file="$1"
 # Ensure that the patch applies without conflicts.
 git am -3 "$file" || return
 # TODO: Replace with appropriate checks for this patch
 # (e.g. checkpatch.pl).
 true
}

validate_series () {
 # TODO: Replace with appropriate checks for the whole series
 # (e.g. quick build, coding style checks, etc.).
 true
}

# main -----------------------------------------------------------------

if test "$GIT_SENDEMAIL_FILE_COUNTER" = 1
then
 remote=$(git config --default origin --get sendemail.validateRemote) &&
 ref=$(git config --default HEAD --get sendemail.validateRemoteRef) &&
 worktree=$(mktemp --tmpdir -d sendemail-validate.XXXXXXX) &&
 git worktree add -fd --checkout "$worktree" "refs/remotes/$remote/$ref" &&
```

```
 git config --replace-all sendemail.validateWorktree "$worktree"
else
 worktree=$(git config --get sendemail.validateWorktree)
fi || {
 echo "sendemail-validate: error: failed to prepare worktree" >&2
 exit 1
}

unset GIT_DIR GIT_WORK_TREE
cd "$worktree" &&

if grep -q "^diff --git " "$1"
then
 validate_patch "$1"
else
 validate_cover_letter "$1"
fi &&

if test "$GIT_SENDEMAIL_FILE_COUNTER" = "$GIT_SENDEMAIL_FILE_TOTAL"
then
 git config --unset-all sendemail.validateWorktree &&
 trap 'git worktree remove -ff "$worktree"' EXIT &&
 validate_series
fi
```

C:\Users\david/src\Ligaturizer\.git\hooks\update.sample

```
#!/bin/sh
#
# An example hook script to block unannotated tags from entering.
# Called by "git receive-pack" with arguments: refname sha1-old sha1-new
#
# To enable this hook, rename this file to "update".
#
# Config
# ------
# hooks.allowunannotated
#   This boolean sets whether unannotated tags will be allowed into the
#   repository.  By default they won't be.
# hooks.allowdeletetag
#   This boolean sets whether deleting tags will be allowed in the
#   repository.  By default they won't be.
# hooks.allowmodifytag
#   This boolean sets whether a tag may be modified after creation. By default
#   it won't be.
# hooks.allowdeletebranch
#   This boolean sets whether deleting branches will be allowed in the
#   repository.  By default they won't be.
# hooks.denycreatebranch
#   This boolean sets whether remotely creating branches will be denied
#   in the repository.  By default this is allowed.
#
```

```
# --- Command line
refname="$1"
oldrev="$2"
newrev="$3"

# --- Safety check
if [ -z "$GIT_DIR" ]; then
 echo "Don't run this script from the command line." >&2
 echo " (if you want, you could supply GIT_DIR then run" >&2
 echo "  $0 <ref> <oldrev> <newrev>)" >&2
 exit 1
fi

if [ -z "$refname" -o -z "$oldrev" -o -z "$newrev" ]; then
 echo "usage: $0 <ref> <oldrev> <newrev>" >&2
 exit 1
fi

# --- Config
allowunannotated=$(git config --type=bool hooks.allowunannotated)
allowdeletebranch=$(git config --type=bool hooks.allowdeletebranch)
denycreatebranch=$(git config --type=bool hooks.denycreatebranch)
allowdeletetag=$(git config --type=bool hooks.allowdeletetag)
allowmodifytag=$(git config --type=bool hooks.allowmodifytag)

# check for no description
projectdesc=$(sed -e '1q' "$GIT_DIR/description")
```

```
case "$projectdesc" in
"Unnamed repository"* | "")
 echo "*** Project description file hasn't been set" >&2
 exit 1
 ;;
esac

# --- Check types
# if $newrev is 0000...0000, it's a commit to delete a ref.
zero=$(git hash-object --stdin </dev/null | tr '[0-9a-f]' '0')
if [ "$newrev" = "$zero" ]; then
 newrev_type=delete
else
 newrev_type=$(git cat-file -t $newrev)
fi

case "$refname","$newrev_type" in
 refs/tags/*,commit)
  # un-annotated tag
  short_refname=${refname##refs/tags/}
  if [ "$allowunannotated" != "true" ]; then
   echo "*** The un-annotated tag, $short_refname, is not allowed in this repository" >&2
   echo "*** Use 'git tag [ -a | -s ]' for tags you want to propagate." >&2
   exit 1
  fi
  ;;
 refs/tags/*,delete)
```

```
 # delete tag
 if [ "$allowdeletetag" != "true" ]; then
  echo "*** Deleting a tag is not allowed in this repository" >&2
  exit 1
 fi
 ;;
refs/tags/*,tag)
 # annotated tag
 if [ "$allowmodifytag" != "true" ] && git rev-parse $refname > /dev/null 2>&1
 then
  echo "*** Tag '$refname' already exists." >&2
  echo "*** Modifying a tag is not allowed in this repository." >&2
  exit 1
 fi
 ;;
refs/heads/*,commit)
 # branch
 if [ "$oldrev" = "$zero" -a "$denycreatebranch" = "true" ]; then
  echo "*** Creating a branch is not allowed in this repository" >&2
  exit 1
 fi
 ;;
refs/heads/*,delete)
 # delete branch
 if [ "$allowdeletebranch" != "true" ]; then
  echo "*** Deleting a branch is not allowed in this repository" >&2
  exit 1
```

```
  fi
  ;;
 refs/remotes/*,commit)
  # tracking branch
  ;;
 refs/remotes/*,delete)
  # delete tracking branch
  if [ "$allowdeletebranch" != "true" ]; then
   echo "*** Deleting a tracking branch is not allowed in this repository" >&2
   exit 1
  fi
  ;;
 *)
  # Anything else (is there anything else?)
  echo "*** Update hook: unknown type of update to ref $refname of type $newrev_type" >&2
  exit 1
  ;;
esac

# --- Finished
exit 0
```

C:\Users\david/src\Ligaturizer\.git\info\exclude

git ls-files --others --exclude-from=.git/info/exclude

Lines that start with '#' are comments.

For a project mostly in C, the following would be a good set of

exclude patterns (uncomment them if you want to use them):

*.[oa]

*~

C:\Users\david/src\Ligaturizer\.git\logs\HEAD

0000000000000000000000000000000000000000 c4065187a544a8fab40826fc91db1c6180a2d342 Ian <ia

C:\Users\david/src\Ligaturizer\.git\logs\refs\heads\master

0000000000000000000000000000000000000000 c4065187a544a8fab40826fc91db1c6180a2d342 Ian <ia

C:\Users\david/src\Ligaturizer\.git\logs\refs\remotes\origin\HEAD

0000000000000000000000000000000000000000 c4065187a544a8fab40826fc91db1c6180a2d342 Ian <ia

C:\Users\david/src\Ligaturizer\.git\objects\pack\pack-cfc472a771c38e2e9643a7479b0c6df7fa208daa.idx

C:\Users\david/src\Ligaturizer\.git\objects\pack\pack-cfc472a771c38e2e9643a7479b0c6df7fa208daa.pack

C:\Users\david/src\Ligaturizer\.git\objects\pack\pack-cfc472a771c38e2e9643a7479b0c6df7fa208daa.rev

C:\Users\david/src\Ligaturizer\.git\refs\heads\master

c4065187a544a8fab40826fc91db1c6180a2d342

C:\Users\david/src\Ligaturizer\.git\refs\remotes\origin\HEAD

ref: refs/remotes/origin/master

C:\Users\david/src\Ligaturizer\.mypy_cache\.gitignore

```
# Automatically created by mypy

*
```

C:\Users\david/src\Ligaturizer\.mypy_cache\CACHEDIR.TAG

Signature: 8a477f597d28d172789f06886806bc55

This file is a cache directory tag automatically created by mypy.

For information about cache directory tags see https://bford.info/cachedir/

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\@plugins_snapshot.json

{}

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\abc.data.json

{".class":"MypyFile","_fullname":"abc","future_import_flags":[],"is_partial_stub_package":false,"is_stub":true

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\abc.meta.json

{"data_mtime":1708019931,"dep_lines":[4,1,2,5,6,1],"dep_prios":[5,5,10,5,5,5],"dependencies":["collections

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\build.data.json

{".class":"MypyFile","_fullname":"build","future_import_flags":[],"is_partial_stub_package":false,"is_stub":fal

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\build.meta.json

{"id":"build","path":"c:\\Users\\david\\src\\Ligaturizer\\build.py","mtime":1708020064,"size":5114,"hash":"0a5

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\builtins.data.json

{".class":"MypyFile","_fullname":"builtins","future_import_flags":[],"is_partial_stub_package":false,"is_stub":

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\builtins.meta.json

{"data_mtime":1708019931,"dep_lines":[30,1,2,3,4,5,31,35,56,1,1],"dep_prios":[5,10,5,10,5,5,5,5,5,30,30],

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\char_dict.data.json

{".class":"MypyFile","_fullname":"char_dict","future_import_flags":[],"is_partial_stub_package":false,"is_stub

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\char_dict.meta.json

{"data_mtime":1708019932,"dep_lines":[1,1,1],"dep_prios":[5,30,30],"dependencies":["builtins","abc","typin

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\codecs.data.json

{".class":"MypyFile","_fullname":"codecs","future_import_flags":[],"is_partial_stub_package":false,"is_stub":

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\codecs.meta.json

{"data_mtime":1708019931,"dep_lines":[6,1,2,3,4,5,7,8,1,1],"dep_prios":[5,10,10,5,5,5,5,5,5,30],"depender

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\contextlib.data.json

{".class":"MypyFile","_fullname":"contextlib","future_import_flags":[],"is_partial_stub_package":false,"is_stu

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\contextlib.meta.json

{"data_mtime":1708019931,"dep_lines":[5,1,2,3,6,7,8,1,1],"dep_prios":[5,5,10,5,5,5,5,5,30],"dependencies"

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\dataclasses.data.json

{".class":"MypyFile","_fullname":"dataclasses","future_import_flags":[],"is_partial_stub_package":false,"is_s

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\dataclasses.meta.json

{"data_mtime":1708019931,"dep_lines":[6,1,2,3,4,5,7,8,1],"dep_prios":[5,10,10,5,5,5,5,5,30],"dependencies

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\enum.data.json

{".class":"MypyFile","_fullname":"enum","future_import_flags":[],"is_partial_stub_package":false,"is_stub":tr

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\enum.meta.json

{"data_mtime":1708019931,"dep_lines":[6,1,2,3,5,7,8,1],"dep_prios":[5,5,10,10,5,5,5,30],"dependencies":["

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\genericpath.data.json

{".class":"MypyFile","_fullname":"genericpath","future_import_flags":[],"is_partial_stub_package":false,"is_s

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\genericpath.meta.json

{"data_mtime":1708019931,"dep_lines":[3,1,2,4,5,1,1],"dep_prios":[5,10,5,5,5,5,30],"dependencies":["colle

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\glob.data.json

{".class":"MypyFile","_fullname":"glob","future_import_flags":[],"is_partial_stub_package":false,"is_stub":tru

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\glob.meta.json

{"data_mtime":1708019932,"dep_lines":[3,1,2,4,1,1],"dep_prios":[5,10,5,5,5,30],"dependencies":["collection

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\io.data.json

{".class":"MypyFile","_fullname":"io","future_import_flags":[],"is_partial_stub_package":false,"is_stub":true,"

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\io.meta.json

{"data_mtime":1708019930,"dep_lines":[6,1,2,3,4,5,7,8,9,10,1],"dep_prios":[5,10,10,10,10,5,5,5,5,5,30],"de

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\ligatures.data.json

{".class":"MypyFile","_fullname":"ligatures","future_import_flags":[],"is_partial_stub_package":false,"is_stub

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\ligatures.meta.json

{"data_mtime":1708019932,"dep_lines":[1,1,1],"dep_prios":[5,30,30],"dependencies":["builtins","abc","typin

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\ligaturize.data.json

{".class":"MypyFile","_fullname":"ligaturize","future_import_flags":[],"is_partial_stub_package":false,"is_stub

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\ligaturize.meta.json

{"data_mtime":1708019932,"dep_lines":[13,12,14,16,17,1,1,1,1,1,10,11],"dep_prios":[10,10,10,5,5,5,30,30

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\ntpath.data.json

{".class":"MypyFile","_fullname":"ntpath","future_import_flags":[],"is_partial_stub_package":false,"is_stub":t

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\ntpath.meta.json

{"data_mtime":1708019930,"dep_lines":[1,2,3,16,19,45,46,1,1],"dep_prios":[10,5,5,5,5,5,5,5,30],"depender

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\pathlib.data.json

{".class":"MypyFile","_fullname":"pathlib","future_import_flags":[],"is_partial_stub_package":false,"is_stub":t

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\pathlib.meta.json

{"data_mtime":1708019930,"dep_lines":[12,1,2,13,14,15,16,17,1,1,1],"dep_prios":[5,10,5,5,5,5,5,5,5,30,30

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\posixpath.data.json

{".class":"MypyFile","_fullname":"posixpath","future_import_flags":[],"is_partial_stub_package":false,"is_stu

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\posixpath.meta.json

{"data_mtime":1708019930,"dep_lines":[3,1,2,4,17,18,19,1,1],"dep_prios":[5,10,5,5,5,5,5,5,30],"dependenc

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\re.data.json

{".class":"MypyFile","_fullname":"re","future_import_flags":[],"is_partial_stub_package":false,"is_stub":true,"

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\re.meta.json

{"data_mtime":1708019930,"dep_lines":[5,1,2,3,4,6,7,8,11,1,1,1],"dep_prios":[5,10,10,10,5,5,5,5,5,5,30,30

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\sre_compile.data.json

{".class":"MypyFile","_fullname":"sre_compile","future_import_flags":[],"is_partial_stub_package":false,"is_s

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\sre_compile.meta.json

{"data_mtime":1708019930,"dep_lines":[1,2,4,5,1,1],"dep_prios":[5,5,5,5,5,30],"dependencies":["re","sre_c

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\sre_constants.data.json

{".class":"MypyFile","_fullname":"sre_constants","future_import_flags":[],"is_partial_stub_package":false,"is

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\sre_constants.meta.json

{"data_mtime":1708019930,"dep_lines":[1,2,3,1,1,1],"dep_prios":[10,5,5,5,30,30],"dependencies":["sys","ty

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\sre_parse.data.json

{".class":"MypyFile","_fullname":"sre_parse","future_import_flags":[],"is_partial_stub_package":false,"is_stu

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\sre_parse.meta.json

{"data_mtime":1708019930,"dep_lines":[2,1,3,4,6,7,1,1,1],"dep_prios":[5,10,5,5,5,5,5,30,30],"dependencies

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\subprocess.data.json

{".class":"MypyFile","_fullname":"subprocess","future_import_flags":[],"is_partial_stub_package":false,"is_s

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\subprocess.meta.json

{"data_mtime":1708019930,"dep_lines":[3,1,2,4,5,6,2603,1,1,1,1],"dep_prios":[5,10,5,5,5,5,5,5,30,30,30],"d

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\sys.data.json

{".class":"MypyFile","_fullname":"sys","future_import_flags":[],"is_partial_stub_package":false,"is_stub":true

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\sys.meta.json

{"data_mtime":1708019930,"dep_lines":[4,5,6,2,3,7,8,9,10,1,1],"dep_prios":[5,5,5,5,5,5,5,5,5,30,30],"deper

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\types.data.json

{".class":"MypyFile","_fullname":"types","future_import_flags":[],"is_partial_stub_package":false,"is_stub":tr

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\types.meta.json

{"data_mtime":1708019930,"dep_lines":[3,16,1,2,19,20,631,1,1],"dep_prios":[5,5,10,5,5,5,5,30,30],"depend

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\typing.data.json

{".class":"MypyFile","_fullname":"typing","future_import_flags":[],"is_partial_stub_package":false,"is_stub":tr

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\typing.meta.json

{"data_mtime":1708019930,"dep_lines":[1,2,3,4,5,6,7,8,9,1],"dep_prios":[10,10,5,5,5,5,5,5,5,5],"dependenc

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\typing_extensions.data.json

{".class":"MypyFile","_fullname":"typing_extensions","future_import_flags":[],"is_partial_stub_package":fals

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\typing_extensions.meta.json

{"data_mtime":1708019930,"dep_lines":[454,1,2,3,4,5,6,74,1],"dep_prios":[5,10,10,10,5,5,5,5,5],"depender

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12_ast.data.json

{".class":"MypyFile","_fullname":"_ast","future_import_flags":[],"is_partial_stub_package":false,"is_stub":tru

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12_ast.meta.json

{"data_mtime":1708019931,"dep_lines":[1,2,3,1,1,1,1],"dep_prios":[10,5,5,5,30,30,30],"dependencies":["sys

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12_codecs.data.json

{".class":"MypyFile","_fullname":"_codecs","future_import_flags":[],"is_partial_stub_package":false,"is_stub

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12_codecs.meta.json

{"data_mtime":1708019931,"dep_lines":[4,1,2,3,5,6,1,1,1],"dep_prios":[5,10,10,5,5,5,5,30,30],"dependenci

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12_collections_abc.data.json

{".class":"MypyFile","_fullname":"_collections_abc","future_import_flags":[],"is_partial_stub_package":false,

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12_collections_abc.meta.json

{"data_mtime":1708019931,"dep_lines":[1,2,3,4,35,1,1],"dep_prios":[10,5,5,5,5,5,30],"dependencies":["sys"

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12_winapi.data.json

{".class":"MypyFile","_fullname":"_winapi","future_import_flags":[],"is_partial_stub_package":false,"is_stub"

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12_winapi.meta.json

{"data_mtime":1708019931,"dep_lines":[3,1,2,4,5,1,1,1],"dep_prios":[5,10,5,5,5,5,30,30],"dependencies":["

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\collections\abc.data.json

{".class":"MypyFile","_fullname":"collections.abc","future_import_flags":[],"is_partial_stub_package":false,"is

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\collections\abc.meta.json

{"data_mtime":1708019931,"dep_lines":[1,1,1,1],"dep_prios":[5,5,30,30],"dependencies":["_collections_abc

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\collections__init__.data.json

{".class":"MypyFile","_fullname":"collections","future_import_flags":[],"is_partial_stub_package":false,"is_stu

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\collections__init__.meta.json

{"data_mtime":1708019931,"dep_lines":[11,1,2,3,4,5,8,1,1],"dep_prios":[5,10,5,5,5,5,5,5,30],"dependencies

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email\charset.data.json

{".class":"MypyFile","_fullname":"email.charset","future_import_flags":[],"is_partial_stub_package":false,"is_

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email\charset.meta.json

{"data_mtime":1708019931,"dep_lines":[1,2,3,1,1],"dep_prios":[5,5,5,5,30],"dependencies":["collections.ab

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email\contentmanager.data.json

{".class":"MypyFile","_fullname":"email.contentmanager","future_import_flags":[],"is_partial_stub_package"

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email\contentmanager.meta.json

{"data_mtime":1708019931,"dep_lines":[1,2,3,1,1],"dep_prios":[5,5,5,5,30],"dependencies":["collections.ab

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email\errors.data.json

{".class":"MypyFile","_fullname":"email.errors","future_import_flags":[],"is_partial_stub_package":false,"is_s

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email\errors.meta.json

{"data_mtime":1708019931,"dep_lines":[1,1,1,1,1],"dep_prios":[10,5,30,30,30],"dependencies":["sys","builti

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email\header.data.json

{".class":"MypyFile","_fullname":"email.header","future_import_flags":[],"is_partial_stub_package":false,"is_

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email\header.meta.json

{"data_mtime":1708019931,"dep_lines":[1,2,3,1,1],"dep_prios":[5,5,5,5,30],"dependencies":["collections.ab

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email\message.data.json

{".class":"MypyFile","_fullname":"email.message","future_import_flags":[],"is_partial_stub_package":false,"i

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email\message.meta.json

{"data_mtime":1708019931,"dep_lines":[1,3,4,5,6,2,7,8,1,1],"dep_prios":[5,5,5,5,5,5,5,5,5,30],"dependenci

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email\policy.data.json

{".class":"MypyFile","_fullname":"email.policy","future_import_flags":[],"is_partial_stub_package":false,"is_s

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email\policy.meta.json

{"data_mtime":1708019931,"dep_lines":[2,3,4,5,6,1,7,8,1],"dep_prios":[5,5,5,5,5,5,5,5,5],"dependencies":["

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email__init__.data.json

{".class":"MypyFile","_fullname":"email","future_import_flags":[],"is_partial_stub_package":false,"is_stub":tr

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\email__init__.meta.json

{"data_mtime":1708019931,"dep_lines":[1,2,3,4,5,1,1],"dep_prios":[5,5,5,5,5,5,30],"dependencies":["collect

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\importlib\abc.data.json

{".class":"MypyFile","_fullname":"importlib.abc","future_import_flags":[],"is_partial_stub_package":false,"is_

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\importlib\abc.meta.json

{"data_mtime":1708019930,"dep_lines":[14,15,1,2,3,4,13,16,17,18,1,1,1],"dep_prios":[5,5,10,10,10,5,5,5,5

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\importlib\machinery.data.json

{".class":"MypyFile","_fullname":"importlib.machinery","future_import_flags":[],"is_partial_stub_package":fal

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\importlib\machinery.meta.json

{"data_mtime":1708019930,"dep_lines":[1,5,9,1,2,3,4,6,1,1,1],"dep_prios":[10,5,5,20,10,10,5,5,5,30,30],"de

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\importlib__init__.data.json

{".class":"MypyFile","_fullname":"importlib","future_import_flags":[],"is_partial_stub_package":false,"is_stub

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\importlib__init__.meta.json

{"data_mtime":1708019931,"dep_lines":[1,2,3,1,1,1],"dep_prios":[5,5,5,5,30,30],"dependencies":["collection

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\importlib\metadata_meta.data.json

{".class":"MypyFile","_fullname":"importlib.metadata._meta","future_import_flags":[],"is_partial_stub_packa

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\importlib\metadata_meta.meta.json

{"data_mtime":1708019930,"dep_lines":[1,2,1,1],"dep_prios":[5,5,5,30],"dependencies":["collections.abc","t

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\importlib\metadata__init__.data.json

{".class":"MypyFile","_fullname":"importlib.metadata","future_import_flags":[],"is_partial_stub_package":fals

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\importlib\metadata__init__.meta.json

{"data_mtime":1708019930,"dep_lines":[31,5,6,7,1,2,3,4,8,10,11,12,1],"dep_prios":[5,5,5,5,10,5,10,5,5,5,5

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\os\path.data.json

{".class":"MypyFile","_fullname":"os.path","future_import_flags":[],"is_partial_stub_package":false,"is_stub":

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\os\path.meta.json

{"data_mtime":1708019930,"dep_lines":[1,4,1,1,1],"dep_prios":[10,5,5,30,30],"dependencies":["sys","ntpath

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\os__init__.data.json

{".class":"MypyFile","_fullname":"os","future_import_flags":[],"is_partial_stub_package":false,"is_stub":true,"

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12\os__init__.meta.json

{"data_mtime":1708019930,"dep_lines":[23,30,1,2,21,22,24,25,26,27,28,33,1],"dep_prios":[5,10,10,5,5,5,5

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12_typeshed__init__.data.json

{".class":"MypyFile","_fullname":"_typeshed","future_import_flags":[],"is_partial_stub_package":false,"is_stu

C:\Users\david/src\Ligaturizer\.mypy_cache\3.12_typeshed__init__.meta.json

{"data_mtime":1708019931,"dep_lines":[6,5,7,8,9,10,11,1,1,1],"dep_prios":[5,10,5,5,5,5,5,5,30,30],"depend

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-Black.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-BlackItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-Bold.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-BoldItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-ExtraLight.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-ExtraLightItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-Italic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-Light.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-LightItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-Medium.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-MediumItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-Regular.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-SemiBold.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFont-SemiBoldItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-Black.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-BlackItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-Bold.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-BoldItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-ExtraLight.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-ExtraLightItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-Italic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-Light.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-LightItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-Medium.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-MediumItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-Regular.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-SemiBold.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontMono-SemiBoldItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-Black.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-BlackItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-Bold.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-BoldItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-ExtraLight.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-ExtraLightItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-Italic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-Light.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-LightItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-Medium.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-MediumItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-Regular.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-SemiBold.ttf

C:\Users\david/src\Ligaturizer\fonts\SauceCodeProNerdFontPropo-SemiBoldItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\SourceCodeVariable-Roman.ttf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight\FantasqueSansMono-Bold.otf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight\FantasqueSansMono-Bold.ttf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight\FantasqueSansMono-BoldItalic

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight\FantasqueSansMono-BoldItalic

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight\FantasqueSansMono-Italic.otf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight\FantasqueSansMono-Italic.ttf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight\FantasqueSansMono-Regular.

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight\FantasqueSansMono-Regular.t

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight\LICENSE.txt

This Font Software is licensed under the SIL Open Font License, Version 1.1.
This license is copied below, and is also available with a FAQ at:
http://scripts.sil.org/OFL

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

PREAMBLE
The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

DEFINITIONS

"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS

Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components,

in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION

This license becomes null and void if any of the above conditions are

not met.

DISCLAIMER

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight-NoLoopK\FantasqueSansMono

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight-NoLoopK\FantasqueSansMono

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight-NoLoopK\FantasqueSansMono

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight-NoLoopK\FantasqueSansMono

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight-NoLoopK\FantasqueSansMono

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight-NoLoopK\FantasqueSansMono

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight-NoLoopK\FantasqueSansMono

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight-NoLoopK\FantasqueSansMono

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-LargeLineHeight-NoLoopK\LICENSE.txt

This Font Software is licensed under the SIL Open Font License, Version 1.1.
This license is copied below, and is also available with a FAQ at:
http://scripts.sil.org/OFL

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

PREAMBLE
The goals of the Open Font License (OFL) are to stimulate worldwide
development of collaborative font projects, to support the font creation
efforts of academic and linguistic communities, and to provide a free and
open framework in which fonts may be shared and improved in partnership
with others.

The OFL allows the licensed fonts to be used, studied, modified and
redistributed freely as long as they are not sold by themselves. The
fonts, including any derivative works, can be bundled, embedded,
redistributed and/or sold with any software provided that any reserved
names are not used by derivative works. The fonts and derivatives,
however, cannot be released under any other type of license. The
requirement for fonts to remain under this license does not apply
to any document created using the fonts or their derivatives.

DEFINITIONS

"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS

Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components,

in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION

This license becomes null and void if any of the above conditions are

not met.

DISCLAIMER

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-NoLoopK\FantasqueSansMono-Bold.otf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-NoLoopK\FantasqueSansMono-Bold.ttf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-NoLoopK\FantasqueSansMono-BoldItalic.otf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-NoLoopK\FantasqueSansMono-BoldItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-NoLoopK\FantasqueSansMono-Italic.otf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-NoLoopK\FantasqueSansMono-Italic.ttf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-NoLoopK\FantasqueSansMono-Regular.otf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-NoLoopK\FantasqueSansMono-Regular.ttf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-NoLoopK\LICENSE.txt

---

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

PREAMBLE

The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

DEFINITIONS

"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS

Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components,

in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION

This license becomes null and void if any of the above conditions are

not met.

DISCLAIMER

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-Normal\FantasqueSansMono-Bold.otf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-Normal\FantasqueSansMono-Bold.ttf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-Normal\FantasqueSansMono-BoldItalic.otf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-Normal\FantasqueSansMono-BoldItalic.ttf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-Normal\FantasqueSansMono-Italic.otf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-Normal\FantasqueSansMono-Italic.ttf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-Normal\FantasqueSansMono-Regular.otf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-Normal\FantasqueSansMono-Regular.ttf

C:\Users\david/src\Ligaturizer\fonts\FantasqueSansMono-Normal\LICENSE.txt

---

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

PREAMBLE

The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

DEFINITIONS

"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS

Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components,

in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION

This license becomes null and void if any of the above conditions are

not met.

DISCLAIMER

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

C:\Users\david/src\Ligaturizer\fonts\Hermit\Hermit-Bold.otf

C:\Users\david/src\Ligaturizer\fonts\Hermit\Hermit-BoldItalic.otf

C:\Users\david/src\Ligaturizer\fonts\Hermit\Hermit-Light.otf

C:\Users\david/src\Ligaturizer\fonts\Hermit\Hermit-LightItalic.otf

C:\Users\david/src\Ligaturizer\fonts\Hermit\Hermit-Regular.otf

C:\Users\david/src\Ligaturizer\fonts\Hermit\Hermit-RegularItalic.otf

C:\Users\david/src\Ligaturizer\fonts\output\.gitignore

```
*
```

C:\Users\david/src\Ligaturizer\fonts\output-with-characters\.gitignore

```
*
```

C:\Users\david/src\Ligaturizer\images\banner.png

C:\Users\david/src\Ligaturizer\images\menlo.png

C:\Users\david/src\Ligaturizer\images\sf-mono.png