

# Creating a Clean OTF font file

A quick hold hands guide



# Setting up in Fontforge

# **'Font Info' window**

In Fontforge these parameters are held in the 'Font Info' window, found under the 'Element' menu.

FontForge 2.0

PS Names

General

Layers

PS UID

PS Private

OS/2

TTF Names

StyleSet Names

Grid Fitting

T&X

Size

Comment

FONTLOG

Mark Classes

Mark Sets

Lookups

WOFF

Mac

Mac Features

Dates

Unicode Ranges

Fontname: Untitled1

Family Name: Untitled1

Name For Humans: Untitled1

Weight: Medium

Version: 001.000

sfnt Revision:

☒ Same as Fontname

Base Filename:

Copyright:

Created by vernon adams with FontForge 2.0 (<http://fontforge.sf.net>)

OK Cancel

# Naming your Font

This is not "*what shall i call it?*", this is about making sure that whatever you do name your font shows up properly in font menus, and correctly on all Operating Systems.

We add the names to our font in the 'PS Names' section of our Font Info.

Font Properties dialog box (X)

**PS Names** (selected)

General  
Layers  
PS UID  
PS Private  
OS/2  
TTF Names  
StyleSet Names  
Grid Fitting  
T&X  
Size  
Comment  
FONTLOG  
Mark Classes  
Mark Sets  
Lookups  
WOFF  
Mac  
Mac Features  
Dates  
Unicode Ranges

Fontname: Mynewfont-Bold

Family Name: Mynewfont

Name For Humans: Mynewfont Bold

Weight: Bold

Version: 1

sfnt Revision:

Base Filename: ☒ Same as Fontname ☐

Copyright:  
2011 (c) vernon adams

OK Cancel

# Dimensions Settings

Next: the 'General' section of the Font Info window.



PS Names

**General**

Layers

PS UID

PS Private

OS/2

TTF Names

StyleSet Names

Grid Fitting

T&X

Size

Comment

FONTLOG

Mark Classes

Mark Sets

Lookups

WOFF

Mac

Mac Features

Dates

Unicode Ranges

Ascent:

800

Descent:

200

Em Size:

1000



Scale Outlines

Itallic Angle:

0

Guess

Underline Position:

-100

Height:

50



Has Vertical Metrics

Interpretation:

None



Name List:

AGL For New Fonts



OK

Cancel



# The UPM or 'em size'

UPM = *"Units per Em"*

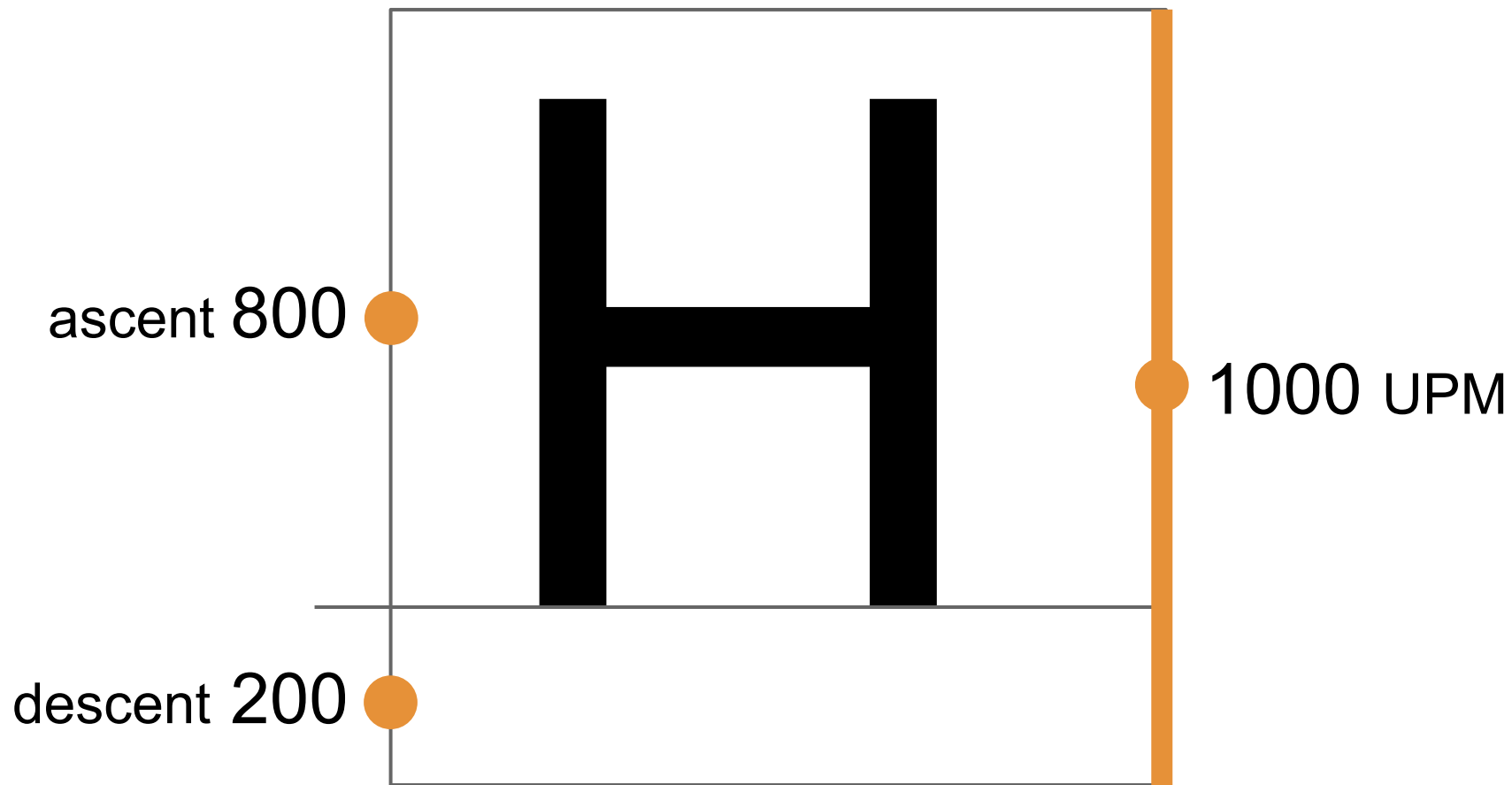
Traditionally the em was equal to the width of the uppercase 'M' but in the digital age, in a typical Open Type postscript font, the UPM size, or the em, is usually set at **1000** units.

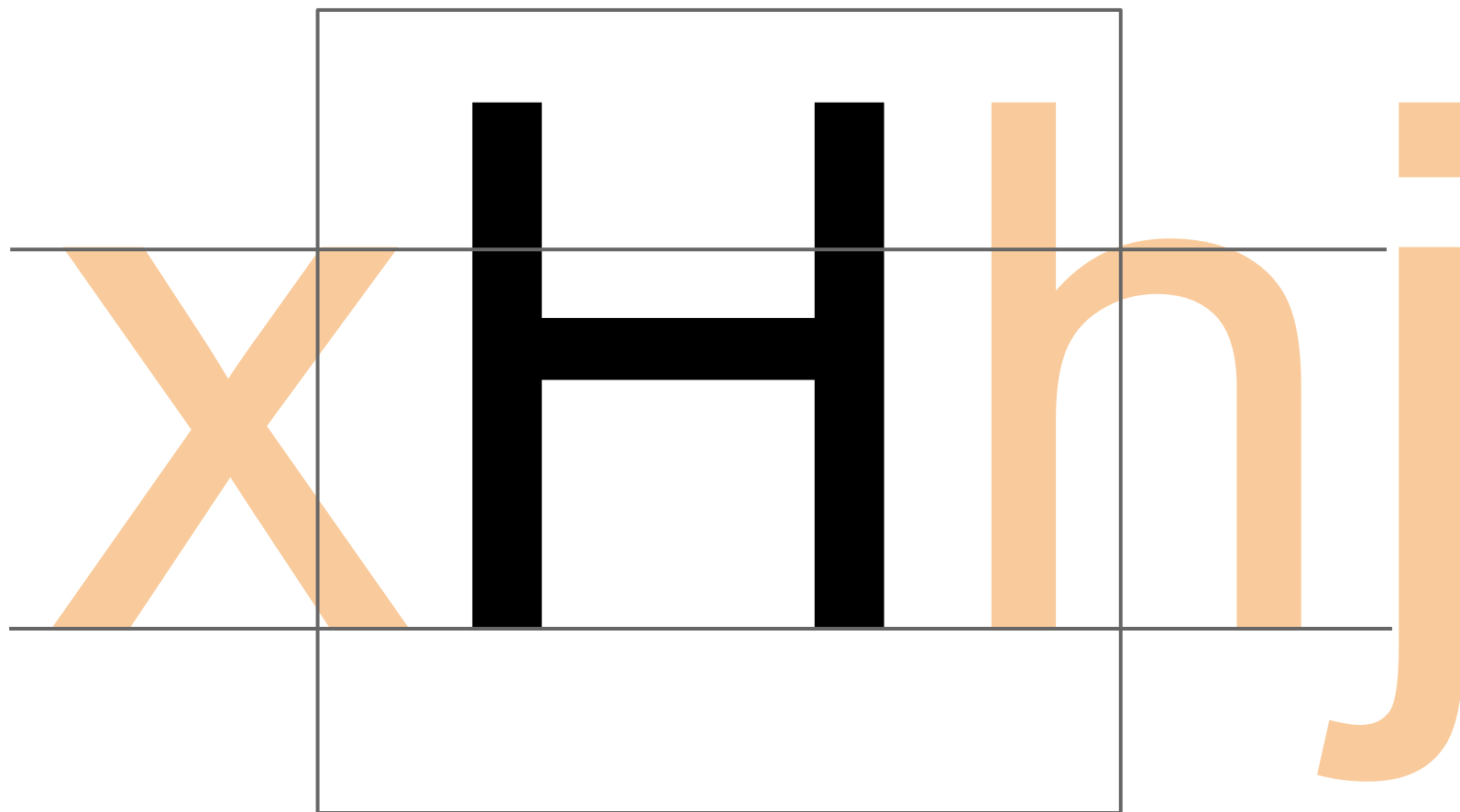
When the font is used to set type, the em, i.e. the 1000 units, are scaled to the desired point size. This means that at 10 pt type, the 1000 font units are scaled to 10 pt.

So if your uppercase H

is 700 units high,

it will be 7 pt high at 10 pt type.

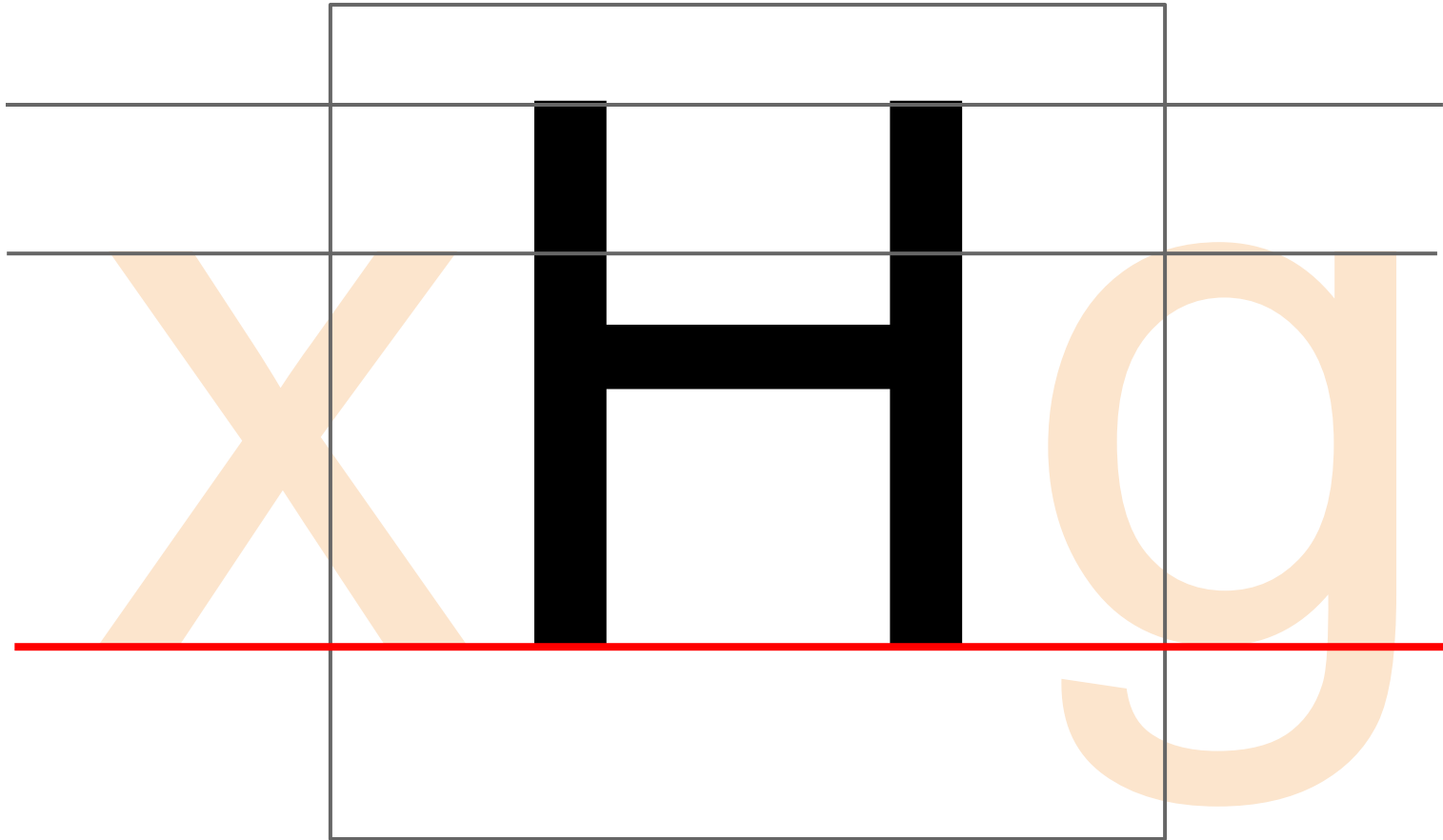




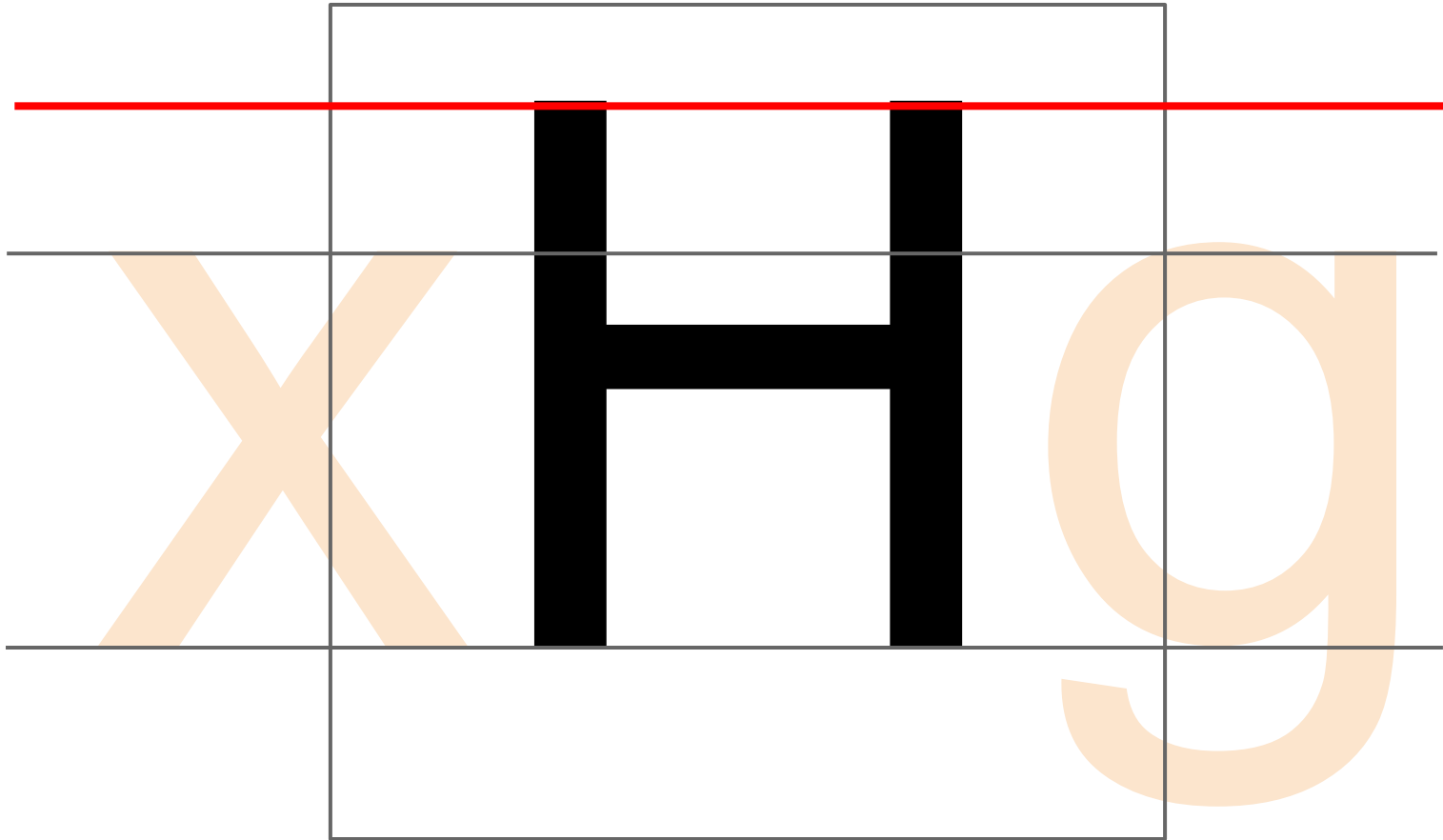
# Setting up in Glyph Window

With the knowledge that your font is using a 1000 UPM, you need to set up the drawing of your glyphs to ensure that aspects of your typeface fit adequately into that 1000 UPM.

# the Baseline

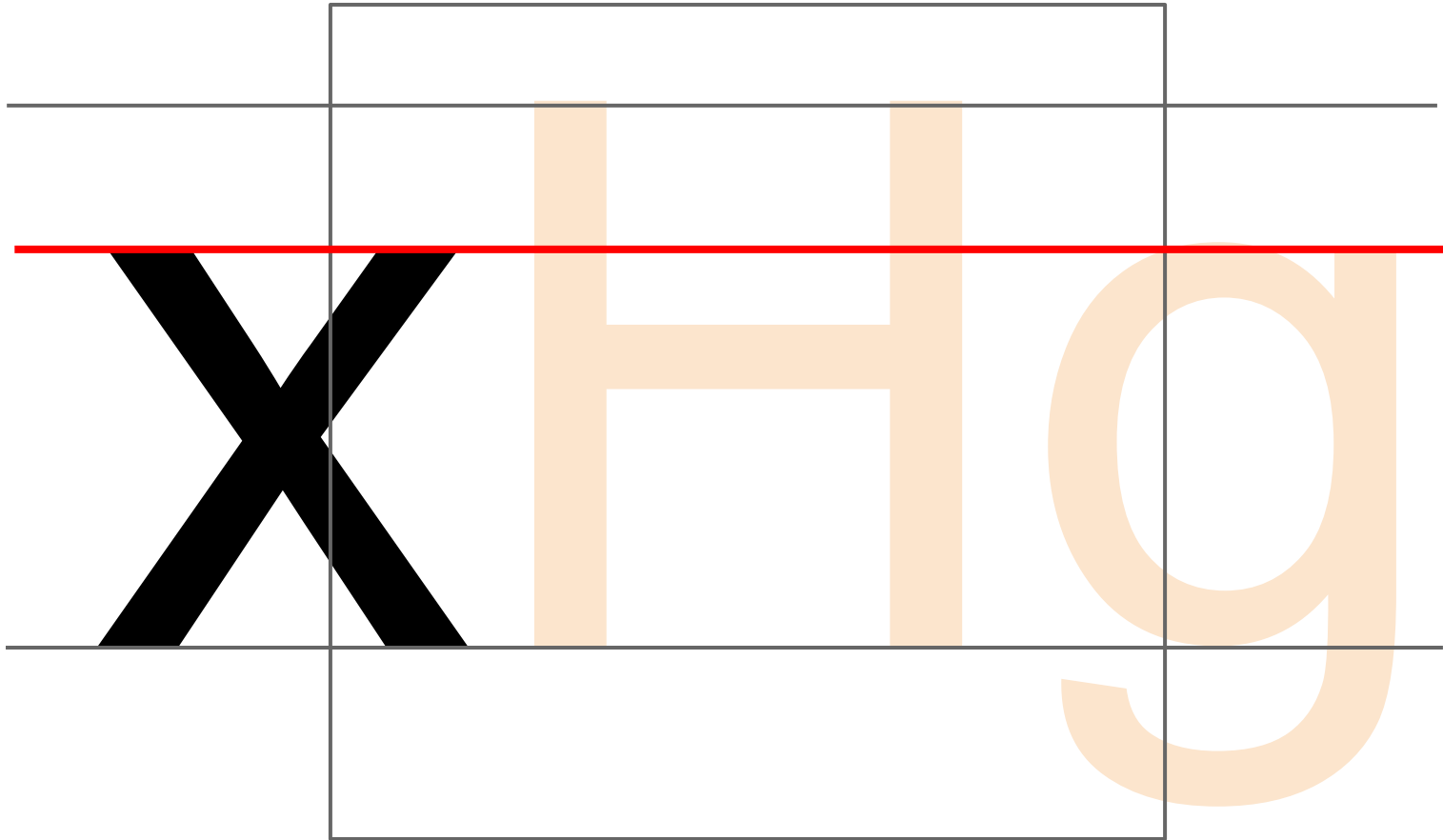


# the Cap Height

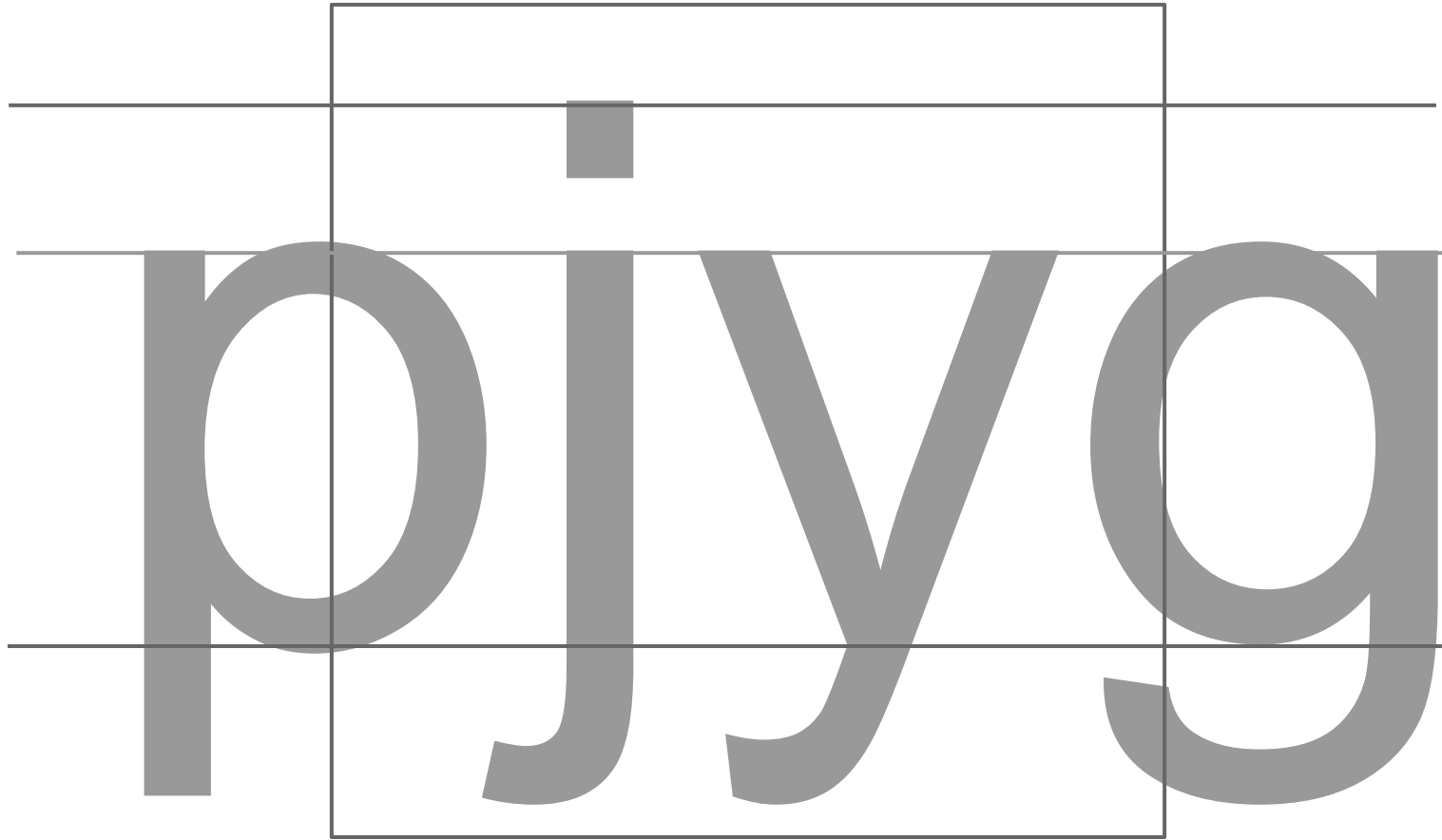




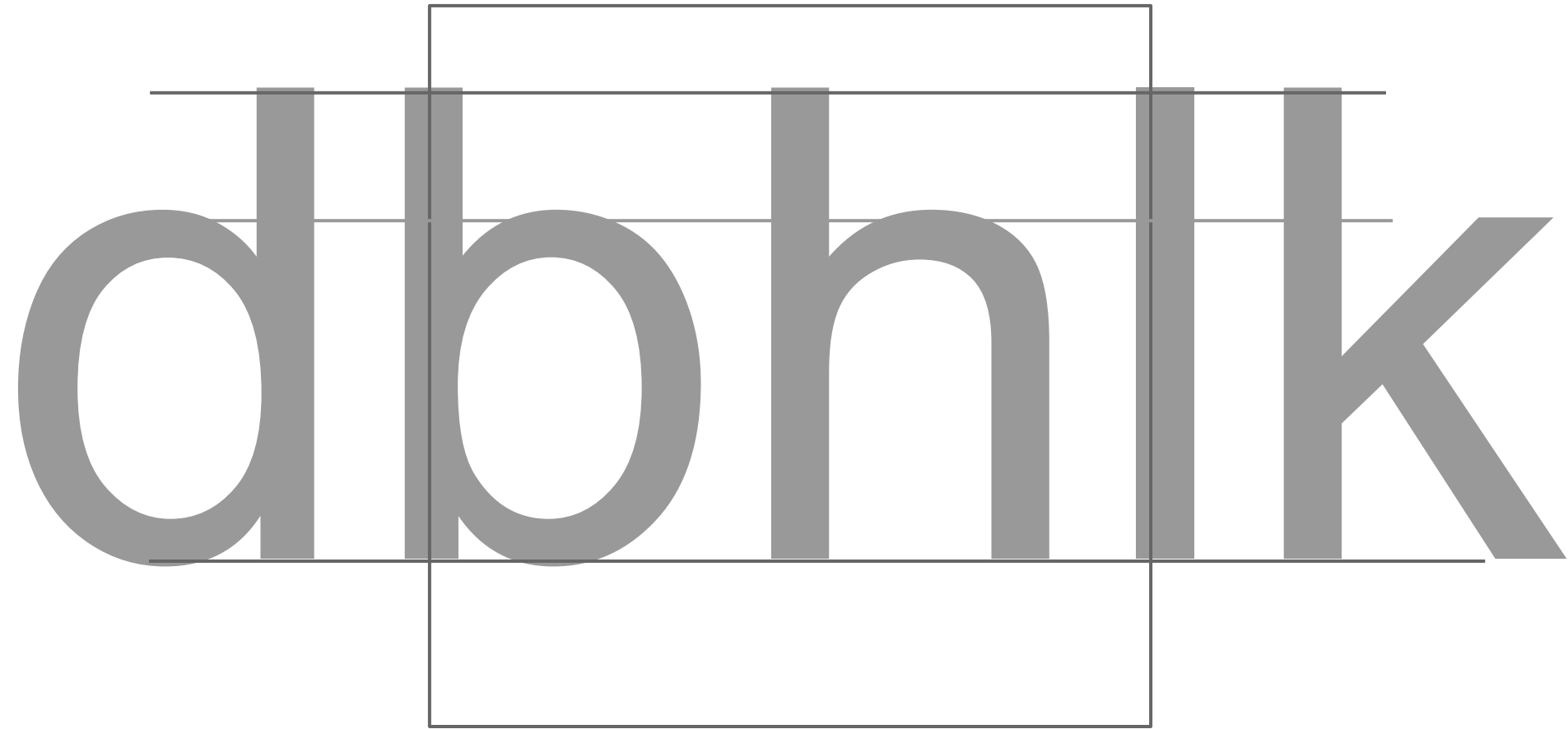
# the x Height



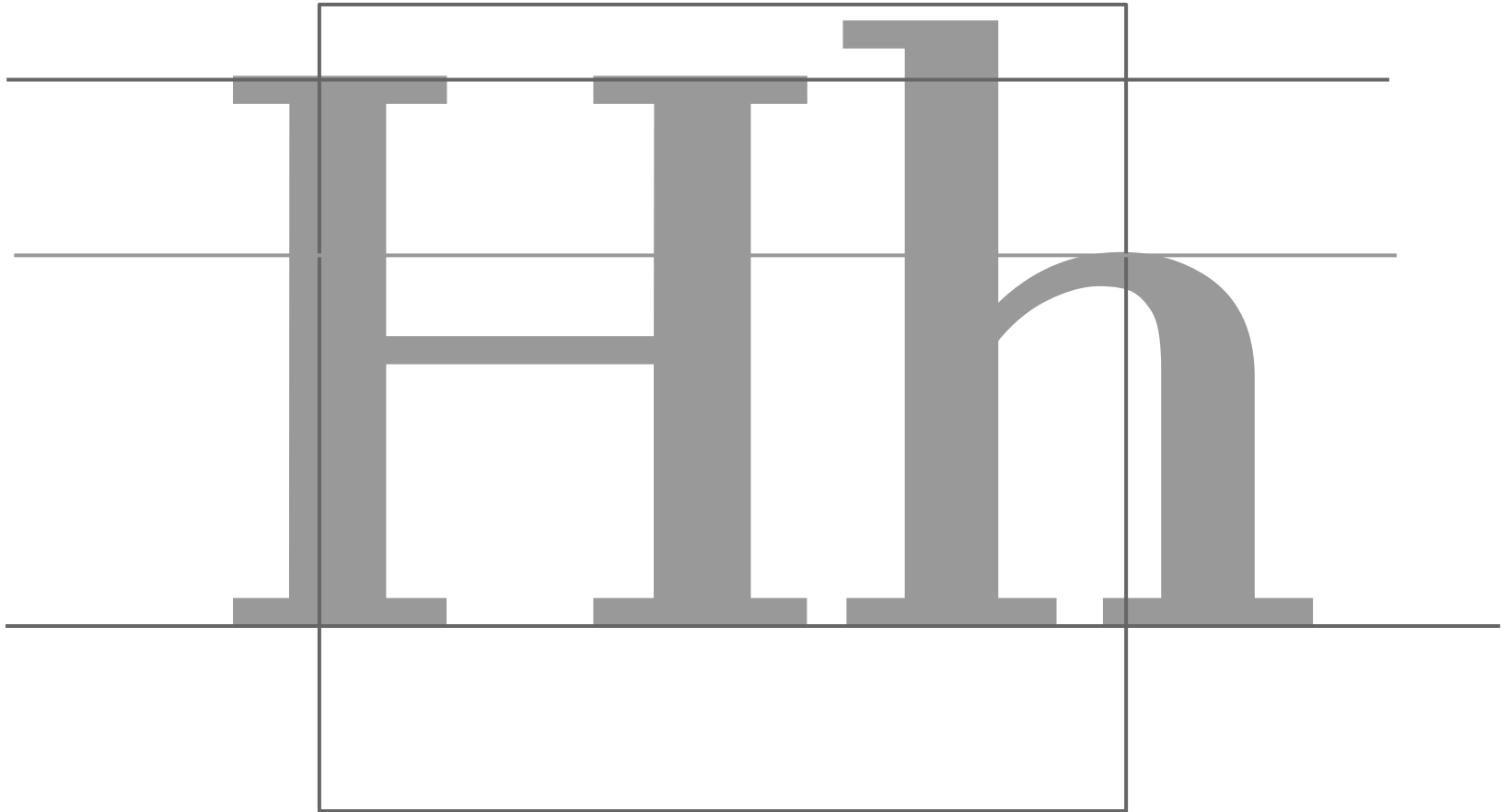
# the descenders



# the ascenders



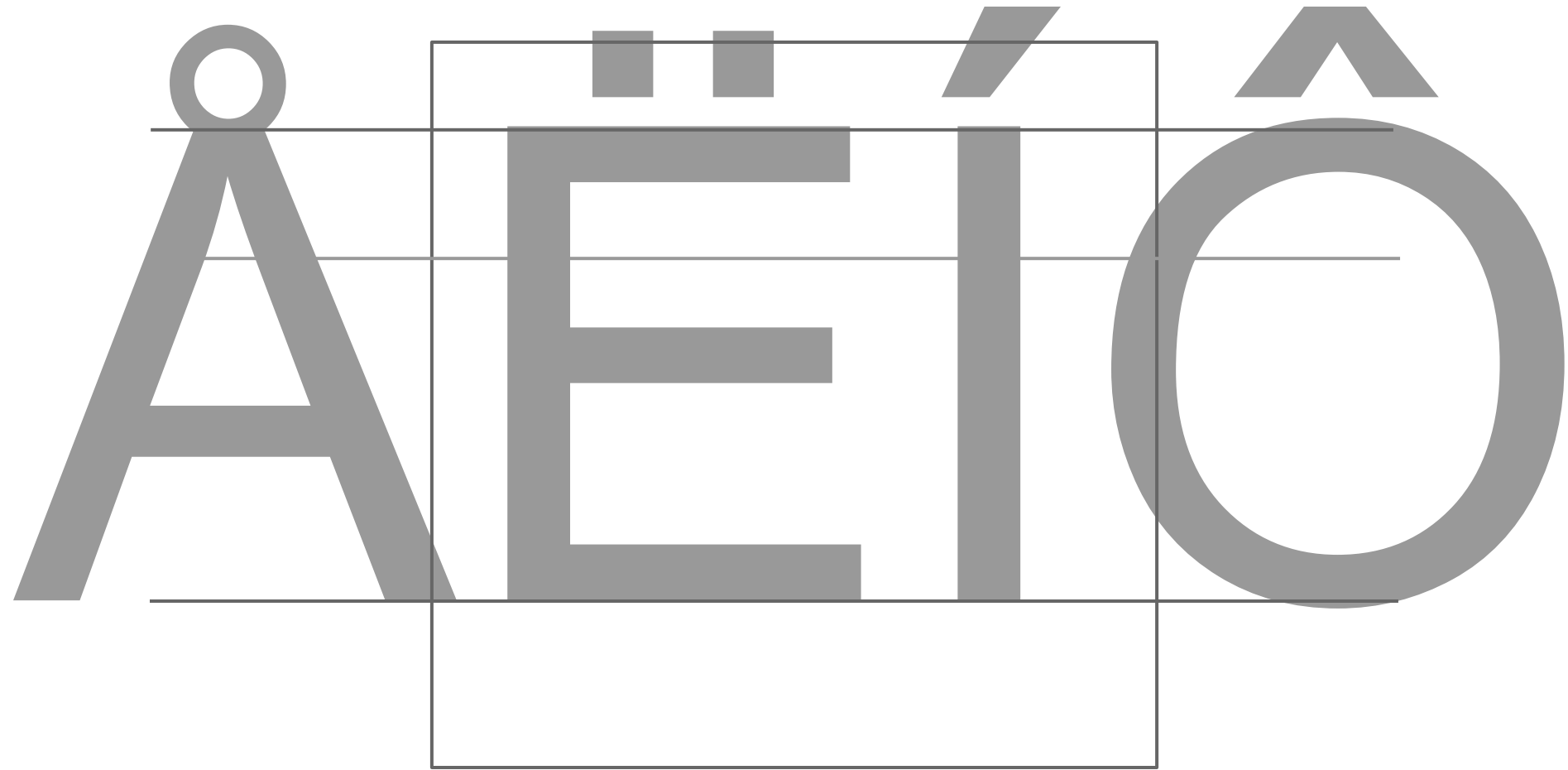
# the ascenders



# the diacritics



# the diacritics





# **Making problem free glyphs**

Once we start drawing glyphs, we need to ensure that they are not buggy or will cause errors when we print. Fontforge has tools to help us with this.



# **Remove overlap**

(ctrl + shift + O)

This function will merge separate, but overlapping objects, into a single object.

# **Simplify**

(ctrl + shift + M)

The simplify command can iron out and tidy up any 'less than perfect' curves and points

# Add Extrema

(ctrl + shift + X)

It is a *very good idea* that the extrema of any curve in a font consists of a point. This function will automatically place a point at these extrema.

# Round to Integer

(ctrl + shift + \_)

As you draw your glyphs points can get placed at fractions of units in the UPM. This can be problematic, so the Round to Integer command nails all points to whole units.

# **Correct Direction**

(ctrl + shift + D)

In OpenType postscript fonts, path must run clockwise from the outermost path and then alternately counterclockwise and clockwise for all subsequent inner paths. This functions auto corrects path directions.

# Find Problems

This is a complex and advanced tool for finding many problems in your glyphs. There are many useful functions here such as 'find open paths'.

Find Problems

CID BB Random

Points Paths Refs Hints ATT

- ☒ Non-Integral coordinates
- ☐ X near<sup>1</sup> 0
- ☐ Y near<sup>1</sup> 0
- ☐ Y near<sup>1</sup> standard heights
- ☒ Control Points near horizontal/vertic
- ☒ Control Points beyond spline
- ☐ Check for irrelevant control points
- Irrelevant Factor: 0.5 %
- ☐ Points too close
- ☐ Points too far

Clear All Set All

<sup>1</sup> "Near" means within 3 em-units

OK Cancel

Find Problems

CID BB Random

Points Paths Refs Hints ATT

- ☒ Open Paths
- ☒ Intersecting Paths
- ☐ Edges near horizontal/vertical
- ☒ Check outermost paths clockwise
- ☒ Check missing extrema
- ☐ More points than: 1500

Clear All Set All

<sup>1</sup> "Near" means within 3 em-units

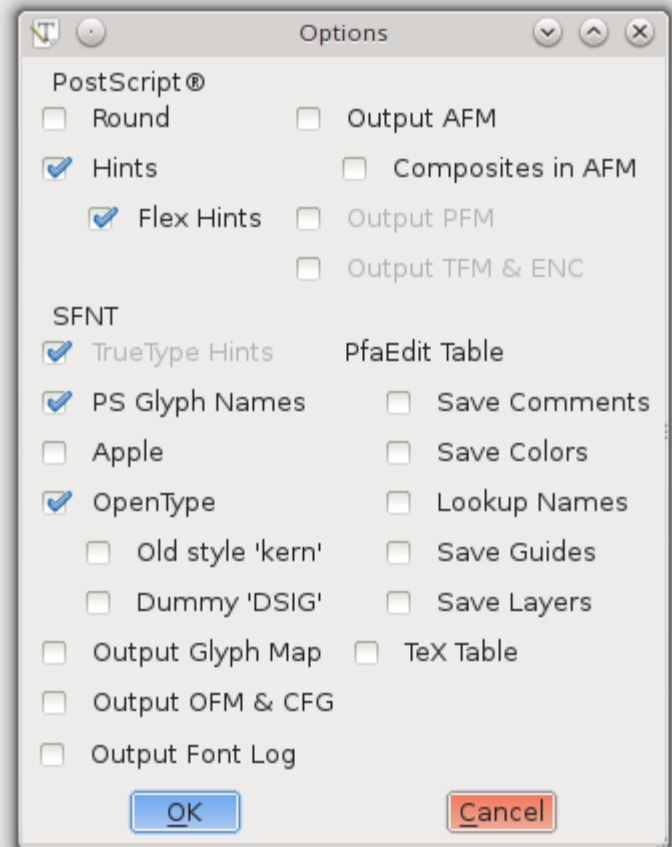
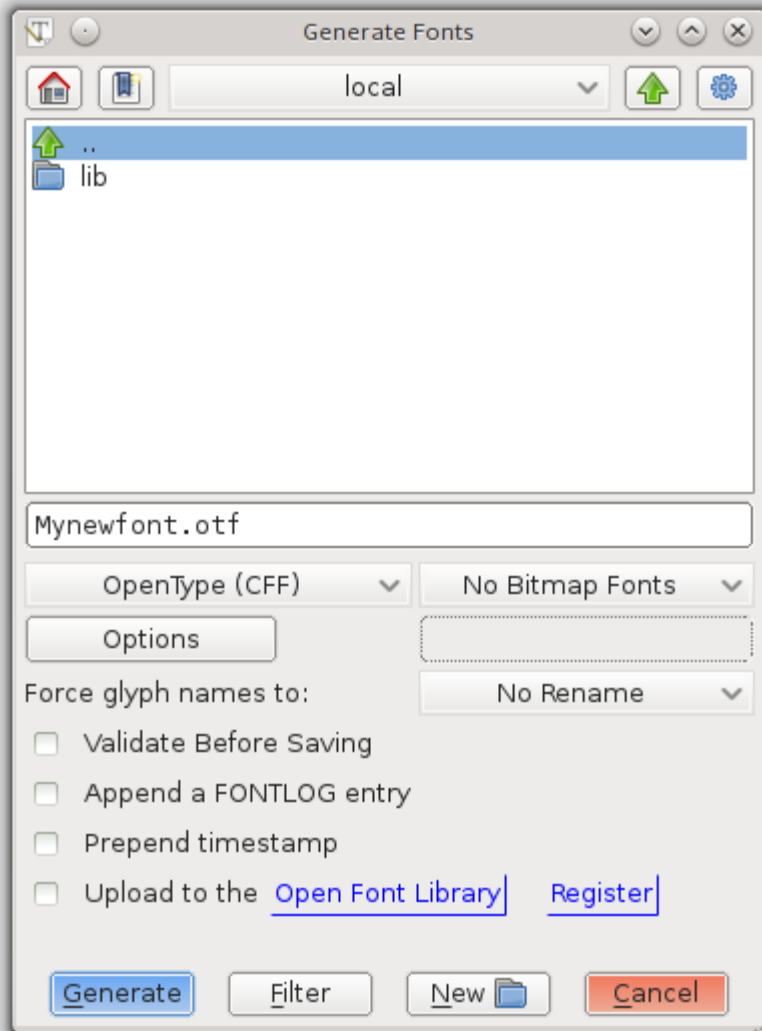
OK Cancel

## **Generate Fonts...**

Once we are sure that our glyphs are error free we can create an OpenType font file and test our new font by running a test print, or three.



From the File menu, choose  
'Generate fonts...' and select  
'OpenType (CFF)'.



You are now ready to install  
your font, print a test, and see  
what you have (so far) created!