



## Custom Fonts for CircuitPython Displays

Created by Ruiz Brothers



Last updated on 2021-04-04 09:14:00 PM EDT

## Guide Contents

|                                |    |
|--------------------------------|----|
| Guide Contents                 | 2  |
| Overview                       | 3  |
| More Fonts                     | 3  |
| Why Bitmaps?                   | 3  |
| Font Forging                   | 3  |
| Getting Started with FontForge | 3  |
| Where Do I Get Fonts?          | 4  |
| Use FontForge                  | 5  |
| Demo Walkthrough               | 5  |
| Open Font                      | 5  |
| Set Font Size                  | 5  |
| Generate Bits                  | 6  |
| Export Converted Font          | 6  |
| BDF Resolution                 | 6  |
| Optimize File Size             | 7  |
| Optimize File Size (Manually)  | 7  |
| Font Colors                    | 7  |
| Use otf2bdf                    | 9  |
| Convert to PCF                 | 10 |

# Overview



## More Fonts

Are you looking to display new fonts on your PyPortal? You can use just about any font on your computer or downloaded from the internet. This guide will walk you through generating bitmap fonts using the [FontForge open-source \(https://adafru.it/DZk\)](https://adafru.it/DZk) project.

## Why Bitmaps?

PyPortal uses the [CircuitPython Bitmap Font Library \(https://adafru.it/DZI\)](https://adafru.it/DZI) to render "live" text on the display. A bitmap font stores each character as an array of pixels. Bitmap fonts are simply groups of images. For each variant of the font, there is a complete set of images, with each set containing an image for each character.

Computers, on the other hand, use variable size 'TrueType' or 'Postscript' fonts, where there's a mathematical algorithm that defines each character, so it can be drawn at *any* size.

## Font Forging

This is where FontForge comes into play. FontForge is an open-source font editor for Windows, Mac OS and GNU+Linux. It features tools for converting existing fonts into different font formats.



## Getting Started with FontForge

Head on over to the [fontforge page \(https://adafru.it/DZm\)](https://adafru.it/DZm) and download the app for your platform. You can choose to donate, subscribe via email or simply click the **"Subscribe/Confirm and Download"** button (no need to enter an email). Follow along with the detailed installation guide to get setup with FontForge.

<https://adafru.it/DZn>

<https://adafru.it/DZn>

## Where Do I Get Fonts?

Here's a list of some neat places to obtain some fresh fonts.

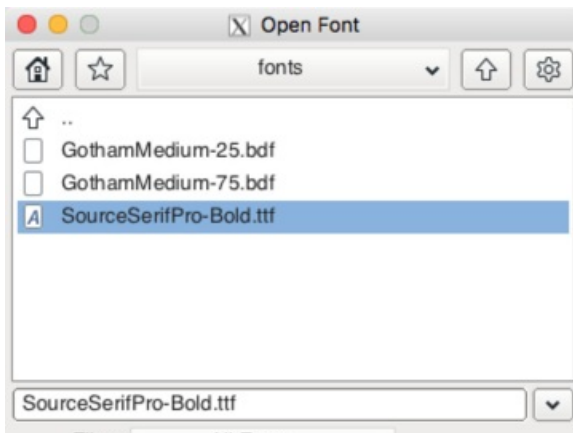
- [Font Squirrel \(https://adafru.it/DZo\)](https://adafru.it/DZo)
- [Google Fonts \(https://adafru.it/DZp\)](https://adafru.it/DZp)
- [Adobe Fonts \(https://adafru.it/DZq\)](https://adafru.it/DZq)
- [DaFont \(https://adafru.it/DZr\)](https://adafru.it/DZr)
- [Font Library \(https://adafru.it/DZs\)](https://adafru.it/DZs)

# Use FontForge



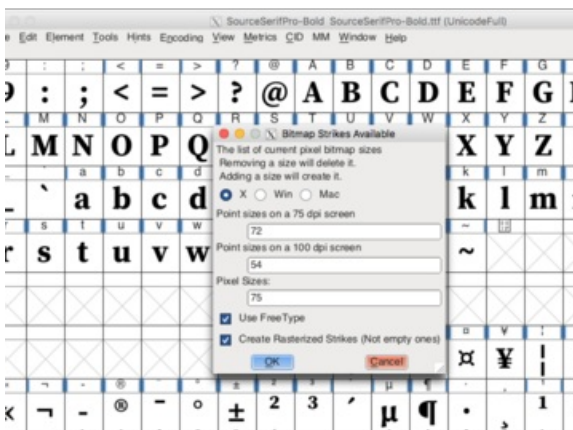
## Demo Walkthrough

In this example, we're going to convert a **.TTF** (TrueType Format) into a **.BDF** (Bitmap Distribution Format). I'm using an open licensed font downloaded from Google Font, [Source Serif Pro](https://adafru.it/DZt) (<https://adafru.it/DZt>).



### Open Font

Use the **file** menu and choose **Open Font** from the list. Navigate to a directory where your desired font resides. Select the font and open it.



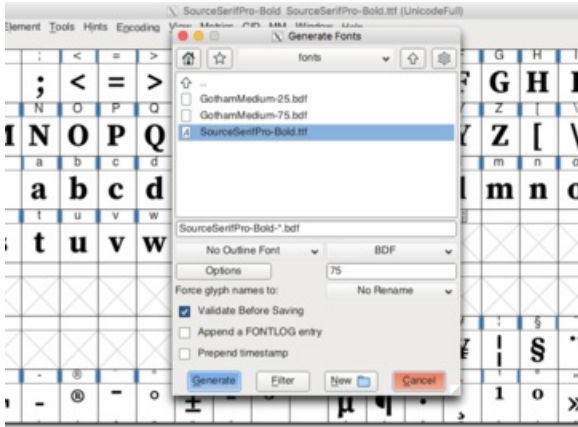
### Set Font Size

From the **element** menu, select **Bitmap Strikes Available**. In this dialog, you will need to specify how large you want your font to be. The font size is fixed with Bitmap fonts, so if you want to use different sizes, you'll need to make separate files.



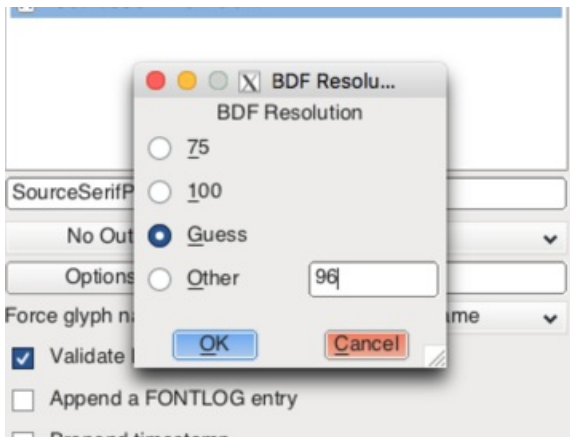
## Generate Bits

From the **element** menu, select **Generate Bitmap**. Similar to the previous dialog, enter the font size of your liking. You can make it smaller here. Be aware, values too small will not generate BDF's.



## Export Converted Font

From the **file** menu, select **Generate Fonts**. In the dialog, select **No Outline Font** and **BDF** from the dropdown options. Use the navigation UI to save the file in your directory of choice. Click the **generate** button to save the file.

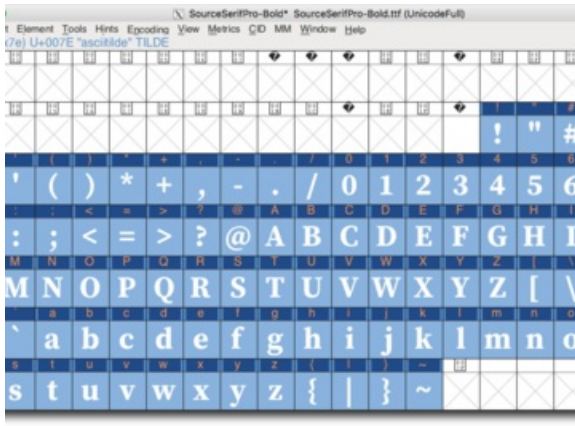


## BDF Resolution

This dialog menu will pop up after clicking generate. You can choose one of the options from the list. If you'd like a different font size, you can enter that in the **Other** labeled input box. Click **OK** to save it!

## Optimize File Size

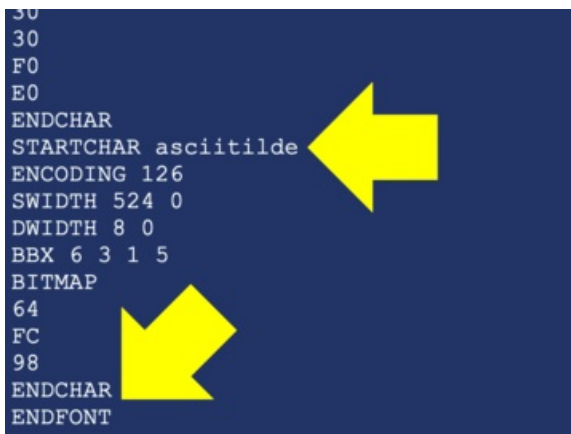
If you take a look at the file size of the .bdf, it's roughly around 900K – That can be a bit larger than needed, especially if you plan to store a lot of image and sound assets. In cases where you need to save on every byte, you can optimize the file size of your fonts by selecting only the characters you want to use. If you scroll through the full list of glyphs, you'll see there's extra special characters – A whole bunch of them! If you don't need them in your project, just select "space" (the glyph just before "!") plus the basic set of upper/lower and alphanumeric characters. You can click + hold and drag to make selections easier. With them selected, go through these steps:



1. Select the glyphs you want to keep
2. Use **Edit→Select→Invert Selection** to change the selection to the *unwanted* glyphs.
3. Use **Encoding→Detach & Remove Glyphs...** to remove the unwanted glyphs. (You'll have to re-load your original font file to undo this step)
4. Use **Element→Regenerate Bitmap** to reprocess the glyphs.
5. Use **File→Generate Font** to save the reduced version of the file

Make sure your font contains the letter capital M

Make sure your final font contains the letter capital **M**, which is used to estimate the height of letters in the font. Otherwise, the font will be incompatible with `adafruit_display_text` and give an error like `AttributeError: 'NoneType' object has no attribute 'height'`



## Optimize File Size (Manually)

If you prefer, you can also use a text editor to remove glyphs from a .bdf file. BDF files are just text!

Open a BDF file and search for “`asciitilde`” — this is usually the highest plain-ASCII-value glyph we want to preserve. A few lines down there will be an “`ENDCHAR`” line.

Delete everything after the `ENDCHAR` line, then add a line containing `ENDFONT`. That's it! Save the file, which is usually just a small fraction of the original size.

You won't get any accented characters or special punctuation this way, so it's not always the right thing for every situation.

For the majority of plain-text programs though, this can really help stretch your CIRCUITPY drive space!

## Font Colors

The color of the fonts can be setup in your code. The CircuitPython library uses HEX color codes. This is similar to web color pickers but formatted slightly different. Most HEX color pickers use a hashtag in the front of the value, like, `#000000`. In CircuitPython, instead of a hashtag, `0x` is used. Here's a few examples.

- Black = `0x000000`
- White = `0xFFFFFF`
- Purple = `0x8f42f4`



# Use otf2bdf

For fast conversion, you can also use a command line tool called otf2bdf. The homepage is here: <http://sofia.nmsu.edu/~mleisher/Software/otf2bdf/> (<https://adafru.it/RpB>)

Linux users can install it with something like `apt-get install otf2bdf`.

Here is a pre-compiled version of **otf2bdf** but for Mac Users

<https://adafru.it/RpC>

<https://adafru.it/RpC>

Use it in a terminal by calling it like:

```
otf2bdf FontFile.ttf -p pointsize -o FontFile.bdf
```

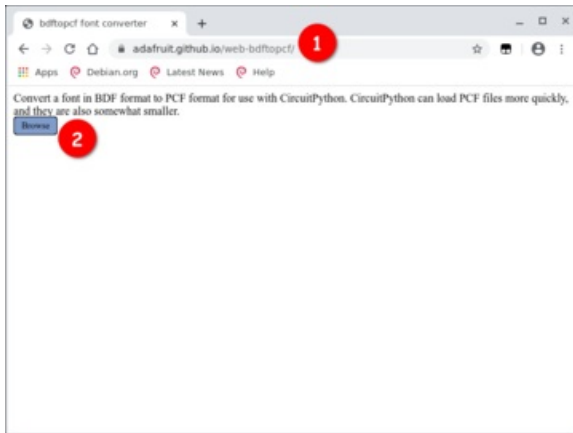
For example here is how to convert a font to a 12 point BDF file:

```
otf2bdf ChicagoFLF.ttf -p 12 -o Chicago-12.bdf
```

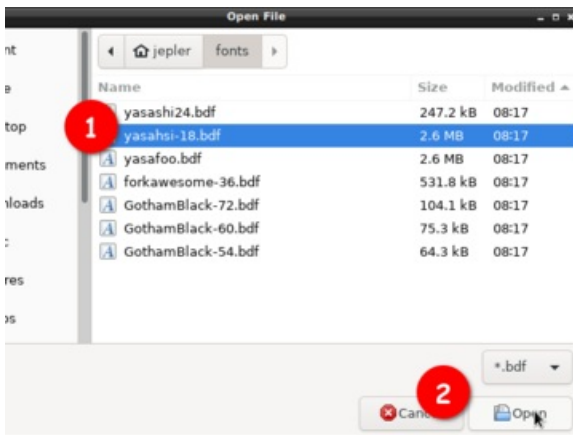
# Convert to PCF

CircuitPython supports two font formats: the textual `.bdf` format and the binary `.pcf` format. By taking the extra step of converting your font to `.pcf` you make fonts load faster and also typically save some storage space on the board **CIRCUITPY** flash drive.

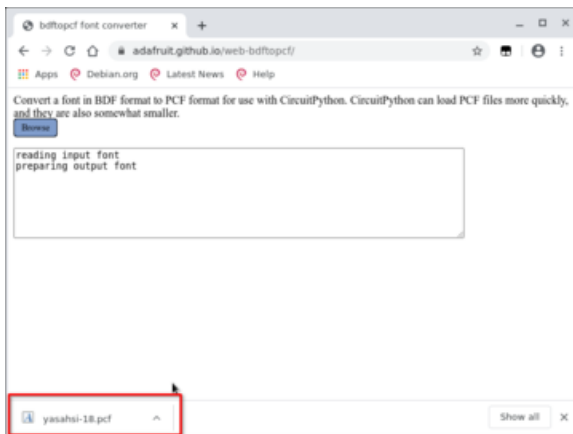
The converter software is hosted on github.io. Thanks to technology called [emscripten](https://adafru.it/PEn) (<https://adafru.it/PEn>), it runs entirely in your web browser—the font file is not uploaded to a server, which also makes it really quick. `web-bdftopcf` is derived from the classic font converter of the same name, a program for Unix/Linux systems. If you're interested, you can [browse the C source on github](https://adafru.it/PEo) (<https://adafru.it/PEo>).



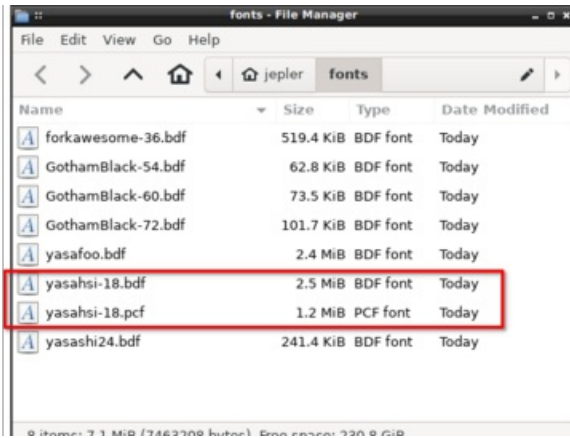
Head to <https://adafruit.github.io/web-bdftopcf/> (<https://adafru.it/PEp>) and click the "Browse" button.



Select a `.bdf` file from your computer and click Open.



After a moment, the font will be prepared in `.pcf` format and depending on your browser settings it may be automatically downloaded or you may have to confirm that you want to download the file.



In this particular case, the .pcf version of the font is only half the size of the .bdf font, which leaves more space on the **CIRCUITPY** drive for other assets like sound files and bitmaps.

