

# UFOs, DesignSpaces, RoboFont, & More

*Using the UFO format for variable  
font design, plus a Q&A / Demo*

## FONTS USED IN THIS PRESENTATION

- **RECURSIVE**, BY ARROW TYPE (ME)
- **NEWSREADER**, BY PRODUCTION TYPE

# **Disclaimer**

**I am not a world-leading  
expert on the UFO format.**

If you really want to go deep on the technical reasoning & history behind the UFO spec, you may wish to talk with someone like Just van Rossum, Tal Leming, or Erik van Blokland (the original creators of the format), or Ben Kiel, Frederik Berlaen, or Denis Moyogo Jacquerye (frequent contributors to the spec, in varying ways).

**However..**

**I have used UFOs & RoboFont  
nearly daily for 3+ years.**

**In this session, I will:**

- Share some of the strengths of the UFO format,  
and try to put it into my own words.
- Demo some useful tools & methods for using  
UFOs, especially RoboFont & DesignSpace files.
- Do my best to answer your questions.

# Why UFO?

(Unified Font Object)

**Other formats  
are nice, too.**

**Here are some  
strengths of UFO.**

“The Unified Font Object (UFO) is a cross-platform, cross-application, human readable, future proof format for storing font data.”

[HTTPS://UNIFIEDFONTOBJECT.ORG/](https://unifiedfontobject.org/)

## THE UFO DESIGN PHILOSOPHY

- THE DATA MUST BE HUMAN READABLE AND HUMAN EDITABLE.
- THE DATA SHOULD BE APPLICATION INDEPENDENT.
- DATA DUPLICATION SHOULD BE AVOIDED UNLESS ABSOLUTELY NECESSARY.

[HTTPS://UNIFIEDFONTOBJECT.ORG/](https://unifiedfontobject.org/)

# The data must be human readable and human editable.

## *Why it matters*

- UFO is based in XML, which is easy(ish) to read, edit, and manipulate by hand or with scripts. It is an open spec itself.
- Not tied to any specific programming language (e.g. Python) – all languages handle XML.
- Changes in a Git diff are possible to read and work with in normal tools for code, like GitHub & VS Code – no special applications are required (but some are useful, and certainly there is room to make this all easier & more accessible).

# The data should be application independent.

## *Why it matters*

- Font sources should be long-lasting. No single company's success or business model should dictate your ability to use your data.
- Font development is more than just drawing. UFO enables scripting outside of an editor, interchange between apps & tools, and new apps to be made.

## *Ben Kiel*

*Type design is more than just the final outlines. Sketches, unused glyphs, layers, notes, marks, etc are all part of a project and have extreme value (especially when you come back to a project years later). This data does not live in a font binary.*

# Data duplication should be avoided unless absolutely necessary.

## *Why it matters*

- The OpenType spec has lots of cases of duplication: e.g. the many Name IDs, or the multiple ways of defining what an “ascender” is.
- Instead of abstracting data and adding *yet more duplication* for designers to understand, the UFO aims to be a fairly direct way to record design data, e.g. to make OpenType fonts (but with possibilities beyond that, too).
- If you come from a GlyphsApp perspective (as I did), some things seem to be “duplicated” between UFO sources in the same type family. I’ll share some ways to deal with this.

# **Backstory**

## **Erik & Just had to exchange glyphs. Tal wanted to make a kerning editor.**

- UFO is 17 years old; the earliest versions were just a project between Erik, Just, and Tal.
- Superpolator came along and changed things for everyone in a good way. UFO started to be taken more seriously, so the team made UFO2 to formalize things.
- Glyphs & RoboFont showed what an editor could be. This led to UFO3. UFO4 is in development.
- The UFO spec is still maintained largely by Erik, Just, Ben, & Tal, as a way to serve & give back to the type community. It provides a foundation for their work and for many other projects.

*From notes on a July  
2020 UFO Spec Meeting*

**Tal Leming**

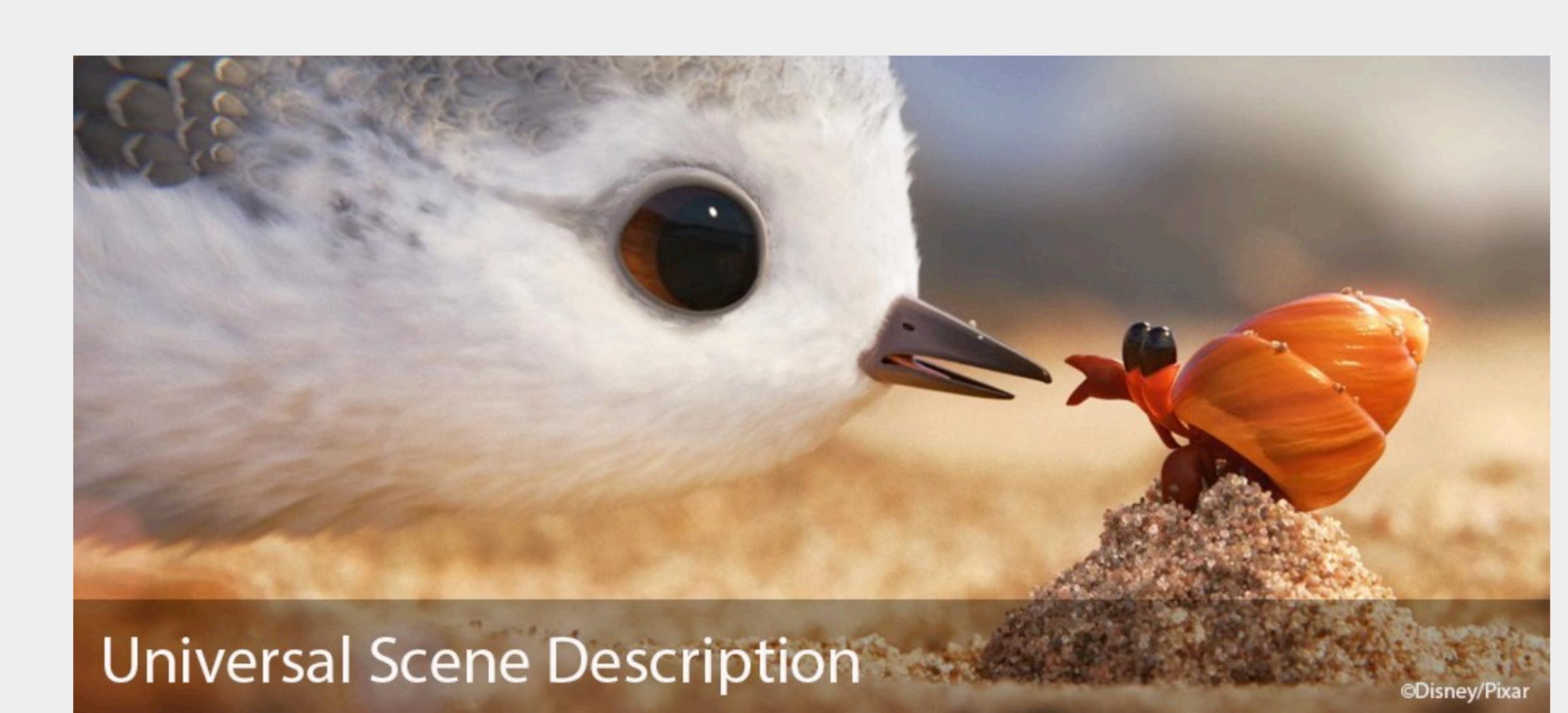
UFO was not designed to be a “be everything to everyone” format. It was designed to be a source format for the design portion of the type design process.

There have been attempts to bend it into a high-efficiency compilation source format, but it's never going to be great at that because the needs of the iterative design phase and the compilation phase are very different. The things that need to be done to turn it into a robust compilation source go against the original purposes of its creation. I have long suggested that someone with deeper pockets than me fund the creation of a format designed specifically for compilation and the compiler for it. UFO, Glyphs, etc. could then export to that format and the compiler would compile. That's a big project, but, hey, so were UFO + ufoLib + defcon + fontParts + fontMath + all that other stuff.

graphics.pixar.com

User Docs API Docs Downloads and Videos Help GITHUB

# Introduction to USD



Universal Scene Description ©Disney/Pixar

- Introduction to USD 
  - Getting Help with USD
    - USD Frequently Asked Questions
    - USD Glossary
    - Maximizing USD Performance
  - USD Developer API Reference
  - USD Toolset
  - USD 3rd Party Plugins
    - Alembic USD Plugin
    - Houdini USD Plugins
      - A Tour of USD Houdini Primitives
      - Writing USD from Houdini
    - Katana USD Plugins
    - Maya USD Plugins
    - RenderMan USD Imaging Plugin
  - USD Tutorials
    - Hello World - Creating Your First USD Stage
    - Hello World Redux - Using Generic Prims
    - Inspecting and Authoring Properties
    - Referencing Layers
    - Converting Between Layer Formats
    - Traversing a Stage
    - Authoring Variants
    - Variants Example in Katana
    - Transformations, Time-sampled Animation, and Layer Offsets
    - Simple Shading in USD
    - End to End Example
    - Generating New Schema Classes
    - Houdini USD Example Workflow
    - Creating a Usdview Plugin
  - Contributing to USD
  - USD Contributors
  - Press
    - Open Source Announcement
    - Open Source Release
  - USD White Papers
    - Adapting UsdLux to the Needs of Renderers
    - Asset Resolution (Ar) 2.0
    - Coordinate Systems in USD Proposal
    - Generalizing Connectable Nodes Beyond UsdShade
    - Render Settings in USD Proposal
    - Rigid Body Physics in USD Proposal

## What is USD?

Pipelines capable of producing computer graphics films and games typically generate, store, and transmit great quantities of 3D data, which we call "scene description". Each of many cooperating applications in the pipeline (modeling, shading, animation, lighting, fx, rendering) typically has its own special form of scene description tailored to the specific needs and workflows of the application, and with no standard way to interoperate between them. Universal Scene Description (USD) is the first publicly available software

# **Long-term reliability**

# What happens to my Adobe CC files if I unsubscribe?

-\\_(ツ)\_/-

Market Summary > Adobe Inc

460.20 USD

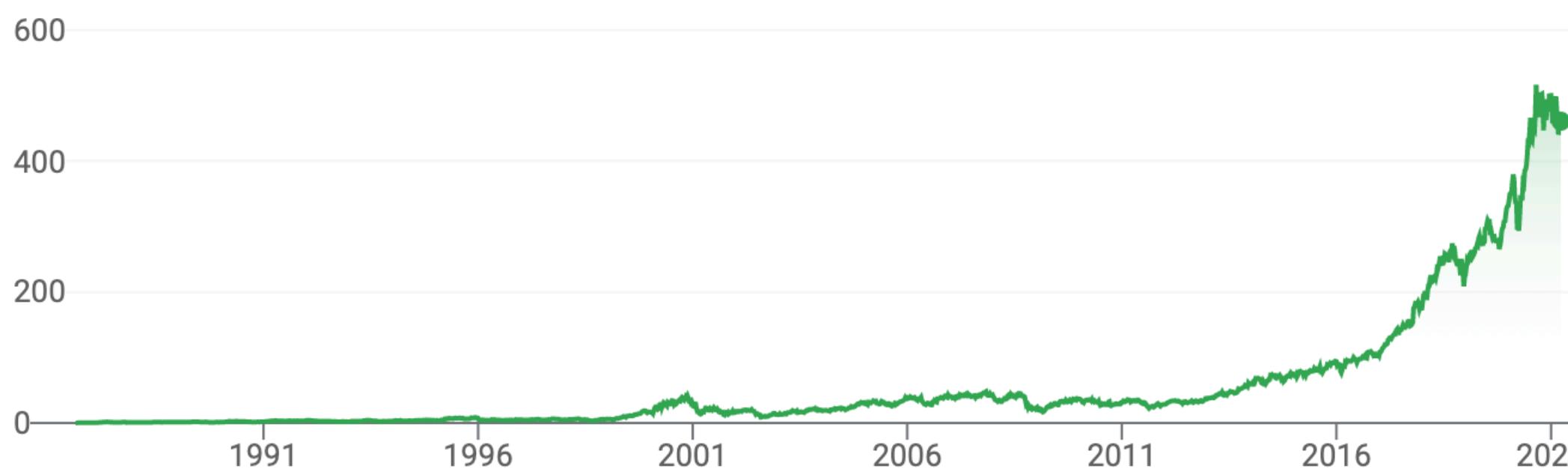
+7.79 (1.72%) ↑

NASDAQ: ADBE

+ Follow

Closed: Mar 23, 7:59 PM EDT · Disclaimer  
After hours 460.20 0.00 (0.00%)

1 day | 5 days | 1 month | 6 months | ytd | 1 year | 5 years | max



Open	457.35	Mkt cap	220.59B	Prev close	452.41
High	466.88	P/E ratio	42.43	52-wk high	536.88
Low	456.41	Div yield	-	52-wk low	287.78

# **Designed for design-time flexibility**

*(In some ways, beyond what  
font editors make obvious)*

## One example: Layers!

There are two primary layer models in the font world:

- font level layers: In this model, all glyphs have the same layers. A good example of this is a chromatic font.
- glyph level layers: In this model, individual glyphs may have their own unique layers.

*From the FontParts Docs*

## **Also: lib.plist**

This file is a place to store authoring tool specific, user specific or otherwise arbitrary data for the font. It is optional. If it is not defined in the UFO, there is no lib data. Data that is too complex or too large for the lib can be stored in the data directory.

*From the UFO Spec*

# Tools

## TOOLS I USE WITH UFOS INCLUDE...

- ROBOFONT
- VS CODE, GIT, & GITHUB
- PYTHON & FONTPARTS
- DESIGNSPACES & DESIGNSPACE EDIT
- METRICS MACHINE, PREPOLATOR, & SKATEBOARD

# RoboFont

UFO ≠ RoboFont

# **UFO is specifically not tied to a single tool, but RoboFont is a handy editor.**

The operating philosophy behind RoboFont is:

The tools you choose influence your creative process.

Because of this, RoboFont provides many opportunities for the user to tailor the application to their design process. It is strict about not performing ‘auto-magic’ on one’s font files, ‘auto-’ anything is avoided if possible.

*From the Welcome page  
of the RoboFont Docs*

The screenshot shows a dark-themed web browser window. At the top, the address bar displays "blog.arrowtype.com". The main content area features a large, bold, white title: "GETTING THE MOST OUT OF ROBOFONT". Above the title, a smaller text reads "PUBLISHED DECEMBER 11, 2020". Below the title, there is a paragraph of text: "Type designers are lucky to have a variety of extremely-capable tools for creating new fonts. My favorite of these is the font editor RoboFont, along with its ecosystem." Further down, another paragraph states: "I covered some of the reasons I like RoboFont in my earlier post, [Getting Started in Type](#). This post actually started as a section of that article, but I have since realized that I would like to add more extensions and more details. If you haven't already read that article, there are lots of helpful tips there, so consider giving it a look, too!" At the bottom, a section titled "First things first" includes a small decorative icon. A note below it says: "Like most powerful software, RoboFont has a lot of functionality, and you will save time by doing a bit of reading before diving in. It is extremely useful to". The browser interface includes standard controls like back/forward buttons, a search bar, and a tab labeled "Info".

The Arrow Type Blog

PUBLISHED DECEMBER 11, 2020

# GETTING THE MOST OUT OF ROBOFONT

Type designers are lucky to have a variety of extremely-capable tools for creating new fonts. My favorite of these is the font editor RoboFont, along with its ecosystem.

I covered some of the reasons I like RoboFont in my earlier post, [Getting Started in Type](#). This post actually started as a section of that article, but I have since realized that I would like to add more extensions and more details. If you haven't already read that article, there are lots of helpful tips there, so consider giving it a look, too!

**First things first** ⚡

Like most powerful software, RoboFont has a lot of functionality, and you will save time by doing a bit of reading before diving in. It is extremely useful to

# From the blog post: “RoboFont extensions that I find extremely useful.”

- Mechanic, to handle extensions. It's pretty essential, and allows you to easily find and install most of the other extensions listed here. Extensions mentioned but not available via Mechanic are linked to.
- Add Overlap, to add corner overlaps to selected points. I like Okay Type's fork of this, which adds a little UI at the bottom-right corner of Glyph Windows which allows positive, negative, or zero overlap to corner points.
- Add Segment Guideline, a very simple but useful way to add a guideline to a selected segment
- Batch, to help you build fonts from RoboFont
- CornerTools – really nice tools to modify corners (read the readme on there for instructions)
- Glyph Browser, to make it easy to add specific glyphs with correct unicodes & naming
- DesignSpace Edit, to help set up designspace documents that you can build with FontMake
- EditThatNextMaster, to help working with multiple related fonts
- Eyeliner, to make it easy to set when a point is on a guide or metric line
- GlyphMirror, an extension to draw mirrored versions of your drawing in a background visual, which makes it fast and easy to achieve symmetry. (This one is not yet available on Mechanic; get it from its GitHub repo
- GlyphMirror, an extension to draw mirrored versions of your drawing in a background visual, which makes it fast and easy to achieve symmetry. (This one is not yet available on Mechanic; get it from its GitHub repo
- Properties, to display distances in selected points, etc
- Overlay UFOs, to display glyphs from the same or other open UFOs while you draw
- Outliner, if you want to draw a single line, then expand it, e.g. as a quick way to start a Comic Sans-style font.
- Ramsay St, to show related glyphs in the glyph view (though, I use the Space Center quite a bit more for this)
- ScalingEditTool, a super-necessary tool to move points in a way that is less disruptive to attached curves
- ScaleFast, extremely awesome for scaling glyphs to speed up the process of making things like superiors & inferiors, small caps, legal symbols, and more
- Shape Tool, to draw circles & rectangles
- SpeedPunk, as one way to consider the smoothness of curves
- Theme Editor, to make it easy to control your drawing theme
- word-o-mat, to create strings to judge type in the Space Center

# Other tools for working with UFOs

- Code editors & Git/GitHub
- FontParts, a Python API to manipulate UFOs – this can be used in and outside of RoboFont
- Defcon, designed to make building GUI type tools that interact with the UFO easier to build on Mac
- Glyphs, FontLab, and FontForge will import & export UFOs. RuneBender uses UFO primarily.
- FontMake uses UFO in its pipeline, even for building binary fonts from Glyphs files.
- Lots of other one-off or single purpose tools: RoundUFO, UFOStretch, FontGoggles, etc.

*Key goal of fontParts: an open api, which is easy to understand, and is as easy as possible for font editors to implement*

# **FontParts is awesome and pretty intuitive**

*Ben Kiel*

Key goals of FontParts:

- An open API
- Easy to understand
- As easy as possible for font editors to implement

# **VS Code & Git**

The screenshot shows a dark-themed code editor interface. In the center, a large white 'Code Editor' logo is displayed. The top navigation bar shows 'README.md — recursive'. The left sidebar has sections for 'SOURCE CONTROL', 'CHANGES' (11 items), 'COMMIT', 'REPOSITORIES', 'FILE HISTORY', 'BRANCHES', 'REMOTES', 'STASHES', and 'TAGS'. The bottom right corner features a 'Git UI' logo.

README.md — recursive

Source Control

Changes (11)

- ampersand.code\_experimental\_2.glif src/ufo/mono/Re...
- ampersand.code\_experimental.glif src/ufo/mono/Rec...
- ampersand.crossbar.glif src/ufo/mono/Recursive Mono...
- dagger.daggyer.glif src/ufo/mono/Recursive Mono-Cas...
- dollar.lower.glif src/ufo/mono/Recursive Mono-Casual B...
- g.extra.glif src/ufo/mono/Recursive Mono-Casual B.ufo/...
- onehalf.v1.glif src/ufo/mono/Recursive Mono-Casual B.u...
- r.simple\_italic.glif src/ufo/mono/Recursive Mono-Casual...
- x.ital\_nov19.glif src/ufo/mono/Recursive Mono-Casual B...
- Z.\_noserif.glif src/ufo/mono/Recursive Mono-Casual B.u...
- zero.ss02.glif src/ufo/mono/Recursive Mono-Casual B.u...

Git UI

Source Control

Changes (11)

- ampersand.code\_experimental\_2.glif src/ufo/mono/Re...
- ampersand.code\_experimental.glif src/ufo/mono/Rec...
- ampersand.crossbar.glif src/ufo/mono/Recursive Mono...
- dagger.daggyer.glif src/ufo/mono/Recursive Mono-Cas...
- dollar.lower.glif src/ufo/mono/Recursive Mono-Casual B...
- g.extra.glif src/ufo/mono/Recursive Mono-Casual B.ufo/...
- onehalf.v1.glif src/ufo/mono/Recursive Mono-Casual B.u...
- r.simple\_italic.glif src/ufo/mono/Recursive Mono-Casual...
- x.ital\_nov19.glif src/ufo/mono/Recursive Mono-Casual B...
- Z.\_noserif.glif src/ufo/mono/Recursive Mono-Casual B.u...
- zero.ss02.glif src/ufo/mono/Recursive Mono-Casual B.u...

Code Editor

You, a week ago | 5 authors (arrowtype and others)

10 # Recursive Sans & Mono

11 Learn more on [the Recursive website](#) ([specimens](#), [repo-artwork](#): [recursive.design](#))

12 ! [Recursive Sans & Mono Repo Artwork] ([specimens/repo-artwork/recursive-repo-artwork.png](#))

13

14

15

16 Recursive Sans & Mono is a variable-type family built for better code. [See it in action](#).

TERMINAL PROBLEMS 270 OUTPUT DEBUG CONSOLE 1: zsh ▾ + ⌫ ⌄ ⌈ ×

~/type-repos/recursive main ✘

▶ git status

On branch main

Your branch is behind 'origin/main' by 2 commits, and can be fast-forwarded.

(use "git pull" to update your local branch)

Revert currently in progress.

(run "git revert --continue" to continue)

(use "git revert --skip" to skip this patch)

(use "git revert --abort" to cancel the revert operation)

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in the working tree)

- modified: src/ufo/mono/Recursive Mono-Casual B.ufo/glyphs/\_noserif.glif
- modified: src/ufo/mono/Recursive Mono-Casual B.ufo/glyphs/ampersand.code\_experimental.glif
- modified: src/ufo/mono/Recursive Mono-Casual B.ufo/glyphs/ampersand.code\_experimental\_2.glif
- modified: src/ufo/mono/Recursive Mono-Casual B.ufo/glyphs/ampersand.crossbar.glif
- modified: src/ufo/mono/Recursive Mono-Casual B.ufo/glyphs/dagger.daggyer.glif
- modified: src/ufo/mono/Recursive Mono-Casual B.ufo/glyphs/dollar.lower.glif
- modified: src/ufo/mono/Recursive Mono-Casual B.ufo/glyphs/g.extra.glif
- modified: src/ufo/mono/Recursive Mono-Casual B.ufo/glyphs/onehalf.v1.glif
- modified: src/ufo/mono/Recursive Mono-Casual B.ufo/glyphs/r.simple\_italic.glif
- modified: src/ufo/mono/Recursive Mono-Casual B.ufo/glyphs/x.ital\_nov19.glif
- modified: src/ufo/mono/Recursive Mono-Casual B.ufo/glyphs/zero.ss02.glif

no changes added to commit (use "git add" and/or "git commit -a")

main\* ↻ 2↓ 0↑ Python 3.7.3 64-bit ('venv': venv) ↻ 150 ⚡ 61 ⓘ 59 ⚡ Blame arrowtype (8 months ago) ⚡ swr410@gmail.com 65 overall score Spaces: 2 UTF-8 LF Markdown ✓ Prettier ⚡ ⚡

Y\_glif (Working Tree) — recursive

src > ufo > sans > Recursive Sans-Linear A.ufo > glyphs > Y\_glif > glyph > outline > contour

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <glyph name="Y" format="2">
3   <unicode hex="0059"/>
4   <advance width="650"/>
5   <outline>
6     > <contour>
7-    >   <point x="345" y="0" type="c"/>
8-    >   <point x="345" y="32"/>
9-    >   <point x="345" y="65"/>
10-   >   <point x="345" y="100" type="c"/>
11-   >   <point x="345" y="170"/>
12-   >   <point x="345" y="250"/>
13-   >   <point x="345" y="320" type="c"/>
14-   >   <point x="295" y="320" type="c"/>
15-   >   <point x="295" y="217"/>
16-   >   <point x="295" y="114"/>
17-   >   <point x="295" y="11" type="c"/>
18-   >   <point x="295" y="5"/>
19-   >   <point x="299" y="0"/>
20-   >   <point x="306" y="0" type="c"/>
21-   >   <point x="319" y="0"/>
22-   >   <point x="332" y="0"/>
23-   > </contour>
24-   > <contour>
25-   >   <point x="41" y="700" type="c"/>
26-   >   <point x="120" y="569"/>
27-   >   <point x="218" y="409"/>
28-   >   <point x="297" y="281" type="c"/>
29-   >   <point x="300" y="278"/>
30-   >   <point x="307" y="277"/>
31-   >   <point x="341" y="277" type="c"/>
1 <?xml version="1.0" encoding="UTF-8"?>
2 <glyph name="Y" format="2">
3   <unicode hex="0059"/>
4   <advance width="650"/>
5   <outline>
6     > <contour>
7+    >   <point x="352" y="0" type="c"/>
8+    >   <point x="352" y="32"/>
9+    >   <point x="352" y="65"/>
10+   >   <point x="352" y="100" type="c"/>
11+   >   <point x="352" y="170"/>
12+   >   <point x="352" y="250"/>
13+   >   <point x="352" y="320" type="c"/>
14+   >   <point x="302" y="320" type="c"/>
15+   >   <point x="302" y="217"/>
16+   >   <point x="302" y="114"/>
17+   >   <point x="302" y="11" type="c"/>
18+   >   <point x="302" y="5"/>
19+   >   <point x="306" y="0"/>
20+   >   <point x="313" y="0" type="c"/>
21+   >   <point x="326" y="0"/>
22+   >   <point x="339" y="0"/>
23+   > </contour>
24+   > <contour>
25+   >   <point x="48" y="700" type="c"/>
26+   >   <point x="127" y="569"/>
27+   >   <point x="225" y="409"/>
28+   >   <point x="304" y="281" type="c"/>
29+   >   <point x="307" y="278"/>
30+   >   <point x="314" y="277"/>
31+   >   <point x="348" y="277" type="c"/>
```

SOURCE CONTROL

Message (⌘Enter to commit on 'main')

Changes

Y\_glif src/ufo/sans/Recursive Sans-Linear A.ufo... ↗ ↘ + M 1

COMMS

REPOSITORIES

FILE HISTORY

BRANCHES

REMOTES

STASHES

TAGS

main\* ⌂ 1↓ 1↑ Python 3.7.3 64-bit ('venv': venv) ⌂ 6 △ 3 ⌂ Not Committed Yet ⌂ You, seconds ago Ln 7, Col 1 Tab Size: 4 UTF-8 LF XML ⌂ ⌂

Inter.glyphs (Working Tree) — inter

src > Inter.glyphs

```
60841 angle = 90;
60842 position = "{804, 1160}";
60843 }
60844 );
60845 layerId = "C698F293-3EC0-4A5A-A3A0-0F
60846 paths = (
60847 {
60848 closed = 1;
60849 nodes = (
60850 "936 -36 OFFCURVE",
60851 "1076 108 OFFCURVE",
60852 "1124 208 CURVE",
60853 "1136 208 LINE",
60854 "1136 0 LINE",
60855 "1372 0 LINE",
60856 "1372 1012 LINE SMOOTH",
60857 "1372 1500 OFFCURVE",
60858 "1000 1556 OFFCURVE",
60859 "804 1556 CURVE SMOOTH",
60860 "572 1556 OFFCURVE",
60861 "308 1476 OFFCURVE",
60862 "188 1196 CURVE",
60863 "412 1116 LINE",
60864 "464 1228 OFFCURVE",
60865 "587 1348 OFFCURVE",
60866 "812 1348 CURVE SMOOTH",
60867 "1029 1348 OFFCURVE",
60868 "1136 1233 OFFCURVE",
60869 "1136 1036 CURVE SMOOTH",
60870 "1136 1028 LINE SMOOTH",
60871 "1136 014 OFFCURVE"
60844 angle = 90;
60845 position = "{804, 1160}";
60846 }
60847 );
60848 layerId = "C698F293-3EC0-4A5A-A3A0-0F
60849 paths = (
60850 {
60851 closed = 1;
60852 nodes = (
60853 "956 -36 OFFCURVE",
60854 "1096 108 OFFCURVE",
60855 "1144 208 CURVE",
60856 "1156 208 LINE",
60857 "1156 0 LINE",
60858 "1392 0 LINE",
60859 "1392 1012 LINE SMOOTH",
60860 "1392 1500 OFFCURVE",
60861 "1020 1556 OFFCURVE",
60862 "824 1556 CURVE SMOOTH",
60863 "592 1556 OFFCURVE",
60864 "328 1476 OFFCURVE",
60865 "208 1196 CURVE",
60866 "432 1116 LINE",
60867 "484 1228 OFFCURVE",
60868 "607 1348 OFFCURVE",
60869 "832 1348 CURVE SMOOTH",
60870 "1049 1348 OFFCURVE",
60871 "1156 1233 OFFCURVE",
60872 "1156 1036 CURVE SMOOTH",
60873 "1156 1028 LINE SMOOTH",
60874 "1156 014 OFFCURVE"
```

SOURCE CONTROL

Message (⌘Enter to commit on 'qa')

Changes

Inter.glyphs src

COMMS

REPOSITORIES

FILE HISTORY

BRANCHES

REMOTES

STASHES

TAGS

qa\* ⌘ 0↓ 4↑ Python 3.7.3 64-bit ⌘ 0 ▲ 0 ⌘ Not Committed Yet ⌘ swr410@gmail.com ⌘ You, seconds ago Ln 60888, Col 1 Spaces: 4 UTF-8 LF Plain Text ⌘ ⌘ ⌘

# **Python & FontParts**

copy-kerning-without-metrics\_machine.py — recursive

copy-kerning-without-metrics\_machine.py

src > 01-shell-scripts-for-sources > metrics > copy-kerning-without-metrics\_machine.py > ...

```
38
39
40     groupsPath = f"{ufoToCopyFrom}/groups.plist"
41     kerningPath = f"{ufoToCopyFrom}/kerning.plist"
42
43     for ufo in sorted(ufosToCopyTo):
44         ufoPath = f"{head}/{ufo}"
45
46         if ufoPath != ufoToCopyFrom and 'features' not in ufo and 'instances' not in ufo:
47             print(ufoPath)
48             groupsDest = f"{ufoPath}/groups.plist"
49             # good if mono or sans
50             shutil.copyfile(groupsPath, groupsDest)
51             # only copy kerning file for sans UFOs
52             if 'Sans' in ufo:
53                 kerningDest = f"{ufoPath}/kerning.plist"
54                 shutil.copyfile(kerningPath, kerningDest)
55             else:
56                 pass
57
```

TERMINAL PROBLEMS 4 OUTPUT DEBUG CONSOLE

1: zsh

```
4d 4d
▶ py src/01-shell-scripts-for-sources/metrics/copy-kerning-without-metrics_machine.py "src/ufo/sans/Recursive Sans-Casual C.ufo" src/ufo/sans
Copying from UFO to UFOs in another Directory
src/ufo/sans/Recursive Sans-Casual A Slanted.ufo
src/ufo/sans/Recursive Sans-Casual A.ufo
src/ufo/sans/Recursive Sans-Casual B Slanted.ufo
src/ufo/sans/Recursive Sans-Casual B.ufo
src/ufo/sans/Recursive Sans-Casual C Slanted.ufo
src/ufo/sans/Recursive Sans-Linear A Slanted.ufo
src/ufo/sans/Recursive Sans-Linear A.ufo
src/ufo/sans/Recursive Sans-Linear B Slanted.ufo
src/ufo/sans/Recursive Sans-Linear B.ufo
src/ufo/sans/Recursive Sans-Linear C Slanted.ufo
src/ufo/sans/Recursive Sans-Linear C.ufo
src/ufo/sans/kerning
src/ufo/sans/partial-designspaces
```

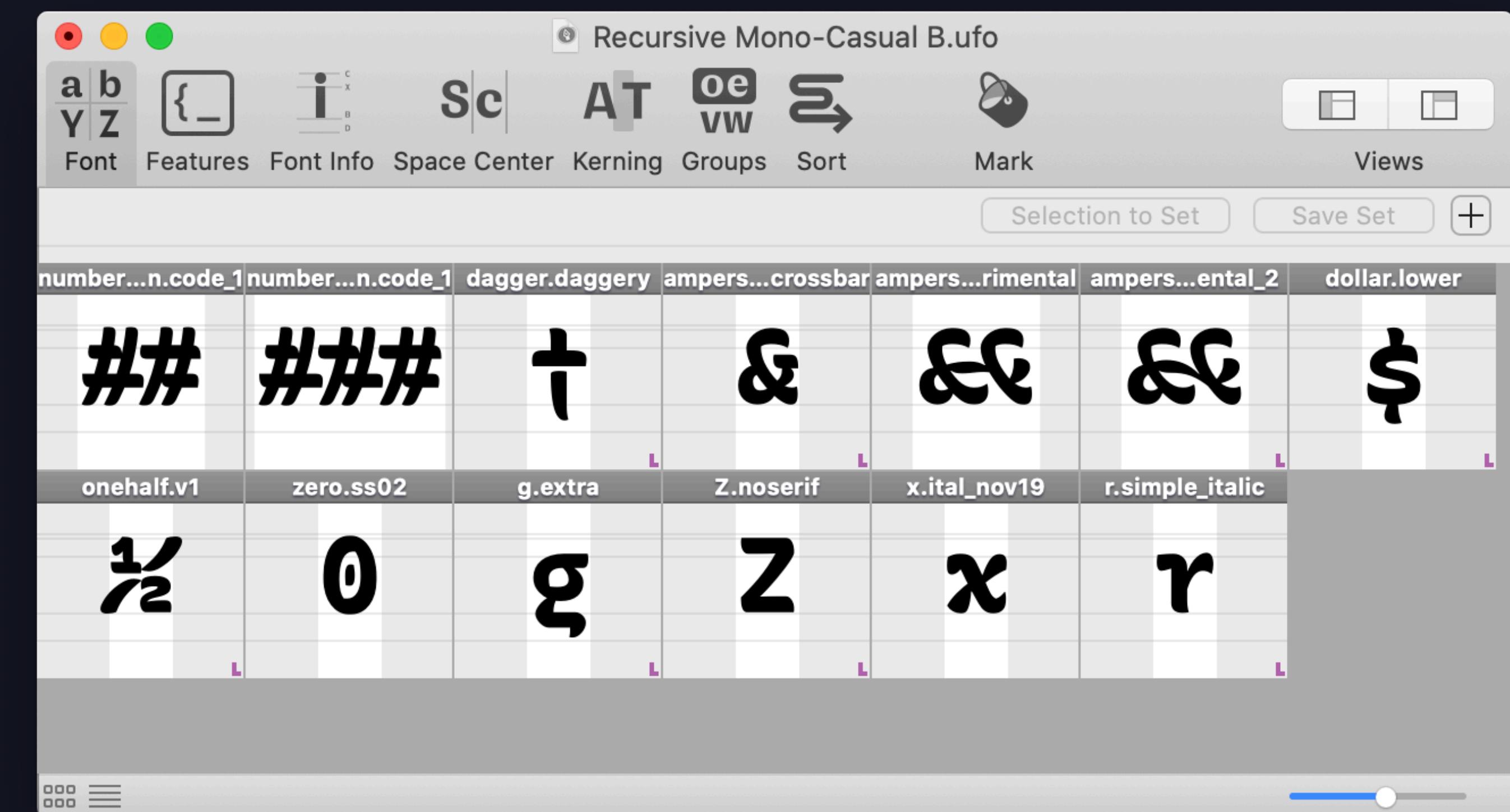
~/type-repos/recursive main x

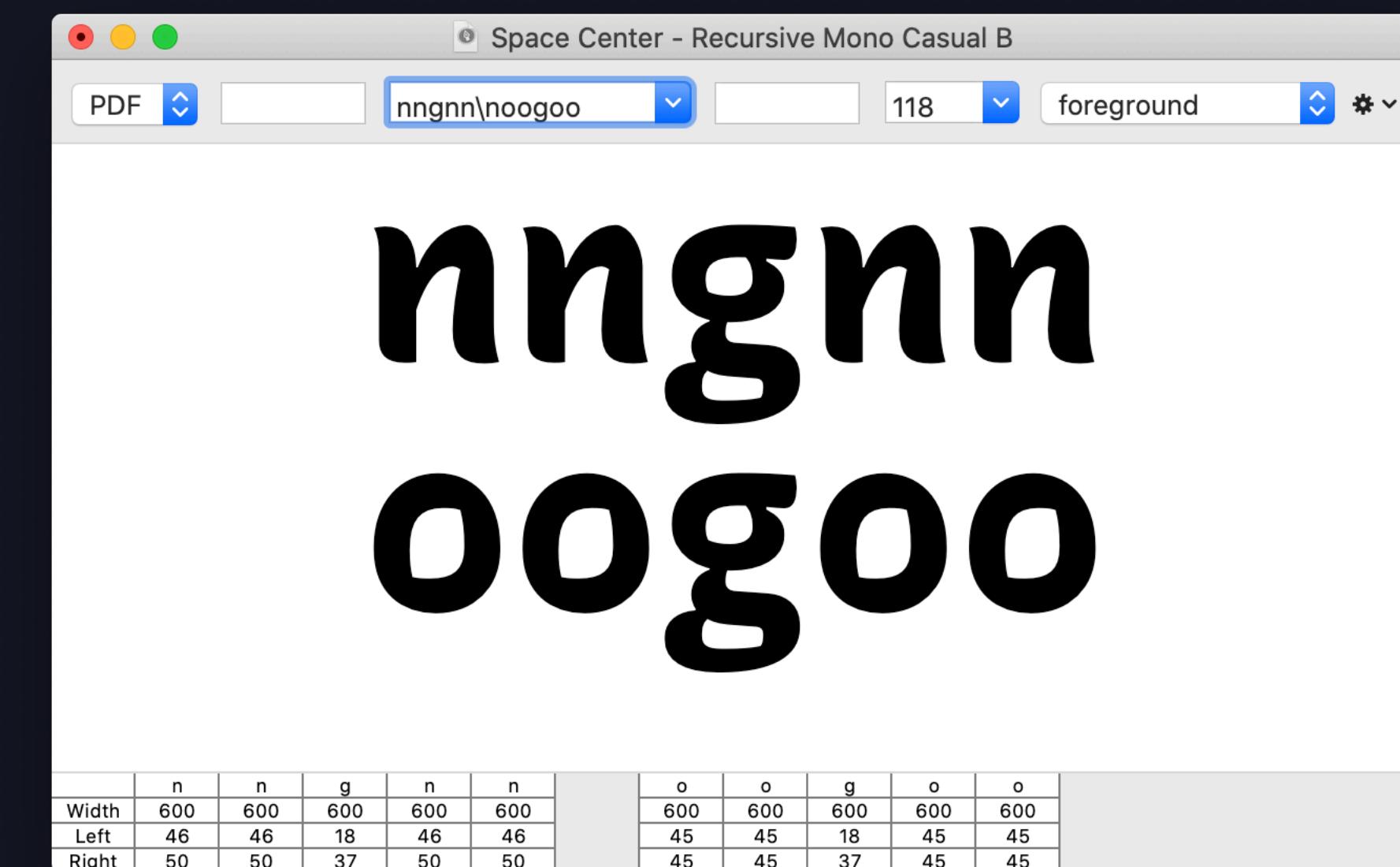
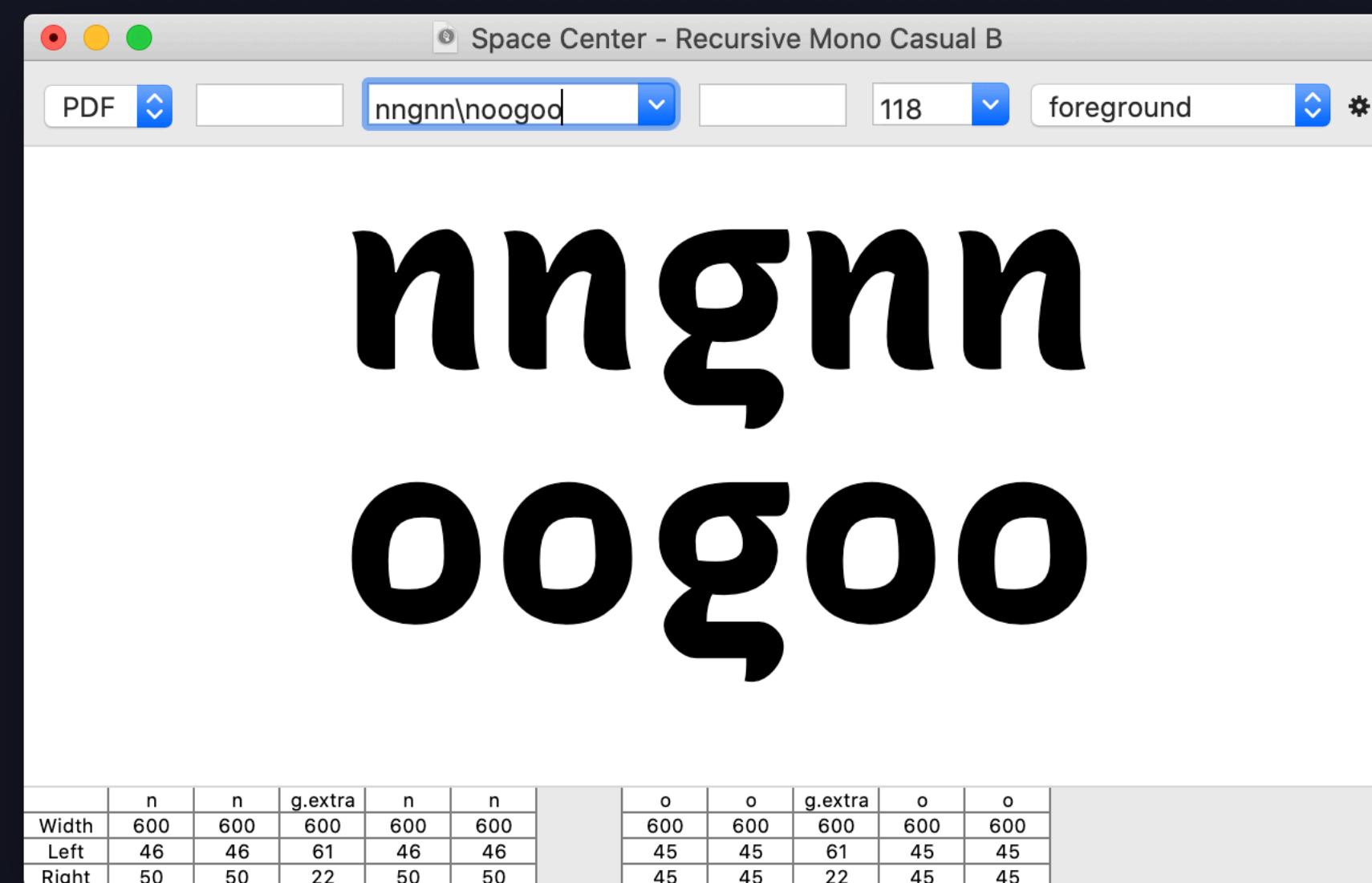
4d 4d

main\* 2 0 1 Python 3.7.3 64-bit ('venv': venv) 1 3 Blame Stephen Nixon (1 year ago) Stephen Nixon, a year ago Ln 1, Col 1 Spaces: 4 UTF-8 LF Python

# **FontParts usecase:**

## **build-prep scripts**





```
def makeCompatible(fonts):
    """
    Checks all glyphs in *fonts* for compatibility. Removes any glyphs that
    aren't compatible from all of the fonts.

    *fonts* is a 'list' of font objects (Defcon or FontParts).
    """

    local_report = report.get("Removed non-compatible glyphs", [])
    nonCompatible = []

    for glyph in fonts[0]:
        for font in fonts[1:]:
            if glyph.name in font.keys():
                compatibility = glyph.isCompatible(font[glyph.name])
                if not compatibility[0]:
                    nonCompatible.append((glyph.name, str(compatibility)))
                else:
                    nonCompatible.append((glyph.name, "Missing in font"))

    for font in fonts:
        removeGlyphs(font, [name for name, _ in nonCompatible])

    if nonCompatible != []:
        local_report.append(nonCompatible)

    report["Removed non-compatible glyphs"] = local_report
```

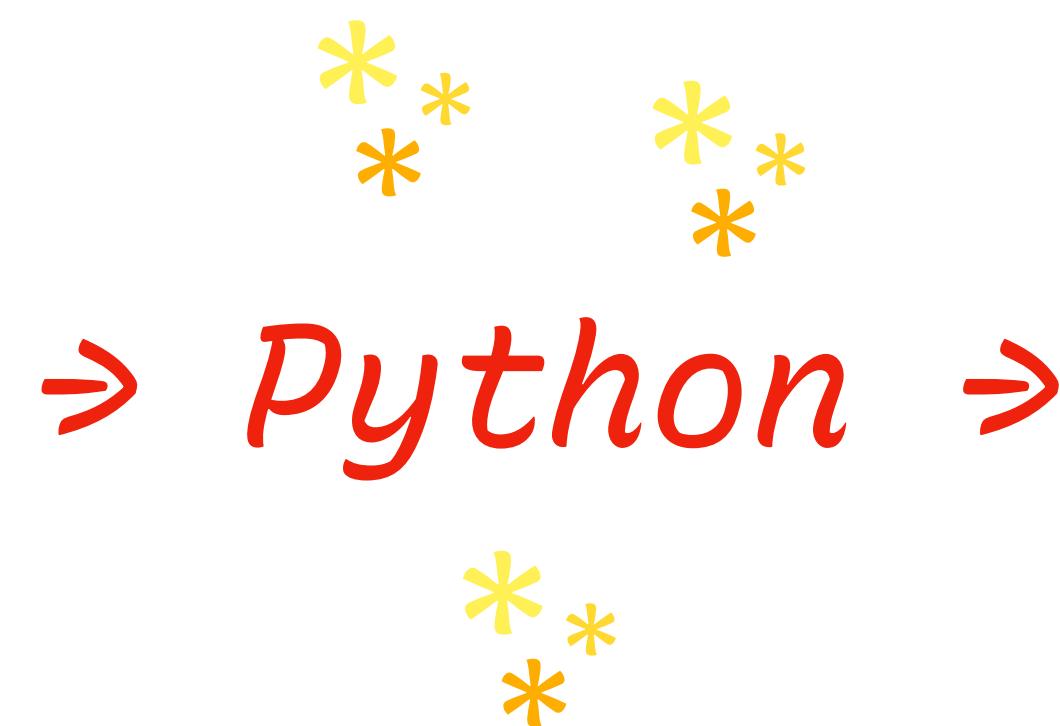
*From:*  
recursive/  
mastering/  
prep\_fonts.py

## IN ANOTHER PROJECT...

What I draw:  
1 glyph per character

## Working Sources

- Light
- XBold
- Irregular Light
- Irregular XBold



What I build fonts from:  
3 alts per main character

## Build Sources

- Light
- XBold
- Irregular Light
- Irregular XBold
- Bouncy Light
- Bouncy XBold
- Reverse Bouncy Light
- Reverse Bouncy XBold

**DesignSpaces**  
+  
**DesignSpaceEdit**

Code File Edit Selection View Go Run Terminal Window Help

- Select All ⌘A
- Expand Selection ⌘↑⌘→
- Shrink Selection ⌘↑⌘←
- Copy Line Up ⌘↑
- Copy Line Down ⌘↓
- Move Line Up ⌘↑
- Move Line Down ⌘↓
- Duplicate Selection

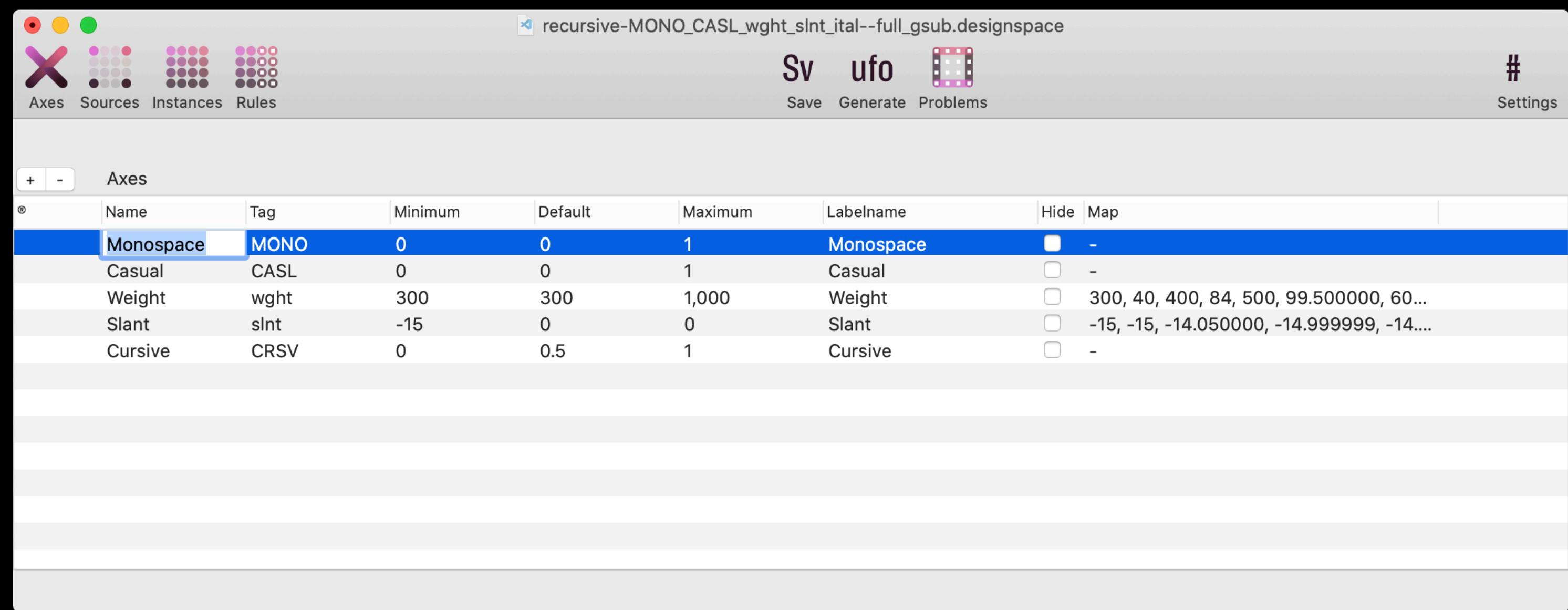
- Add Cursor Above ⌘⌘↑
- Add Cursor Below ⌘⌘↓
- Add Cursors to Line Ends ⌘↑⌘I
- Add Next Occurrence ⌘D
- Add Previous Occurrence ⌘P
- Select All Occurrences ⌘⌘L
- Switch to Cmd+Click for Multi-Cursor
- Column Selection Mode

```
545     <dimension name="Monospace" xvalue="1" />
546     </location>
547   </source>
548   <source filename="mono/Recursive Mono-Linear B.ufo">
549     <location>
550       <dimension name="Monospace" xvalue="1" />
551       <dimension name="Casual" xvalue="0" />
552       <dimension name="Weight" xvalue="146" />
553       <dimension name="Slant" xvalue="0" />
554       <dimension name="Cursive" xvalue="0.500000" />
555     </location>
556   </source>
557   <source filename="mono/Recursive Mono-Linear C Slanted.ufo">
558     <location>
559       <dimension name="Monospace" xvalue="1" />
560       <dimension name="Casual" xvalue="0" />
561       <dimension name="Weight" xvalue="215" />
562       <dimension name="Slant" xvalue="-15" />
563       <dimension name="Cursive" xvalue="0.500000" />
564     </location>
```

recursive-MONO\_CASL\_wght\_slnt\_ital--full\_gsub.designspace — recursive

546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564

main 2 0 0 Python 3.7.3 64-bit ('venv': venv) 0 0 0 Blame Stephen Nixon (1 year ago) 99 selections (891 characters selected) Spaces: 2 UTF-8 LF XML



## DesignSpaceEdit

# DesignSpaceLib Python API, in FontTools

## DesignSpaceDocument Specification

An object to read, write and edit interpolation systems for typefaces. Define sources, axes, rules and instances.

- [The Python API of the objects](#)
- [The document XML structure](#)

### Python API

#### DesignSpaceDocument object

The DesignSpaceDocument object can read and write `.designspace` data. It imports the axes, sources and instances to very basic **descriptor** objects that store the data in attributes. Data is added to the document by creating such descriptor objects, filling them with data and then adding them to the document. This makes it easy to integrate this object in different contexts.

The **DesignSpaceDocument** object can be subclassed to work with different objects, as long as they have the same attributes. Reader and Writer objects can be subclassed as well.

**Note:** Python attribute names are usually camelCased, the corresponding [XML](#) attributes are usually all lowercase.

```
from fontTools.designspaceLib import DesignSpaceDocument
doc = DesignSpaceDocument()
doc.read("some/path/to/my.designspace")
doc.axes
doc.sources
doc.instances
```

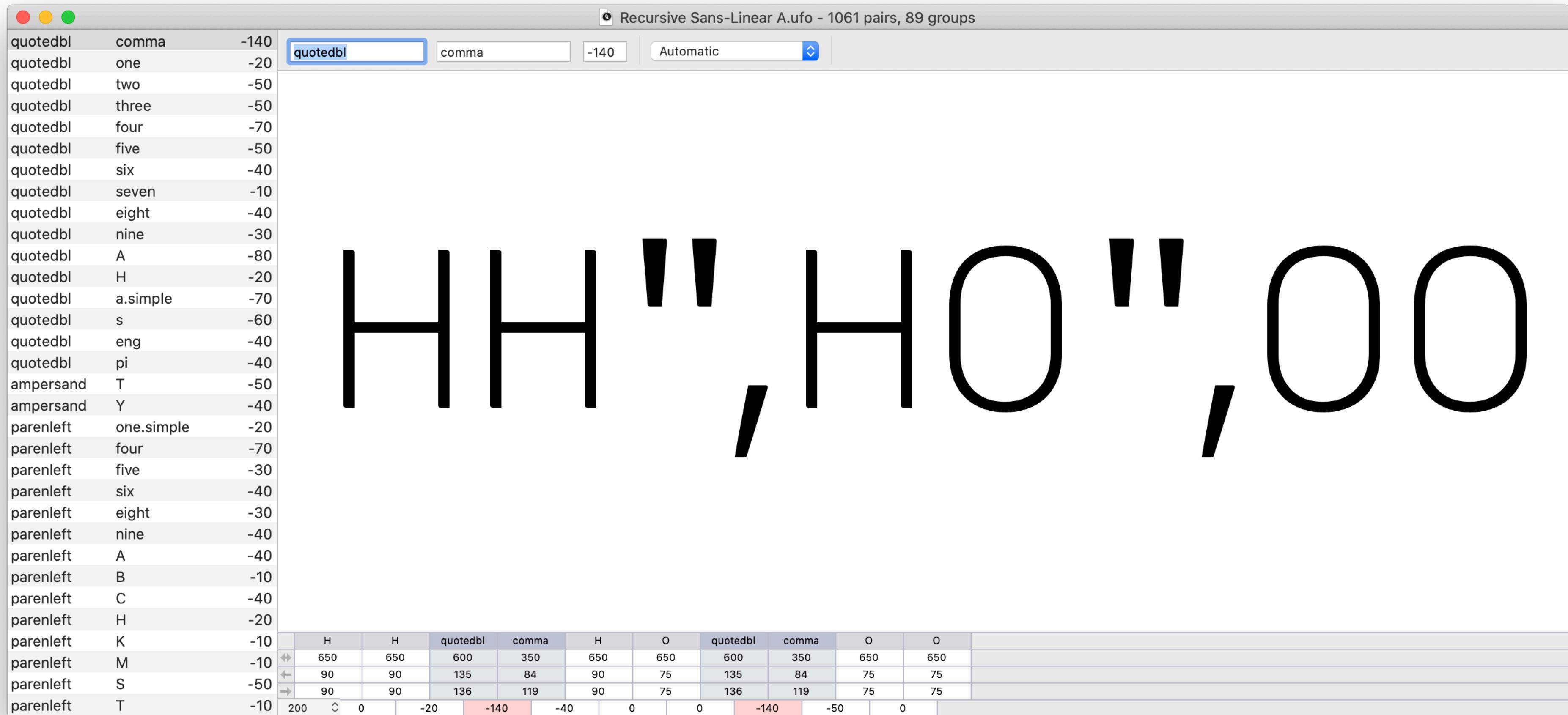
#### Attributes

- `axes` : list of axisDescriptors
- `sources` : list of sourceDescriptors
- `instances` : list of instanceDescriptors
- `rules` : list if ruleDescriptors
- `readerClass` : class of the reader object

```
71
72     ds = DesignSpaceDocument.fromfile(designspacePath)          # read designspace file
73     sources = [source.path for source in ds.sources]           # make list of source paths
74     paths = {}                                                 # set up empty dict
75     for fontPath in sources:                                   # go through sources list
76         f = os.path.split(fontPath)[1]                          # get filename of UFO
77         newPath = os.path.join(outRoot, f)                      # make a filename for new location
78         paths[f] = newPath                                     # add new path to paths dict
79     shutil.copytree(fontPath, newPath, ignore=ignore)        # copy UFO to the new location
80
```

*From:  
recursive/  
mastering/  
prep\_fonts.py*

# **Metrics Machine**



Metrics Machine, helpful for kerning

Recursive Sans-Linear A.ufo - 1061 pairs, 89 groups

A	seven	-50
A	B	-10
A	C	-20
A	T	-80
A	V	-30
A	W	-20
A	Y	-90
A	backslash	-90
A	bracketright	-20
A	a.simple	10
A	b	20
A	l	20
A	v	-20
A	braceright	-20
A	dotlessi	-10
A	IJ	-20
A	eng	10
A	Nhookleft	10
A	acute	-50
A	H	650
A	H	650
A	A	650
A	Y	650
A	H	650
A	O	650
A	A	650
A	Y	650
A	O	650
A	O	650
B	acute	-50
B	H	650
B	H	650
B	A	650
B	Y	650
B	H	650
B	O	650
B	A	650
B	Y	650
B	O	650
B	O	650
A	pi	-30
A	arrowleft	-10
B	parenright	-50
	200	0
	0	0
	-90	-20
	0	0
	-10	-90
	-50	0

A Y HAY H O A Y O O

Space Center - Recursive Sans Linear Light

AYA AYES SHIPWAY LATTER-DAY FRAYN  
 BREAK OF DAY MALAYAN SWAY ROSES  
 ALL THE WAY PRAYER-BOOK DRAY DAY  
 OF ATONEMENT ROCKAWAYS GRAYED MA  
 YDAY MAY-APPLES BACKPAY PAY-DESKS  
 PORTRAYING CUTAWAYS THE BIRTHDA  
 Y PARTY

AYA AYES SHIPWAY LATTER-DAY FRAYN BREAK OF DAY MALAYAN SWAY ROSES ALL THE WAY PRAYER-BOOK DRAY DAY OF ATONEMENT ROCKAWAYS GRAYED MAYDAY MAY-APPLES BACKPAY PAY-DESKS PORTRAYING CUTAWAYS THE BIRTHDAY PARTY

PDF

MM2SpaceCenter

words found: AY  
 20 words  
 English  
 Output as list sorted by width  
 Show open+close context {n}  
 Show mirrored pair (LRL)

Width	A	Y	A	space	A	Y	E	S	space	S	H	I	P	W	A	Y	space	L	A	T	T	E	R	hyphen	D	A	Y	space	F	R	A	Y	N	space	B	R	E	A	K	space	O	F	space	D	A	Y	space
Left	40	48	40		40	48	20	77		77	90	80	30	45	40	48		90	40	60	60	20	30	91	30	40	48		40	30	40	48	80		30	30	20	40	90		75	40		30	40	48	
Right	40	48	40		40	48	60	74		74	90	80	65	45	40	48		60	40	60	60	40	91	75	40	48		60	40	40	48	80		76	40	60	40	40	75	60	75	40	48				

# Tactics

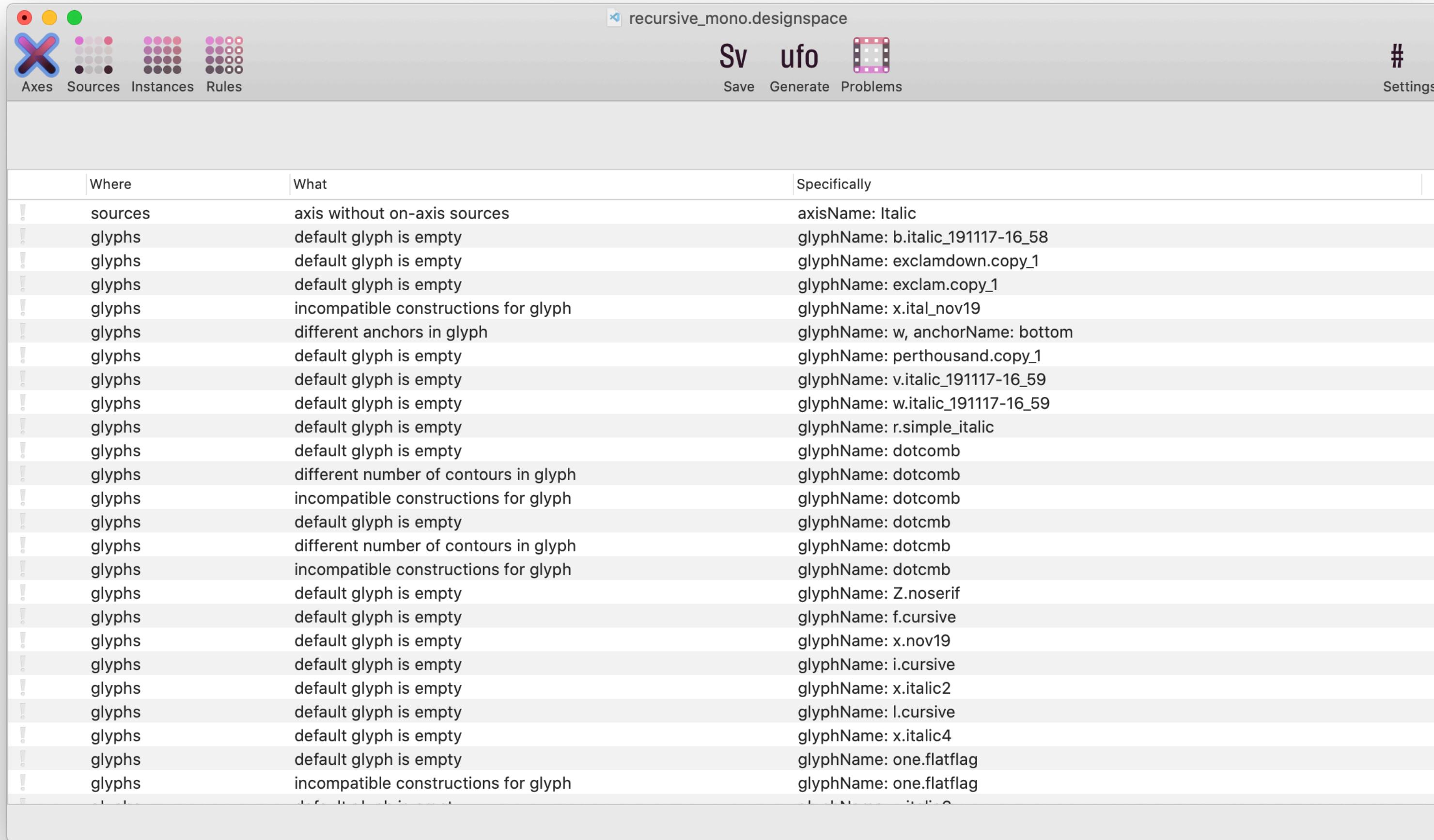
# **Compatibility Checking**

*“Is there a tool / editor approach that can be used across UFO masters to understand where the problem lies without going through every new master outline by hand to identify the problem?”*

## Checking w/ DesignspaceEdit

- Uses library DesignspaceProblems
- This can show many potential errors that might block or impact a build

# Checking w/ DesignspaceEdit



The screenshot shows the DesignspaceEdit application interface. The title bar reads "recursive\_mono.designspace". The menu bar includes "Sv" (Save), "ufo" (Generate), and "Problems". The toolbar has icons for "Axes", "Sources", "Instances", and "Rules". The main area displays a table of problems:

Where	What	Specifically
sources	axis without on-axis sources	axisName: Italic
glyphs	default glyph is empty	glyphName: b.italic_191117-16_58
glyphs	default glyph is empty	glyphName: exclamdown.copy_1
glyphs	default glyph is empty	glyphName: exclam.copy_1
glyphs	incompatible constructions for glyph	glyphName: x.ital_nov19
glyphs	different anchors in glyph	glyphName: w, anchorName: bottom
glyphs	default glyph is empty	glyphName: perthousand.copy.1
glyphs	default glyph is empty	glyphName: v.italic_191117-16_59
glyphs	default glyph is empty	glyphName: w.italic_191117-16_59
glyphs	default glyph is empty	glyphName: r.simple_italic
glyphs	default glyph is empty	glyphName: dotcomb
glyphs	different number of contours in glyph	glyphName: dotcomb
glyphs	incompatible constructions for glyph	glyphName: dotcomb
glyphs	default glyph is empty	glyphName: dotcmb
glyphs	different number of contours in glyph	glyphName: dotcmb
glyphs	incompatible constructions for glyph	glyphName: dotcmb
glyphs	default glyph is empty	glyphName: Z.noserif
glyphs	default glyph is empty	glyphName: f.cursive
glyphs	default glyph is empty	glyphName: x.nov19
glyphs	default glyph is empty	glyphName: i.cursive
glyphs	default glyph is empty	glyphName: x.italic2
glyphs	default glyph is empty	glyphName: l.cursive
glyphs	default glyph is empty	glyphName: x.italic4
glyphs	default glyph is empty	glyphName: one.flatflag
glyphs	incompatible constructions for glyph	glyphName: one.flatflag

## Even better check\_ds\_problems.py

```
1  """
2  ....Check for DesignSpaceProblems:
3  ....python <path>/check-ds-problems.py <designspacePath>
4
5  ....Requires DesignspaceProblems library:
6  ....pip install git+https://github.com/LettError/DesignspaceProblems.git
7  """
8
9  import sys
10 from pprint import pprint
11 from designspaceProblems import DesignSpaceChecker
12
13 designspacePath = sys.argv[1]
14
15 dc = DesignSpaceChecker(designspacePath)
16 dc.checkEverything()
17
18 # now all problems are stored in dc.problems
19 pprint(dc.problems)
20
```

# Checking with RoboFont/ FontParts scripts

Font	Unicodes	Icon
Mono Casual A Slanted	(626,)	🍇
Mono Casual A	(626,)	🍇
Mono Casual B Slanted	(626,)	🍇
Mono Casual B	(626,)	🍇
Mono Casual C Slanted	(626,)	🍇
Mono Casual C	(626,)	🍇
Mono Linear A Slanted	(626,)	🍇
Mono Linear A	(626,)	🍇
Mono Linear B Slanted	(626,)	🍇
Mono Linear B	(626,)	🍇
Mono Linear C Slanted	(626,)	🍇
Mono Linear C	(626,)	🍇
Sans Casual A Slanted	(626,)	🍇
Sans Casual A	(626,)	🍇
Sans Casual B Slanted	(626,)	🍇
Sans Casual B	()	🥑
Sans Casual C Slanted	(626,)	🍇
Sans Casual C	()	🥑
Sans Linear A Slanted	(626,)	🍇
Sans Linear A	()	🥑
Sans Linear B Slanted	(626,)	🍇
Sans Linear B	()	🥑
Sans Linear C Slanted	(626,)	🍇
Sans Linear C	()	🥑

- Can be built on-the-fly
- More targeted way of finding the proverbial “needle in a haystack,” like out-of-sync anchors

*Recursive has a few of these*

# Taking notes for next time!

The screenshot shows a GitHub issue page for the repository `arrowtype/recursive`. The issue is titled "Current full kerning seems to be blocking build. AssertionError: ('Lookup', '[0]', 'list', '.Lookup', 'LookupList', '.LookupList', 'GPOS', '.table', 'table\_G\_P\_O\_S\_') #213". The issue was opened by `arrowtype` on Nov 4, 2019, and has 7 comments. A note from `arrowtype` states: "When I try my latest sources for a build, I am running into the following error: File "/Users/stephennixon/type-repos/recursive/venv/lib/python3.7/site-packages/fontTools/varLib/merger.py", line 11 assert all(sorted(vs, key=sortKey) == vs for vs in lst) AssertionError: ('Lookup', '[0]', 'list', '.Lookup', 'LookupList', '.LookupList', 'GPOS', '.table', 'table\_G\_P\_O\_S\_')". Below this, a list of things tried to fix the issue includes: "I have prepped sources with varfont-prep, and I'm building from these", "I have run my script to assign unicodes to glyphs in all sources", "I have run my script to sort all sources in the same way", "I have run my script to add kerning if none is present (src/00-recursive-scripts-for-robofont/kerning/add-empty-kerning-if-none.py)", and "I have checked for common anchors". The right sidebar shows fields for Assignees (No one assigned), Labels (None yet), Projects (None yet), Milestone (No milestone), and Linked pull requests (Successfully merging a pull request may close).

Current full kerning seems to be blocking build. `AssertionError: ('Lookup', '[0]', 'list', '.Lookup', 'LookupList', '.LookupList', 'GPOS', '.table', 'table_G_P_O_S_')` #213

`arrowtype` commented on Nov 4, 2019 · edited

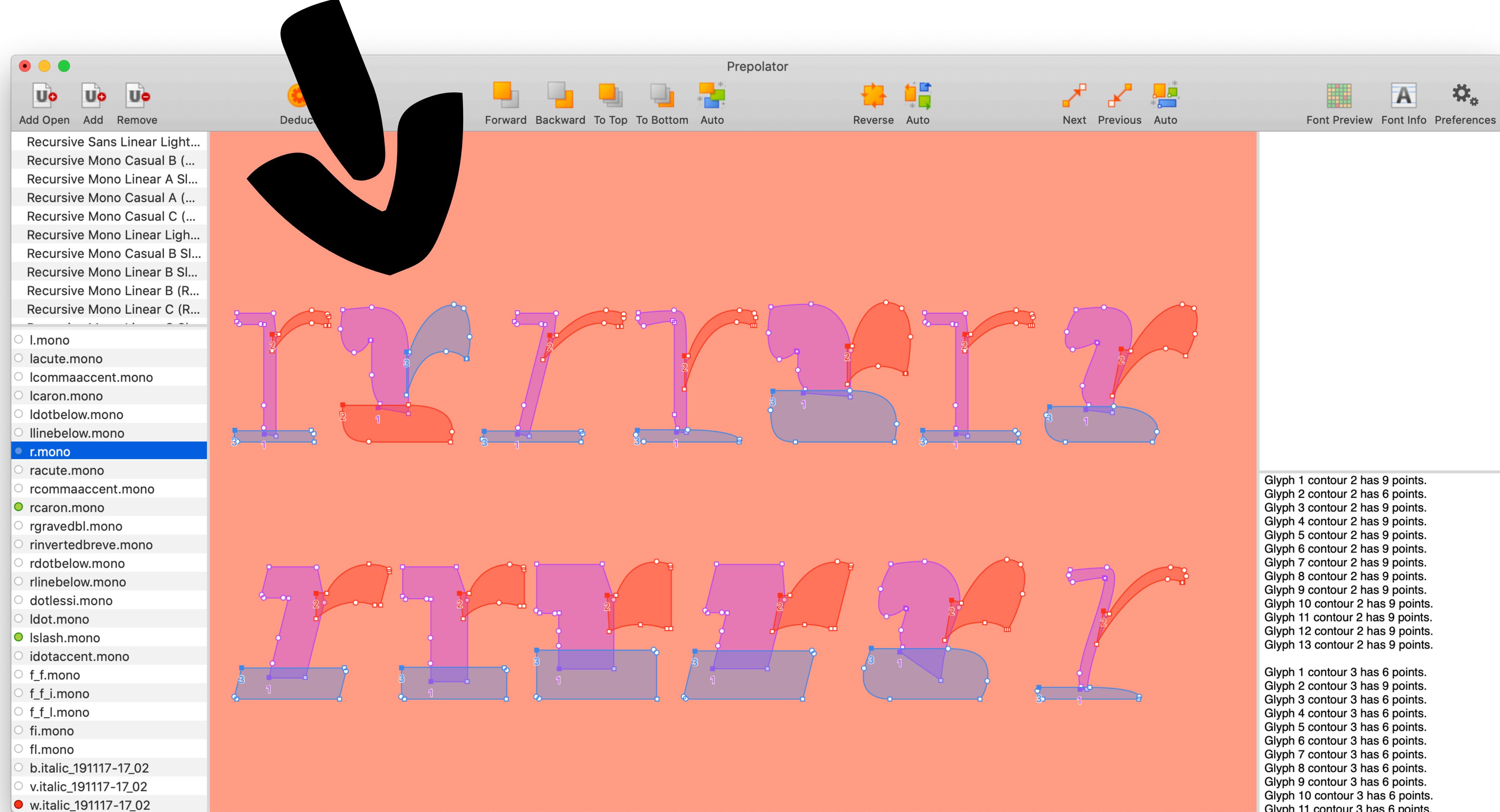
When I try my latest sources for a build, I am running into the following error:

```
File "/Users/stephennixon/type-repos/recursive/venv/lib/python3.7/site-packages/fontTools/varLib/merger.py", line 11
    assert all(sorted(vs, key=sortKey) == vs for vs in lst)
AssertionError: ('Lookup', '[0]', 'list', '.Lookup', 'LookupList', '.LookupList', 'GPOS', '.table', 'table_G_P_O_S_')
```

Things I have tried in an attempt to fix this:

- I have prepped sources with varfont-prep, and I'm building from these
- I have run my script to assign unicodes to glyphs in all sources
- I have run my script to sort all sources in the same way
- I have run my script to add kerning if none is present (src/00-recursive-scripts-for-robofont/kerning/add-empty-kerning-if-none.py)
- I have checked for common anchors

# **Compatibility fixing with Prepolator**

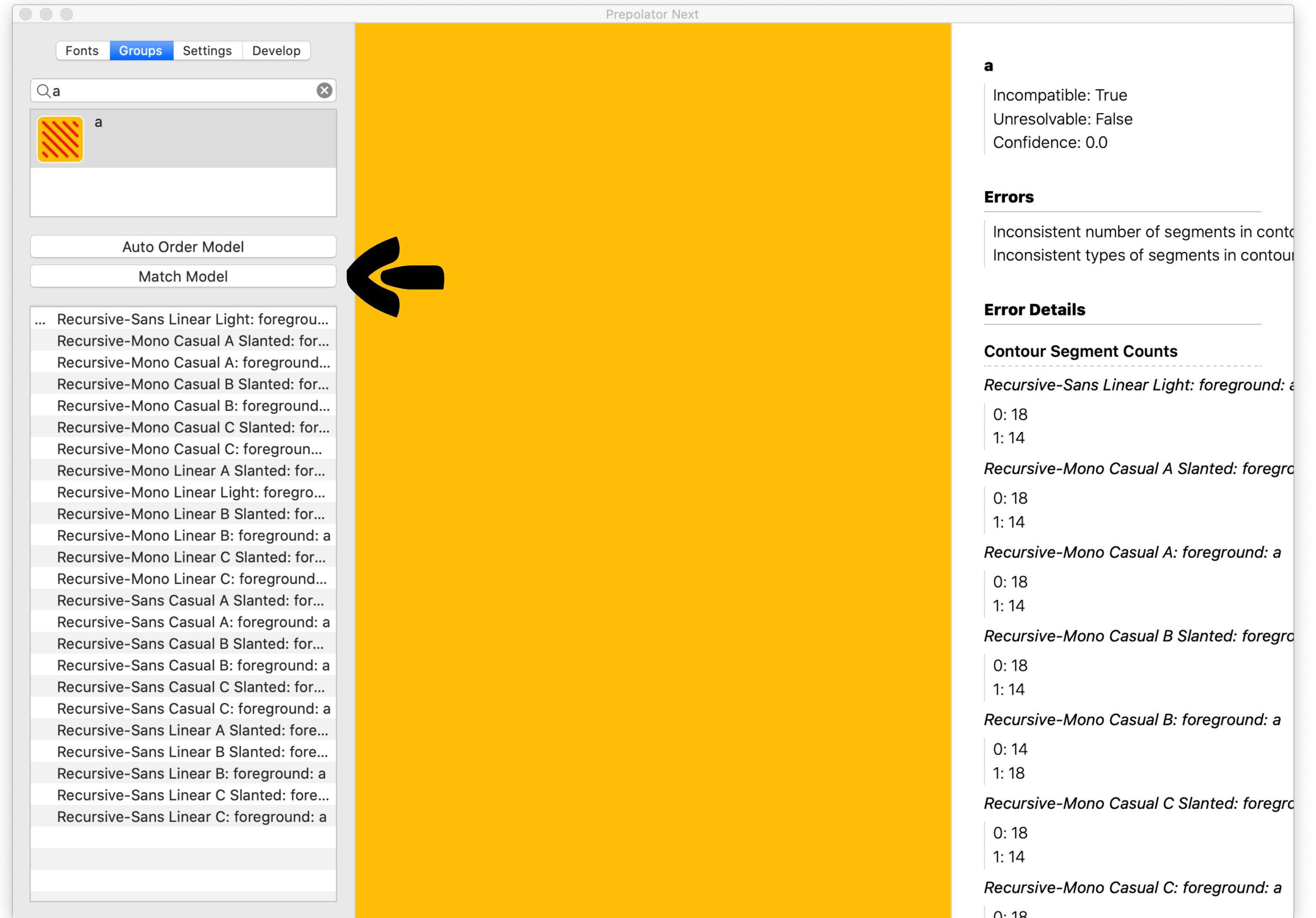


Prepolator

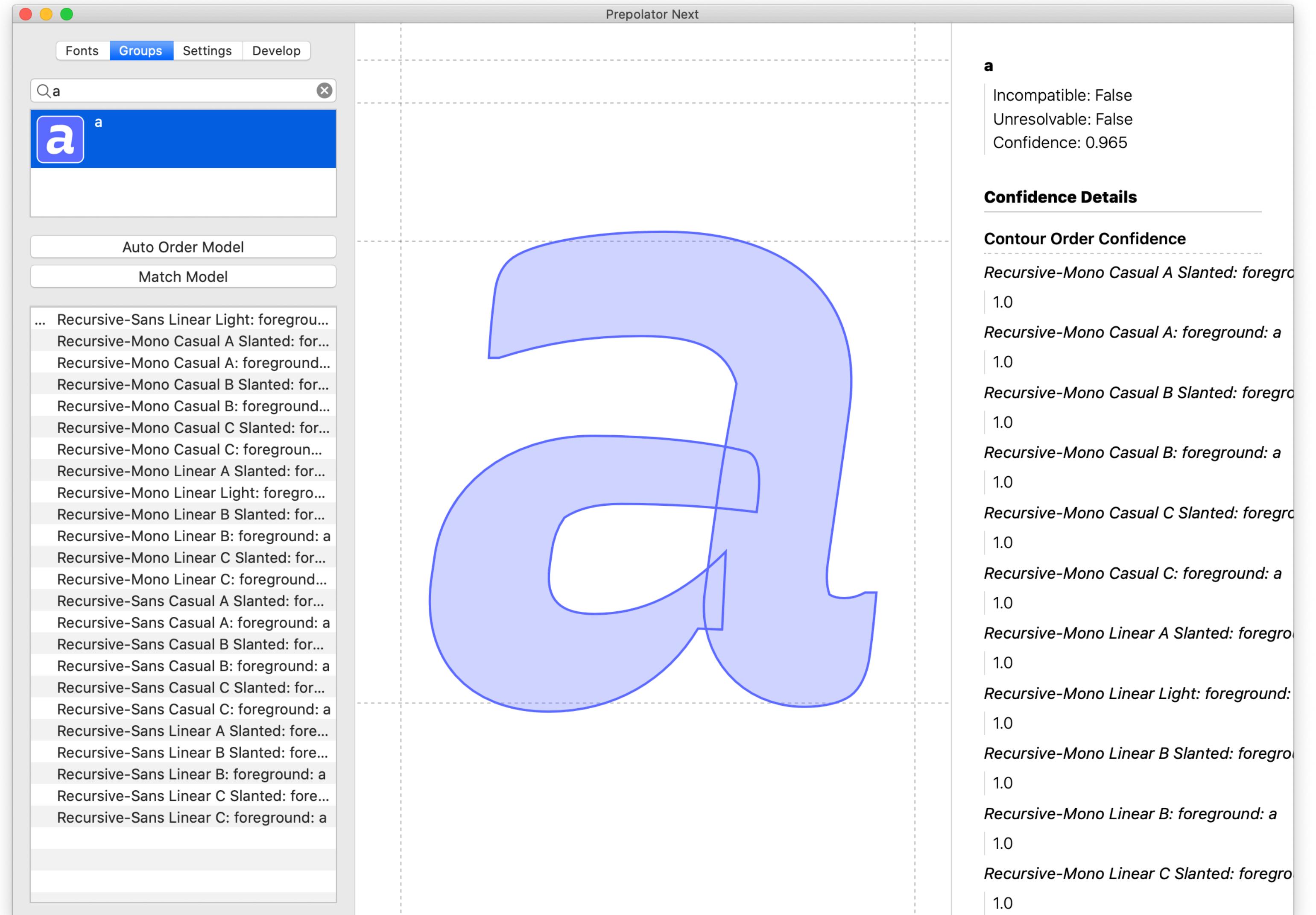
Prepolator

The screenshot shows the Prepolator application interface. The top menu bar includes standard Mac OS X icons (red, yellow, green) and buttons for 'Add Open', 'Add', 'Remove', 'Deduce', 'Match', 'Forward', 'Backward', 'To Top', 'To Bottom', 'Auto', 'Reverse', 'Auto', 'Next', 'Previous', 'Auto', 'Font Preview', 'Font Info', and 'Preferences'. On the left, a sidebar lists various font and character names with corresponding colored circles: Recursive Sans Linear Light..., Recursive Mono Casual B (...), Recursive Mono Linear A Sl..., Recursive Mono Casual A (...), Recursive Mono Casual C (...), Recursive Mono Linear Ligh..., Recursive Mono Casual B Sl..., Recursive Mono Linear B Sl..., Recursive Mono Linear B (R...), Recursive Mono Linear C (R...), r.mono (blue dot), racute.mono (red dot), rcommaaccent.mono (white circle), rcaron.mono (white circle), rgravedbl.mono (white circle), rinvertedbreve.mono (white circle), rdotbelow.mono (white circle), rlinebelow.mono (white circle), dotlessi.mono (white circle), ldot.mono (white circle), lslash.mono (green dot), idotaccent.mono (white circle), f\_f.mono (white circle), f\_f\_i.mono (white circle), f\_f\_l.mono (white circle), fi.mono (white circle), fl.mono (white circle), b.italic\_191117-17\_02 (white circle), v.italic\_191117-17\_02 (white circle), w.italic\_191117-17\_02 (red dot), mu.math (yellow dot), Q.titl (yellow dot), g.compact\_desc (white circle), hyphen.code (white circle), numbersign.code (white circle), quotesingle.code (white circle). The main workspace displays two sets of font outlines for the character 'y'. The top set shows a standard 'y' with stroke segments highlighted in purple and red. The bottom set shows a more complex, italicized 'y' with similar stroke highlights. To the right of the workspace is a preview window showing a large, bold black 'y' character. Below the preview is the text 'Start point positions are questionable.'

Prepolator



Prepolator Next  
(pre-release)



Prepolator Next  
(pre-release)

**Setting font info  
in one place**

# **"Why do I have to set font name, metrics, etc, in each UFO of a type family?"**

***Tal Leming:***

This is a very valid point of criticism of UFO. ...If you can find the ancient alpha RoboFab code (circa 2003), you'll see a bunch of code to generate a UFF (Unified Font Family). We dumped it right at the beginning because the ideas were too vague. ...

UFO works because it is very flexible: you can use the format to store data for PostScript fonts, TrueType fonts, OTF-CFF fonts, OTF-TTF fonts, SVG <font> elements, OTF fonts with color SVG outlines and on and on. This has been the point of debate for "multiple styles in a single file" since the beginning: how do we make it flexible?

gode 21e0a8875

Tilt-Typeface / tools / Font Info / SetFontInfo.py / <> Jump to ▾

Go to file

...



andyclymer Tool and script reorganization, updated font info, new build

Latest commit ab8c04b on Mar 27, 2020 ⏲ History

1 contributor

113 lines (87 sloc) | 3.94 KB

Raw Blame



```
1 import os
2
3
4 def setFontInfo(f, familyName, styleName, version=(0, 0), versionString ""):
5
6     infoDict = dict(
7
8         familyName = familyName,
9         styleName = styleName,
10        styleMapFamilyName = "%s %s" % (familyName, styleName),
11        styleMapStyleName = "regular",
12
13        openTypePreferredFamilyName = familyName,
14        openTypePreferredSubfamilyName = styleName,
15        openTypeCompatibleFullName = "%s %s" % (familyName, styleName),
16
17        versionMajor = version[0],
18        versionMinor = version[1],
19        openTypeNameVersion = "%s.%s %s" % (version[0], version[1], versionString),
20        year = 2019,
21        copyright = "Copyright 2020 Andy Clymer",
22        trademark = "",
23
24        unitsPerEm = 1000,
25        descender = -160,
26        xHeight = 517,
27        capHeight = 680,
28        ascender = 730,
29        italicAngle = None,
30
31        openTypeOS2TypoAscender = 990,
32        openTypeOS2TypoDescender = -272,
33        openTypeOS2TypoLineGap = 0,
```

SetFontInfo.py for Tilt by Andy Clymer

# **Designing interpolations with Skateboard**

Prepolator Next

Fonts Groups Settings Develop

r.mono

r.mono

Kink?

**r.mono**

Incompatible: False  
Unresolvable: False  
Confidence: 0.946

**Confidence Details**

**Contour Order Confidence**

*Recursive-Mono Casual A Slanted: foreground: r.mono*  
1.0

*Recursive-Mono Casual A: foreground: r.mono*  
1.0

*Recursive-Mono Casual B Slanted: foreground: r.mono*  
0.938

*Recursive-Mono Casual B: foreground: r.mono*  
1.0

*Recursive-Mono Casual C Slanted: foreground: r.mono*  
1.0

*Recursive-Mono Casual C: foreground: r.mono*  
0.938

*Recursive-Mono Linear A Slanted: foreground: r.mono*  
0.938

*Recursive-Mono Linear B Slanted: foreground: r.mono*  
1.0

*Recursive-Mono Linear B: foreground: r.mono*  
1.0

*Recursive-Mono Linear C Slanted: foreground: r.mono*  
1.0

*Recursive-Mono Linear C: foreground: r.mono*  
1.0

**Start Segment Confidence**

*Recursive-Mono Casual A Slanted: foreground: r.mono*  
0: 0.867

recursive-MONO\_CASL\_wght\_slnt\_ital--full\_gsub.designspace

**Editor**

**Sources & Layers**

**UFO** Layer

- (U) Recursive Mono-Casual A Slanted.ufo
- (U) Recursive Mono-Casual A.ufo
- (U) Recursive Mono-Casual B Slanted.ufo
- (U) Recursive Mono-Casual B.ufo
- (U) Recursive Mono-Casual C Slanted.ufo
- (U) Recursive Mono-Casual C.ufo
- (U) Recursive Mono-Linear A Slanted.ufo
- (U) Recursive Mono-Linear A.ufo
- (U) Recursive Mono-Linear B Slanted.ufo
- (U) Recursive Mono-Linear B.ufo
- (U) Recursive Mono-Linear C Slanted.ufo
- (U) Recursive Mono-Linear C.ufo
- (U) Recursive Sans-Casual A Slanted.ufo
- (U) Recursive Sans-Casual A.ufo
- (U) Recursive Sans-Casual B Slanted.ufo
- (U) Recursive Sans-Casual B.ufo
- (U) Recursive Sans-Casual C Slanted.ufo
- (U) Recursive Sans-Casual C.ufo

**Preview**

Name	Designspace	Userspace	Normalised	Interaction	
Monospace	1.000	1.0000	ignore	▼	
Casual	0.500	0.5000	ignore	▼	
Weight	126.581	674.716	0.4947	vertical	▼
Slant	-7.637	-7.149	-0.5091	horizontal	▼
Cursive	0.500	0.0000	ignore	▼	

**Locations**

Locations	Sources	Supports	Instances	Interesting	Open
Recursive Mono Linear Light		Instance			
Recursive Mono Linear Light Italic		Instance			
Recursive Mono Casual Light		Instance			
Recursive Mono Casual Light Italic		Instance			
Recursive Mono Linear		Instance			
Recursive Mono Linear Italic		Instance			
Recursive Mono Casual		Instance			
Recursive Mono Casual Italic		Instance			
Recursive Mono Linear Medium		Instance			
Recursive Mono Linear Medium Italic		Instance			
Recursive Mono Casual Medium		Instance			
Recursive Mono Casual Medium Italic		Instance			
Recursive Mono Linear SemiBold		Instance			
Recursive Mono Linear SemiBold Italic		Instance			
Recursive Mono Casual SemiBold		Instance			
Recursive Mono Casual SemiBold Italic		Instance			
Recursive Mono Linear Bold		Instance			
Recursive Mono Linear Bold Italic		Instance			

v 1.6.3

⌘Skateboard Text

# arrow

a/r.mono /r.mono ow

Text Glyph Positive Negative

opsz

⌘Skateboard Space

Monospace 0.000  
Casual 1.000  
Weight 215.000  
Slant -15.000  
Cursive 1.000  
Recursive Sans-Casual ExtraBlack Italic.ufo  
Slant -15.000  
Cursive 1.000  
Recursive Sans-Casual Black Italic.ufo  
Monospace 0.000  
sene a Casual 1.000  
condit Weight 146.000  
Slant -15.000  
Cursive 1.000  
Recursive Sans-Casual ExtraBold Italic.ufo  
Cursive 1.000  
Recursive Sans-Casual Bold Italic.ufo  
Cursive 1.000  
Recursive Sans-Casual SemiBold Italic.ufo  
Cursive 1.000  
Recursive Sans-Casual Medium Italic.ufo  
Cursive 1.000  
Recursive Sans-Casual Italic.ufo  
Monospace 0.000  
Casual 1.000  
Weight 40.000  
Slant -15.000  
Cursive 1.000  
Recursive Sans-Casual Light Italic.ufo

Show labels

**Sources**

Recursive Mono-Casual A Slanted.ufo  
Recursive Mono-Casual A.ufo  
Recursive Mono-Casual B Slanted.ufo  
Recursive Mono-Casual B.ufo  
Recursive Mono-Casual C Slanted.ufo  
Recursive Mono-Casual C.ufo  
Recursive Mono-Linear A Slanted.ufo  
Recursive Mono-Linear A.ufo  
Recursive Mono-Linear B Slanted.ufo  
Recursive Mono-Linear B.ufo  
Recursive Mono-Linear C Slanted.ufo  
Recursive Mono-Linear C.ufo  
Recursive Sans-Casual A Slanted.ufo  
Recursive Sans-Casual A.ufo  
Recursive Sans-Casual B Slanted.ufo  
Recursive Sans-Casual B.ufo  
Recursive Sans-Casual C Slanted.ufo  
Recursive Sans-Casual C.ufo

**Axes**

Name	Designspace	Userspace	Normalised	Interaction
Monospace	1.000	1.0000	ignore	▼
Casual	0.500	0.5000	ignore	▼
Weight	126.581	674.716	0.4947	vertical
Slant	-7.637	-7.149	-0.5091	horizontal
Cursive	0.500	0.0000	ignore	▼

**Locations**

Locations	Sources	Supports	Instances	Interesting	Open
Recursive Mono Linear Light	Instance				
Recursive Mono Linear Light Italic	Instance				
Recursive Mono Casual Light	Instance				
Recursive Mono Casual Light Italic	Instance				
Recursive Mono Linear	Instance				
Recursive Mono Linear Italic	Instance				
Recursive Mono Casual	Instance				
Recursive Mono Casual Italic	Instance				
Recursive Mono Linear Medium	Instance				
Recursive Mono Linear Medium Italic	Instance				
Recursive Mono Casual Medium	Instance				
Recursive Mono Casual Medium Italic	Instance				
Recursive Mono Linear SemiBold	Instance				
Recursive Mono Linear SemiBold Italic	Instance				
Recursive Mono Casual SemiBold	Instance				
Recursive Mono Casual SemiBold Italic	Instance				
Recursive Mono Linear Bold	Instance				
Recursive Mono Linear Bold Italic	Instance				

**Text Window**

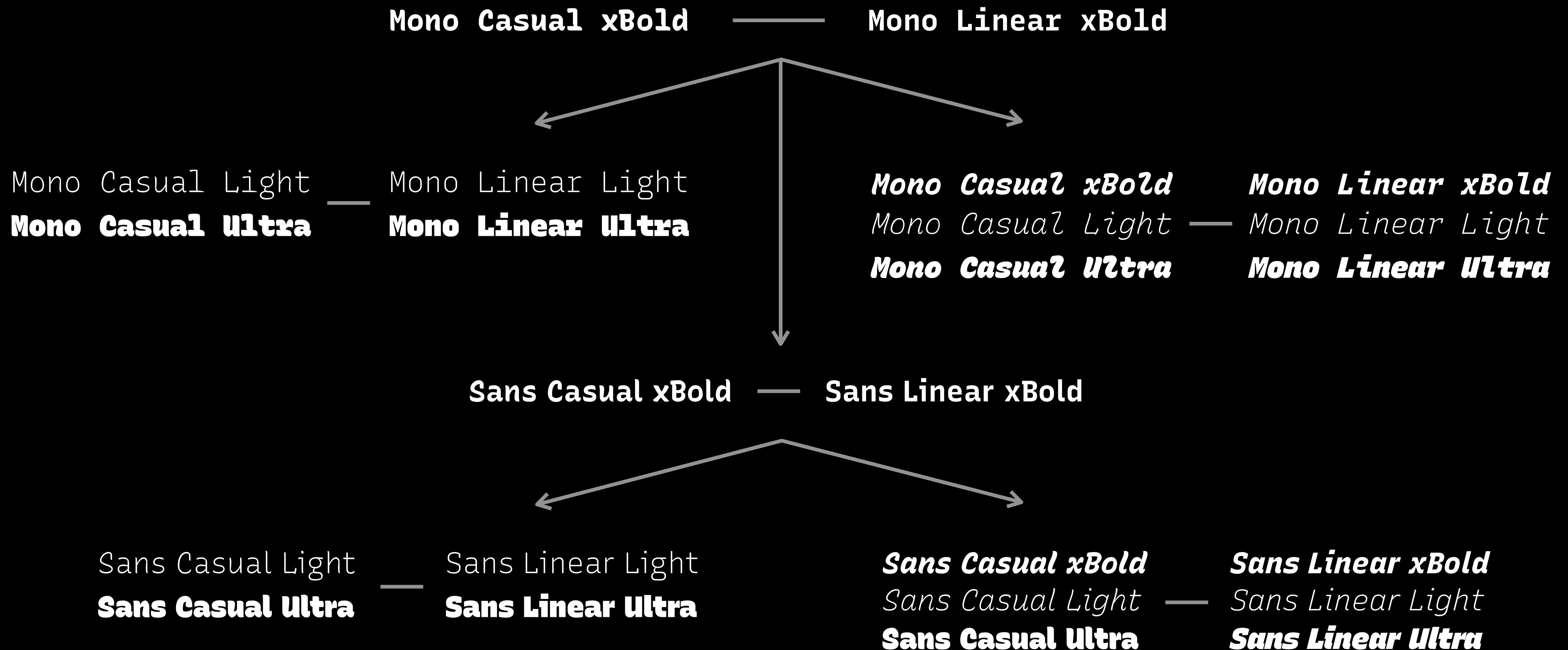
arrow

**Space Window**

Space window showing a grid of font variations for the word "Space". The variations are organized into a 4x4 grid. The columns represent different weights (Monospace 0.000, Casual 1.000, Weight 215.000, Slant 0.000) and the rows represent different slants (Cursive 1.000, Recursive Sans-Casual ExtraBlack Italic.ufo, Slant -15.000, Cursive 1.000). The word "Space" is rendered in a bold italic style.

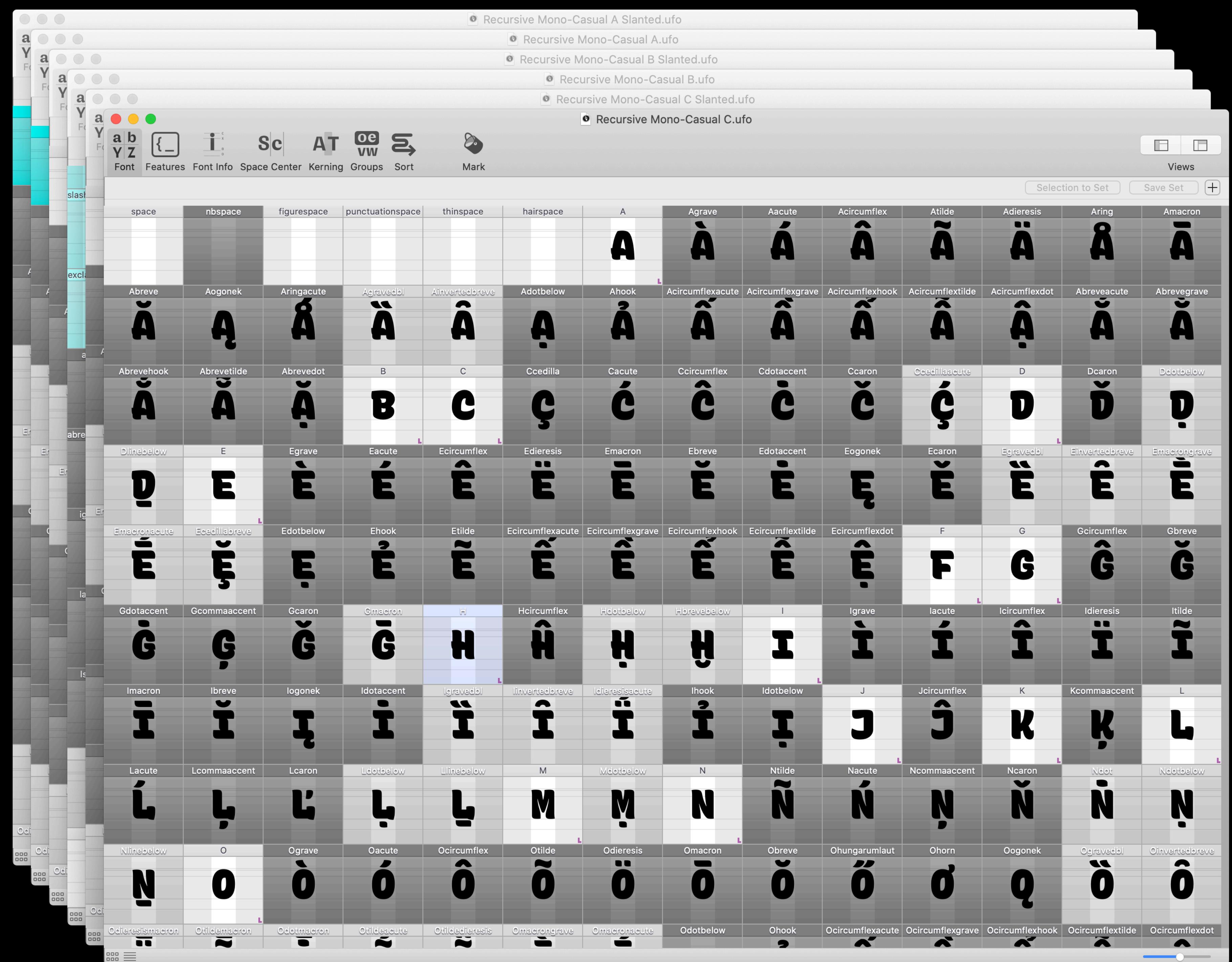
**Working with  
a lot of sources**

## ***DESIGN WORKFLOW (SORT OF)***



## **Strategy 1: Work on a design in one area at a time**

- I typically work on about 1 to 6 open UFOs at a time in RoboFont.
- For my laptop, opening more than that can start to slow down a bit.
- (For my brain, working on more than that can also start to slow down a bit.)
- I make partial designspaces to explore & refine logical subsets of the project in Skateboard.



## **Strategy 2: Open all sources in Simple Font Windows**

- This allows you to open many UFOs at once with very little slow down.
- Useful if you want to do something like manually tweaking the location of an anchor in a glyph across all sources of a large project, or checking point indexes manually (if automatic tools aren't working for some reason)

Select a font:

ufo

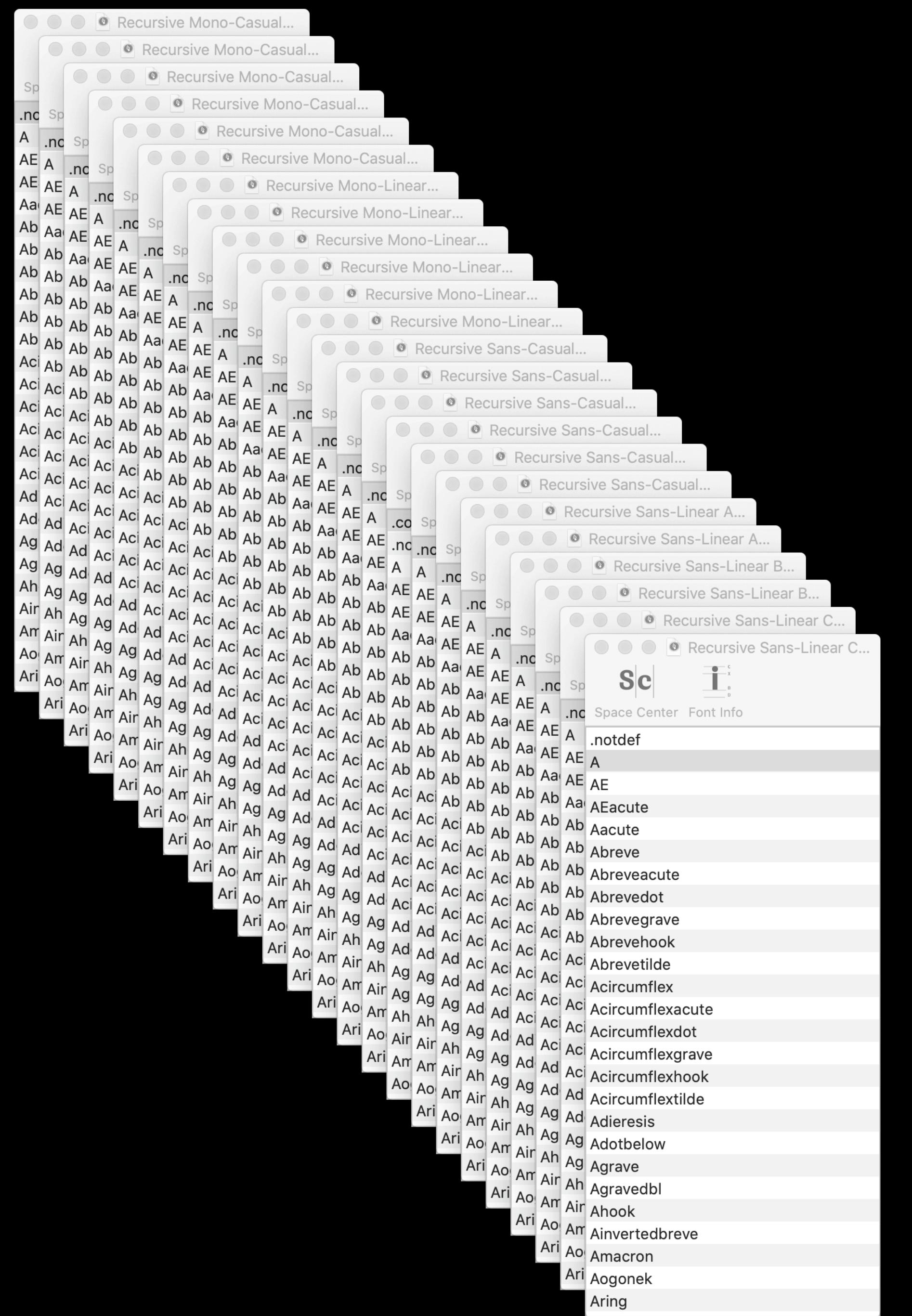
Search

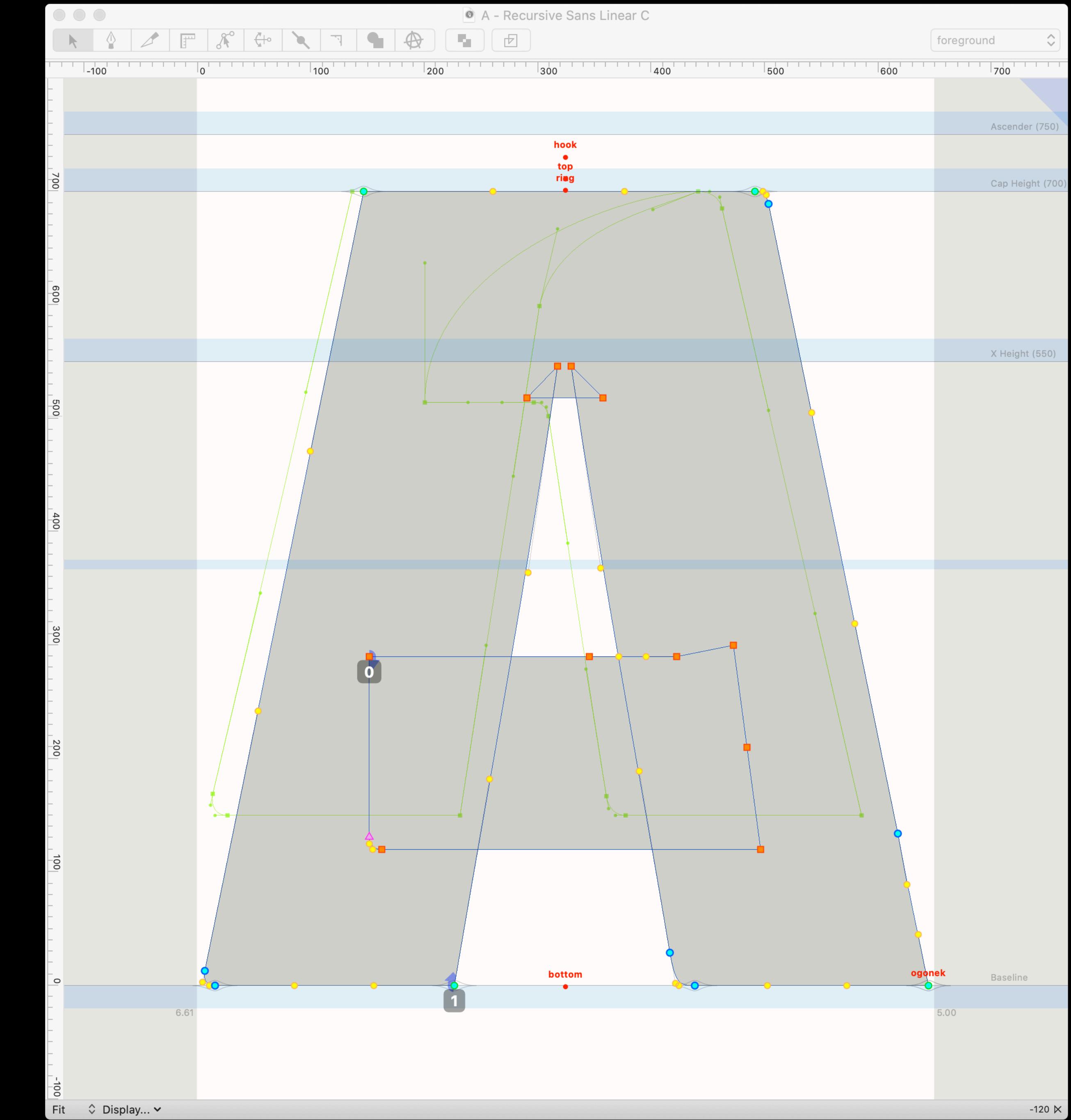
Name	Date Modified	Size	Kind	Date Added
► extras	Oct 6, 2020 at 2:14 PM	--	Folder	Oct 1, 2020 at 5:58 PM
▼ mono	Sep 21, 2020 at 10:55 AM	--	Folder	Apr 4, 2020 at 5:24 PM
► designspaces--experimental	Aug 18, 2020 at 12:06 PM	--	Folder	Aug 18, 2020 at 12:06 PM
► designspaces--partial	Aug 18, 2020 at 12:06 PM	--	Folder	Aug 18, 2020 at 12:06 PM
gsub-rules.txt	Aug 18, 2020 at 12:06 PM	18 KB	Plain Text	Aug 18, 2020 at 12:06 PM
Ag Recursive Mono-Casual A Slanted.ufo	Mar 12, 2021 at 4:08 PM	5.1 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Mono-Casual A.ufo	Mar 12, 2021 at 4:08 PM	3.7 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Mono-Casual B Slanted.ufo	Mar 12, 2021 at 4:08 PM	3.1 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Mono-Casual B.ufo	Mar 24, 2021 at 5:32 PM	4 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Mono-Casual C Slanted.ufo	Mar 12, 2021 at 4:08 PM	2.6 MB	Unified...Object	Apr 4, 2020 at 5:24 PM
Ag Recursive Mono-Casual C.ufo	Mar 12, 2021 at 4:08 PM	3.1 MB	Unified...Object	Apr 4, 2020 at 5:24 PM
Ag Recursive Mono-Linear A Slanted.ufo	Mar 12, 2021 at 4:08 PM	2.9 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Mono-Linear A.ufo	Mar 12, 2021 at 4:08 PM	3.6 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Mono-Linear B Slanted.ufo	Mar 12, 2021 at 4:08 PM	3.2 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Mono-Linear B.ufo	Mar 12, 2021 at 4:08 PM	4 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Mono-Linear C Slanted.ufo	Mar 12, 2021 at 4:08 PM	2.6 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Mono-Linear C.ufo	Mar 12, 2021 at 4:08 PM	3 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
recursive_mono.designspace	Aug 18, 2020 at 12:06 PM	31 KB	Design...cument	Aug 18, 2020 at 12:06 PM
recursive-MONO_CASL_wght_slnt_ital--full_gsub.designspace	Jan 21, 2021 at 6:19 PM	72 KB	Design...cument	Jan 21, 2021 at 6:19 PM
▼ sans	Jan 29, 2021 at 7:28 PM	--	Folder	Apr 4, 2020 at 5:24 PM
▼ kerning	Aug 18, 2020 at 12:06 PM	--	Folder	Aug 18, 2020 at 12:06 PM
groups.plist	Mar 24, 2021 at 11:38 AM	80 KB	Property List	Aug 18, 2020 at 12:06 PM
Recursive Sans-Casual C Kernfea	Aug 18, 2020 at 12:06 PM	3 KB	Feature File	Aug 18, 2020 at 12:06 PM
Recursive Sans-kerning_groups.mmg	Aug 18, 2020 at 12:06 PM	6 KB	Document	Aug 18, 2020 at 12:06 PM
Recursive Sans-kerning_super_basic_pairs.txt	Aug 18, 2020 at 12:06 PM	132 bytes	Plain Text	Aug 18, 2020 at 12:06 PM
Recursive Sans-Linear C kernfea	Aug 18, 2020 at 12:06 PM	3 KB	Feature File	Aug 18, 2020 at 12:06 PM
► partial-designspaces	Mar 24, 2021 at 12:00 PM	--	Folder	Aug 18, 2020 at 12:06 PM
Ag Recursive Sans-Casual A Slanted.ufo	Mar 12, 2021 at 4:08 PM	2.1 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Sans-Casual A.ufo	Mar 12, 2021 at 4:08 PM	2.1 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Sans-Casual B Slanted.ufo	Mar 12, 2021 at 4:08 PM	2 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Sans-Casual B.ufo	Mar 12, 2021 at 4:08 PM	2.2 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Sans-Casual C Slanted.ufo	Mar 24, 2021 at 2:37 PM	2.3 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Sans-Casual C.ufo	Mar 24, 2021 at 2:37 PM	2.2 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Sans-Linear A Slanted.ufo	Mar 12, 2021 at 4:08 PM	2 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Sans-Linear A.ufo	Mar 24, 2021 at 2:37 PM	2.1 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Sans-Linear B Slanted.ufo	Mar 12, 2021 at 4:08 PM	2 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Sans-Linear B.ufo	Mar 12, 2021 at 4:08 PM	2.2 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Sans-Linear C Slanted.ufo	Mar 12, 2021 at 4:08 PM	2.3 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
Ag Recursive Sans-Linear C.ufo	Mar 12, 2021 at 4:08 PM	2.3 MB	Unified...Object	Aug 18, 2020 at 12:06 PM
recursive_sans.designspace	Aug 18, 2020 at 12:06 PM	29 KB	Design...cument	Aug 18, 2020 at 12:06 PM

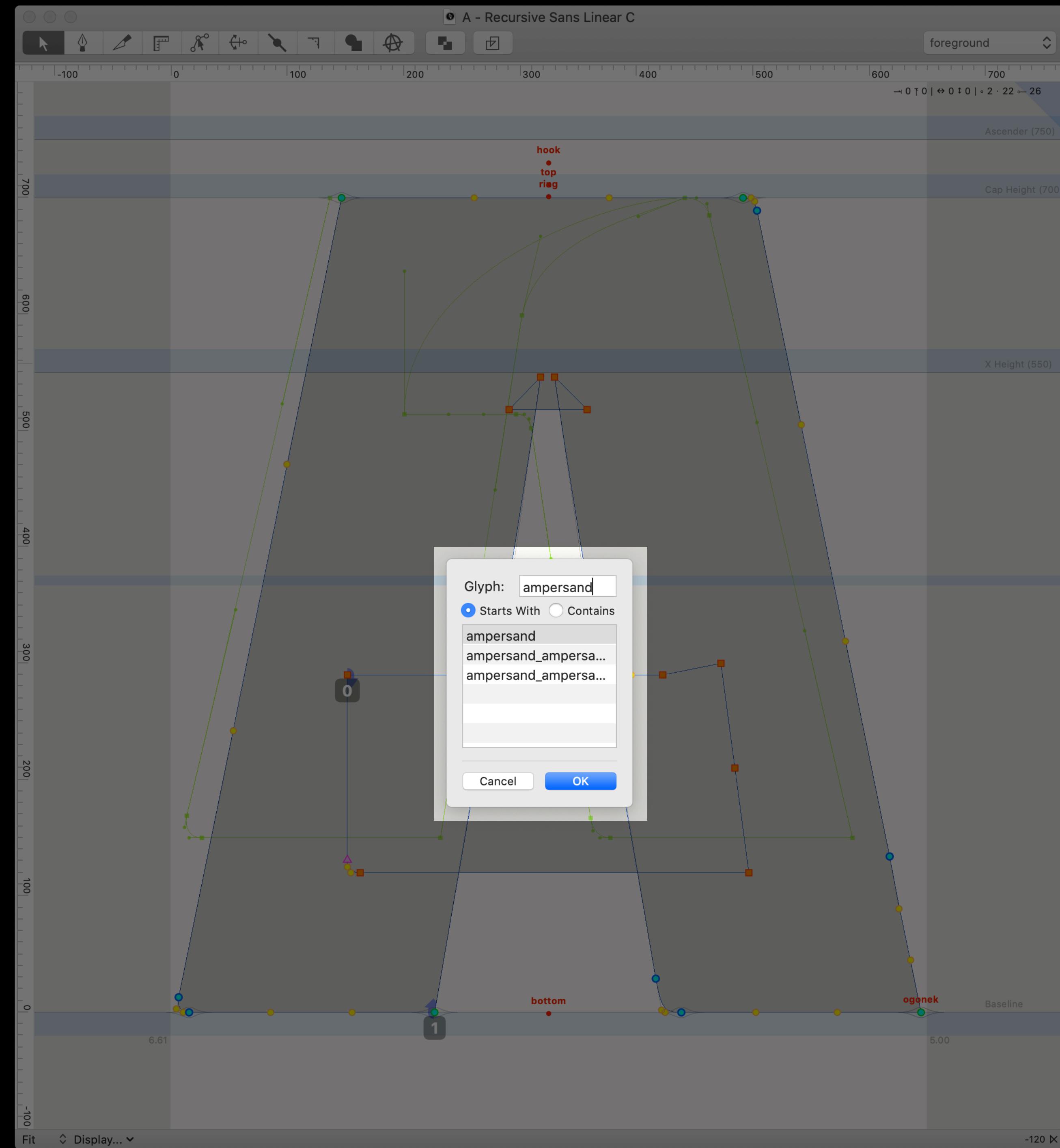
Cancel

Open

# Simple Font Window







**Q&A**

*Thanks!*

@