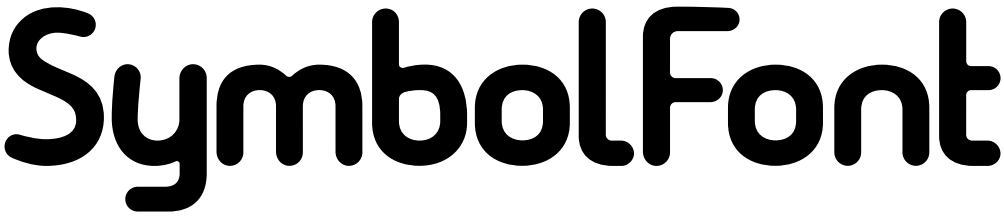
VRD | SymbolFont

∞0.0.1, 2024/02/15

Contents

Preface

SymbolFont is an Open Source project of **VivaRado**.



Introduction

SymbolFont is a method of publishing symbols universally accessible through ligatures. Is also a library of symbols created by VivaRado for various projects as needed, and eventually open sourced. Some symbols will become available.

Contributors:

- VivaRado support@vivarado.com
- Andreas Kalpakidis
- Madina Akhmatova

Introduction / Profile

- Company: VivaRado LLP
- Designer: Andreas Kalpakidis
- Twitter: @vivarado
- OpenSea: @vivarado

Introduction / Project Overview

- Project Name: SymbolFont
- Code Name: SF
- Proposal Date: 15/02/2024

Design

SymbolFont is designed to export symbols from a vector file into individual SVG files and compile them into a font. The file names of those SVG files are the ligatures you will need to use in order to access them on your website, after you successfully declare the font with `@font-face` and activate `font-feature-settings:"liga"` on the element with the ligature text.

It consists of the **SF_AI_Exporter.jsx** to export SVG icons from illustrator (more vector editors should be added) and the **SF_gen.py** that will take care of taking those SVGs and turning them into a ligature Opentype (OTF) font.

Design / Exporters

Currently there is only one exporter for Adobe Illustrator. Feel free to add more exporters for other vector editors like:

- InkScape
- GIMP
- CorelDRAW
- Figma

Design / Exporters / SF_AI_Exporter

The **SF_AI_Exporter** is a JSX script responsible for exporting compound paths / paths from nested layers, created in Adobe Illustrator, into individual SVG files with ligature conventional names (underscore separated letters). A file has been provided in `Templates`.

AI file structure:

```
┌─ SYMBOL_PACKAGE_A (Layer, SymbolPackage)
│  ├─ S_Y_M_B_O_L_A (Layer, Symbol, Choose Name)
│  │  └─ <Symbol> (Group)
│  │     ├─ <CenterLayer> (Group, Centering Method)
│  │     │  ├─ <Path> (PathPoint, TL)
│  │     │  ├─ <Path> (PathPoint, TR)
│  │     │  ├─ <Path> (PathPoint, BR)
│  │     │  └─ <Path> (PathPoint, BL)
│  │     └─ <CompoundPath> (CompoundPath, Artwork)
│  ├─ S_Y_M_B_O_L_B (Layer, Symbol, Choose Name)
│  └─ ...
├─ SYMBOL_PACKAGE_B (Layer, SymbolPackage)
└─ ...
```

The file is a grid of artboards, the symbols are included in a SymbolPackage, you can have multiple SymbolPackages on one file, occupying those same artboards.

> Inside each SymbolPackage are Symbol Layers that include the artwork and a centering layer that is a group with four points, one on each corner, this is used in the SVG generator script to center the icon during export. This layer is optional but the artwork will be centered in the SVG file ignoring the location you drew it on the artboard if not present (other solutions welcome).

> This will generate valid SVG files with the same name as the layer e.g.: `S_Y_M_B_O_L_A.svg`. When you in turn type SYMBOLA in a ligature enabled HTML element with the font active on it, it will produce this symbol.

Before you run the **SF_AI_Exporter.jsx** from the `File/Scripts/Other Scripts` (F12) make sure to click on the SymbolPackage Layer you want exported as the script operates on the **activeLayer**.

Design / Generators

Currently there is only one generator written in python utilizing fontTools and svg.path. Feel free to add more generators.

Design / Generators / SF_gen

The **SF_gen** is a Python script that generates an OpenType (OTF) font file using fontTools and svg.path as requirements.

> By providing a SVG directory with properly named files, this script will import the SVGs, convert them into GLIF and add them to the fonts glyph list.

> It will generate the padding needed for a ligature font to work, like A-Z and some other basic characters, set glyph dimensions, order them and create a character map, create the GSUB with the ligatures and save the file in `Source/fonts`

Installation

To install:

```
pip install -r Lib/requirements.txt
```

Generation

To Generate a ligature font from SVGs:

```
python SF_gen.py -s <folder_path>
```

Arguments:

- **-s** : ligature named SVG file directory.
- **-n** : Optional name of your font file, the default name is **symbolfont** followed by a number increment.

> Generated Font Directory: `Source/fonts`

Usage

To use the fonts, there is a simple example `Source/fonts/example.html`. The document has to link the font with font-face in CSS and enable ligatures on the element that includes the desired text.

CSS:

```
@font-face{
  font-family: 'SymbolFont';
  src: url('symbolfont.otf') format('opentype');
}

.liga {
  /*
  -moz-font-feature-settings:"liga=1";
  -moz-font-feature-settings:"liga";
  -ms-font-feature-settings:"liga";
  -o-font-feature-settings:"liga";
  -webkit-font-feature-settings: "liga";
  */
  font-feature-settings:"liga";
  font-family: 'SymbolFont';

}
```

HTML:

```
<p class='liga'>ABC</p>
```

Glossary

BUMP

Reference

fontTools

svg.path

Preface

SymbolFont is an Open Source project of **VivaRado**.

SymbolFont

Introduction

SymbolFont is a method of publishing symbols universally accessible through ligatures. Is also a library of symbols created by VivaRado for various projects as needed, and eventually open sourced. Some symbols will become available.

Contributors:

- VivaRado support@vivarado.com
- Andreas Kalpakidis
- Madina Akhmatova

Introduction / Profile

- Company: VivaRado LLP
- Designer: Andreas Kalpakidis
- Twitter: @vivarado
- OpenSea: @vivarado

Introduction / Project Overview

- Project Name: SymbolFont
- Code Name: SF
- Proposal Date: 15/02/2024

Design

SymbolFont is designed to export symbols from a vector file into individual SVG files and compile them into a font. The file names of those SVG files are the ligatures you will need to use in order to access them on your website, after you successfully declare the font with `@font-face` and activate `font-feature-settings:"liga"` on the element with the ligature text.

It consists of the **SF_AI_Exporter.jsx** to export SVG icons from illustrator (more vector editors should be added) and the **SF_gen.py** that will take care of taking those SVGs and turning them into a ligature Opentype (OTF) font.

Design / Exporters

Currently there is only one exporter for Adobe Illustrator. Feel free to add more exporters for other vector editors like:

- InkScape
- GIMP
- CorelDRAW
- Figma

Design / Exporters / SF_AI_Exporter

The **SF_AI_Exporter** is a JSX script responsible for exporting compound paths / paths from nested layers, created in Adobe Illustrator, into individual SVG files with ligature conventional names (underscore separated letters). A file has been provided in `Templates`.

AI file structure:

```
┌─ SYMBOL_PACKAGE_A (Layer, SymbolPackage)
│  ├─ S_Y_M_B_O_L_A (Layer, Symbol, Choose Name)
│  │  └─ <Symbol> (Group)
│  │     ├─ <CenterLayer> (Group, Centering Method)
│  │     │  ├─ <Path> (PathPoint, TL)
│  │     │  ├─ <Path> (PathPoint, TR)
│  │     │  ├─ <Path> (PathPoint, BR)
│  │     │  └─ <Path> (PathPoint, BL)
│  │     └─ <CompoundPath> (CompoundPath, Artwork)
│  ├─ S_Y_M_B_O_L_B (Layer, Symbol, Choose Name)
│  └─ ...
├─ SYMBOL_PACKAGE_B (Layer, SymbolPackage)
└─ ...
```

The file is a grid of artboards, the symbols are included in a SymbolPackage, you can have multiple SymbolPackages on one file, occupying those same artboards.

> Inside each SymbolPackage are Symbol Layers that include the artwork and a centering layer that is a group with four points, one on each corner, this is used in the SVG generator script to center the icon during export. This layer is optional but the artwork will be centered in the SVG file ignoring the location you drew it on the artboard if not present (other solutions welcome).

> This will generate valid SVG files with the same name as the layer e.g.: `S_Y_M_B_O_L_A.svg`. When you in turn type SYMBOLA in a ligature enabled HTML element with the font active on it, it will produce this symbol.

Before you run the **SF_AI_Exporter.jsx** from the `File/Scripts/Other Scripts` (F12) make sure to click on the SymbolPackage Layer you want exported as the script operates on the **activeLayer**.

Design / Generators

Currently there is only one generator written in python utilizing fontTools and svg.path. Feel free to add more generators.

Design / Generators / SF_gen

The **SF_gen** is a Python script that generates an OpenType (OTF) font file using fontTools and svg.path as requirements.

> By providing a SVG directory with properly named files, this script will import the SVGs, convert them into GLIF and add them to the fonts glyph list.
>
> It will generate the padding needed for a ligature font to work, like A-Z and some other basic characters, set glyph dimensions, order them and create a character map, create the GSUB with the ligatures and save the file in `Source/fonts`

Installation

To install:

```
pip install -r Lib/requirements.txt
```

Generation

To Generate a ligature font from SVGs:

```
python SF_gen.py -s <folder_path>
```

Arguments:

- **-s** : ligature named SVG file directory.
- **-n** : Optional name of your font file, the default name is **symbolfont** followed by a number increment.

> Generated Font Directory: `Source/fonts`

Usage

To use the fonts, there is a simple example `Source/fonts/example.html`. The document has to link the font with font-face in CSS and enable ligatures on the element that includes the desired text.

CSS:

```
@font-face{
  font-family: 'SymbolFont';
  src: url('symbolfont.otf') format('opentype');
}

.liga {
  /*
  -moz-font-feature-settings:"liga=1";
  -moz-font-feature-settings:"liga";
  -ms-font-feature-settings:"liga";
  -o-font-feature-settings:"liga";
  -webkit-font-feature-settings: "liga";
  */
  font-feature-settings:"liga";
  font-family: 'SymbolFont';

}
```

HTML:

```
<p class='liga'>ABC</p>
```

Glossary

BUMP

Reference

fontTools

svg.path
