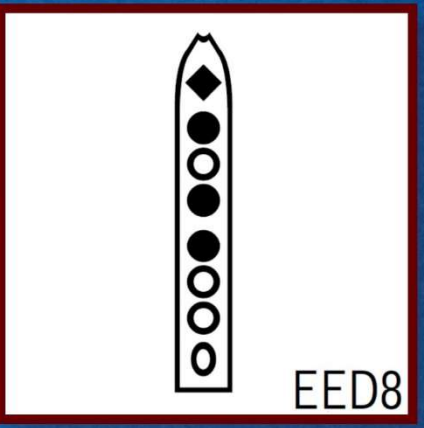
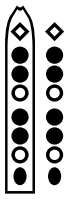
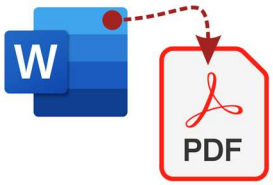
Warbl® Font Folio User's Guide



Warbl Font Folio – User's Guide



The Warbl is an electronic wind instrument designed and produced by ***Andrew Mowry*** in Oregon, USA. This **WARBL FONT FOLIO** is a collection of free fonts with fingering diagrams (such as ◇●●○●●○● and ◇●●○●●○●) for the Warbl electronic wind instrument.



These TrueType fonts are specifically intended for authors who are writing documents related to the Warbl, such as engraving sheet music and authoring educational material. The fonts are particularly useful for people who use *Microsoft Word* to publish PDF documents (such as this User's Guide). This document is set in **WARBL** and **WARBLC** fonts.

In addition to being useful for normal text, these fonts have finger diagrams for all possible combinations of open and closed finger holes on the Warbl instrument, as well as a broad range of characters for typesetting music. Each font has both vertical and horizontal finger diagrams.

The **WARBL** fonts are available at https://www.Flutopedia.com/warbl_fonts.htm. The collection includes two core typefaces:

- **WARBL** provides vertical and horizontal finger diagrams with an outline: ◇●●○●●○●
- **WARBLC** has finger diagrams with no outline: ◇●●○●●○● (the "C" stands for "Clean").

… each of which are in 8 styles:

- Regular weight upright and *Italic characters*.
- Medium weight upright and *Italic characters*, which are slightly heavier than Regular.
- **SemiBold weight upright and *Italic characters*, which are even heavier.**
- **Bold weight upright and *Italic characters*, which are the heaviest.**

Using these fonts, you can write in a mixture of text and finger diagrams without switching fonts to access those finger diagrams. You can mix any of the typefaces in your document without the pitfalls often associated with using multiple fonts in a document. Note, however, that the finger diagrams and most of the music typesetting characters do not have different weights and slant angles in the different fonts.

If you have issues, suggestions, or ideas about the **WARBL** Font Folio, head over to the GitHub repository for reporting and discussing issues at https://GitHub.com/ClintGoss/Warbl.

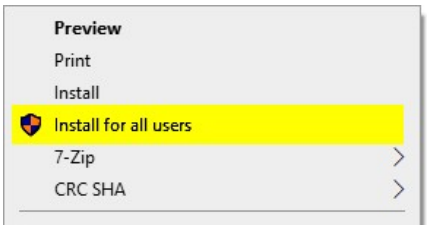
The **WARBL** fonts are and always will be free. The fonts are available to anyone at no charge and licensed under the *SIL Open Font License Version 1.1*. **WARBL** fonts may be used, studied, copied, merged, modified, and redistributed. You may download and embed these fonts into digital documents, use them in commercial projects (including mobile apps), and bundle them (with or without modification) for re-distribution under the terms of the *SIL Open Font License Version 1.1*.

I hope you find these fonts useful! — Clint Goss [clint@goss.com] as of February 13, 2026

If you wish to get started quickly, feel free to use this *Quick Start Guide* and start using the fonts. You will need to come back to this document for addition information – in particular, how to access the finger diagram characters.

Quick Start Guide – Warbl Font Folio

1. **Download** the release package from https://Flutopedia.com/warbl_fonts.htm
2. **Unpack** the release package `.zip` file into a directory and navigate to that directory.
3. **Navigate** to the `/Fonts` subdirectory.
4. **Install** the fonts in this directory.

   On Windows, I recommend:
   [Select fonts] ➞
   [Right-click] ➞
   [Install for all users] (shown above)

   On Apple systems, Right-click (or Control-click) on the font files and choose to open with *Font Book*.

   The *Warbl User's Guide* (`Warbl_UserGuide.pdf`) contains more information on installing fonts as well as how to access the finger diagrams.
5. **Restart** your system (to ensure that your editing application will see the new fonts).
6. **Edit** a document in your favorite editing application (*Microsoft Word*, *OpenOffice*, *WordPerfect*, etc.) Check that the **WARBL** and **WARBLC** fonts are shown in your application's list of available fonts.
7. **Consult** the *Warbl Character Map* PDF document for the codes needed to enter finger diagrams in your document.

```
-----------------------------------------------------------
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.

The Release Packages

A **WARBL** release package is a `.ZIP` file that can be downloaded from https://Flutopedia.com/warbl_fonts.htm. The `.ZIP` file contains all the files of that release.

A `.ZIP` file has a name such as:

`WarblFontFolio_v1.200_Full.zip`

The **blue** characters indicate the version of the release package. This will match the version number embedded in every font in that release package since all fonts are built from scratch for a release. Even if a particular font has not been altered since a prior release, it will carry this new version number.

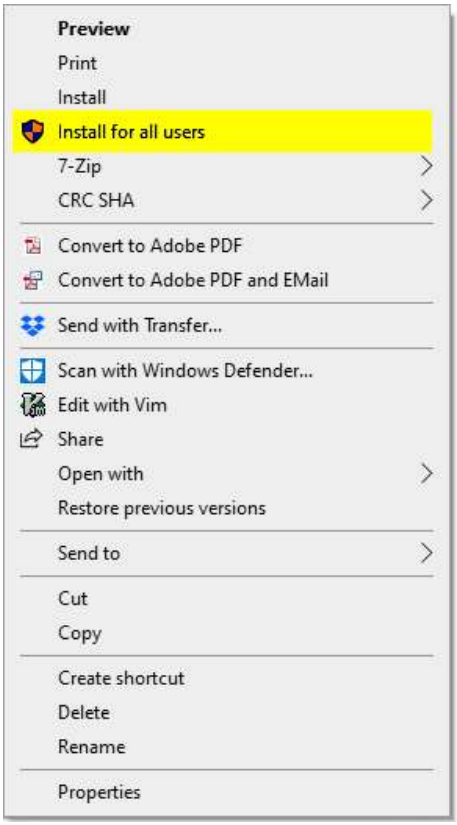
The **dark red** characters indicate the "flavor" of the package. The "Full" flavor of the **WARBL** fonts is the only flavor that is currently distributed.

Installing Fonts

There are many Web-based resources and streaming videos on how to install TrueType fonts on various systems. This section provides one way to install fonts on two of the most common systems.

Windows 10

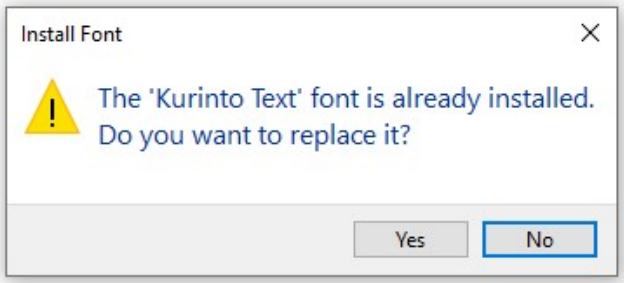
My preference is to install fonts on Windows systems using the Install for all users command. This avoids some rather daunting complexities associated with installing fonts "for the Current User".



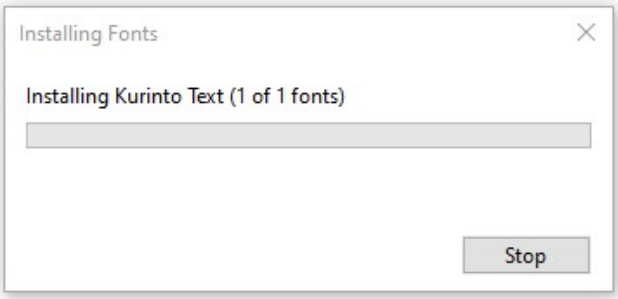
However, you will need to be logged in to an account with Administrator privileges:

1. Download the release package from www.Flutopedia.com/warbl_fonts.htm.
2. Unpack the release package from the `.zip` file. This can be done on most Windows systems by right-clicking on the `.zip` file you downloaded and selecting Extract All ... . If this does not work, there are many Web-based resources on how to accomplish this step. If you need an unzip utility, *7-Zip* is a good option.
3. Open Windows Explorer (the Windows "File Manager") and navigate to the directory (folder) where you unpacked the `.zip` file. Then move down to the `/Fonts` sub-directory.

4. Right click (or you may be able to press and hold) on the `XXX.ttf` font files you want to install.
5. Click/tap on `Install for all users` menu item. (The `Install` menu item would install for the current user only). See the context menu at the right.
6. If this font is already installed, you may this dialog box.[1] Click/tap on `Yes` to replace it:



7. The font should now install:



If you encounter issues installing on Windows or have a non-standard Windows setup, the suggestions in the Windows Font Installation section might be helpful.

[1] The exact behavior depends on the version numbers of the installed and new fonts. If you are installing a later version, the install may proceed without a warning dialog box. If you are installing the same version, you will be prompted with the warning dialog. If you are overwriting a **later** version of the font with an earlier (older) version, Windows may refuse to install the older font and you will need to explicitly uninstall the font to complete the installation. See Windows Font Installation for more details.

Apple Systems

Since I do not work on Apple systems, I am including install instructions from David J. Perry:

> Mac OS X: The easiest way is to right-click (or Control-click) on the font file(s) and choose to open with Font Book, which will typically be the default. You can also drag the font(s) from your folder into Font Book. The file will be validated and then appear in the list of fonts.
>
> There are other ways to install fonts in OS X by copying the font file to various locations. You can use this method if you know what you're doing.

Alternately, here is a page on the Apple web site on installing fonts:
https://support.apple.com/en-us/HT201749

On-Screen Display

WARBL fonts are optimized for print rendering.

For best on-screen rendering, it helps to have ClearType set up properly on Windows. When you bring up the ClearType interface, you can immediately see the effect of the changes you make.

To get started, click on the Windows start menu or in the lower left and type "ClearText in the search box. You are looking for "Adjust ClearType Text / Control Panel".

Here are some web references on setting up ClearType:

- https://www.boxaid.com/blog/how-to-fix-jagged-poor-quality-fonts-or-text-on-windows/
- https://www.winhelp.us/change-font-smoothing-in-windows.html
- https://superuser.com/questions/807951/fonts-smoothing-on-windows-how-to-disable-cleartype-but-still-get-a-smoothing
- https://superuser.com/questions/803637/how-to-disable-directwrite-in-google-chrome-37
- https://support.google.com/chrome/answer/95290?hl=en

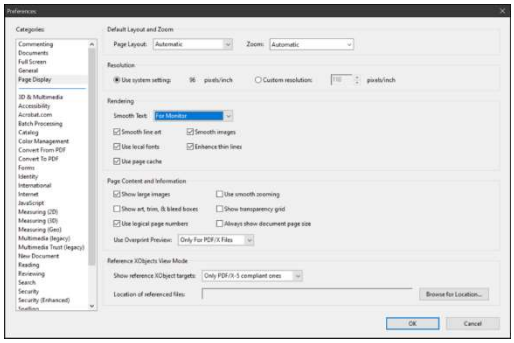
Viewing PDF Files On-Screen

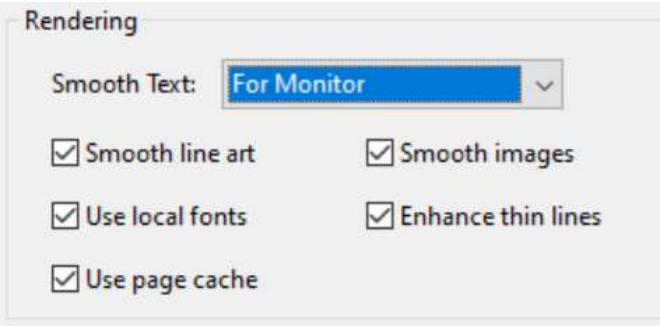
I have been able to dramatically improve the look of PDFs on computer monitors by setting a few configuration parameters. These parameters are in the application you use to view the PDF files (the "PDF viewer").

> One drawback of this situation is that the viewing experience of users of your published PDF files is dependent on **them** to set the parameters on their PDF viewer. You might consider providing them with a version of these instructions, but that is likely to help only a small fraction of your users.

It is beyond the scope of this document to provide specific advice across the huge variety of PDF viewers. However, the general approach is to find the configuration parameters in your PDF viewer, experiment with the settings, and re-open the document to see the effect. Here is one example that has worked for me:

For Adobe Acrobat Pro v9.0, open a PDF file and select Edit ➞ Preferences ➞ Page Display. The Preferences window (shown at the right) has a section for "Rendering":





It is not always obvious what settings will improve your on-screen renderings … you may need to experiment and re-load your PDF reader to see the results.

Using Unicode

An important part of good typesetting is using the full range of Unicode characters that are available. Getting set up with an easy way to (A) locate and (B) enter Unicode characters into your documents will go a **long** way to achieving that goal.

Locating Unicode Characters

To locate Unicode characters – in particular the finger diagrams – the main resource should be the `WarblCharacterMap.pdf` document, included with the release package.

For access to general Unicode characters, you might also consider these resources:

- the *Unicode Code Charts* documents or on-line resources, or
- the *BabelMap* application.

Getting Finger Diagrams into your Document

The approach I usually take is to open `WarblFont_AllFingerDiagrams.docx`, browse around for the character I want, and Copy/Paste it into my document. This file is included in the release package. Of course, you need to be able to run Microsoft Word for this to work.

Failing that, or if you want musical characters other than the finger diagrams, read on …

Getting Musical Characters into your Document

Since the **Warbl** fonts have all the characters from the Standard Music Font Layout (SMuFL), you can look up the character you need at the SMuFL web site at:

https://w3c.github.io/smufl/latest/

… and then use one of the input methods described below to get them into your document.

Text Input, or "How to Type Unicode Characters"

To get enter Unicode text, you will want to set up your keyboard for your primary writing system. If you use multiple writing systems, it helps to have a smooth way to switch your keyboard between the writing systems that you use most. In addition, you will need a tool to assist in entering Unicode characters into your document.

Microsoft Word

To enter a code point such as **U+EF78**, type E F 7 and 8. This will enter "EF78" into your document. Then hold down the Alt key and type X (this is often written as Alt + X). *Word* will convert the "EF78" into "  " – the character in the **Warbl** font at the code point **U+EF78**.

There are some understandable "gotcha's" with this system. When you type Alt + X, *Word* will interpret whatever digits precede the cursor as a Unicode code point. For example, if you type the text "1EF78", the Alt + X approach will get you the code point for **U+1EF78**, which will be "□" – a box indicating a missing character. In this situation, I will typically add a space character before the code point to get the character I want and then remove the extra space after the character is converted.

Web Pages

If you are editing HTML documents for web pages, you can enter any Unicode code point using the syntax, for example, `Ӓ`. However, the number in this syntax is a decimal (base 10) version of the code point number.

So, if you wish to enter the finger diagram at **U+EF78**, you can translate the hexadecimal **EF78** into decimal 61,304 and enter `` into your HTML source code. You can do this conversion using an on-line "hex-to-decimal" conversion utility.

As of this writing, some (but not all) browsers will interpret the syntax `` as a hexadecimal number.

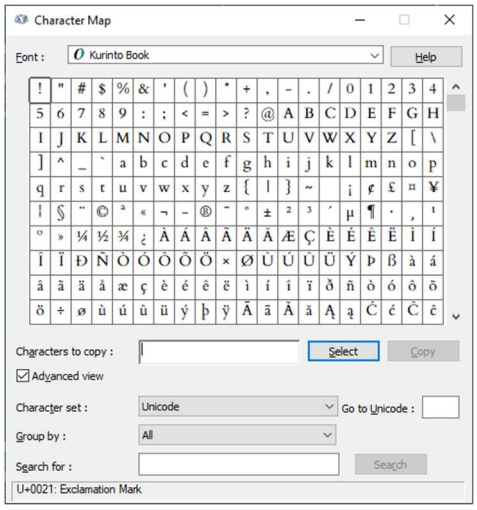
Adobe and Open Office

I do not currently use these applications, but I have been told that *Adobe InDesign* and *OpenOffice* (www.OpenOffice.org) have custom displays of the palette of glyphs in a given font. You can select the characters and enter them by clicking on the code point you want.

Windows

For other applications on *Windows*, you may need to use a separate application to produce the character so that it can be entered into your text.

Character Map is a standard Windows application to display the characters of any font. If you check the Advanced View option, you can select Unicode for the Character Set. Once you locate a character, you can copy it to the clipboard and paste it into your document.



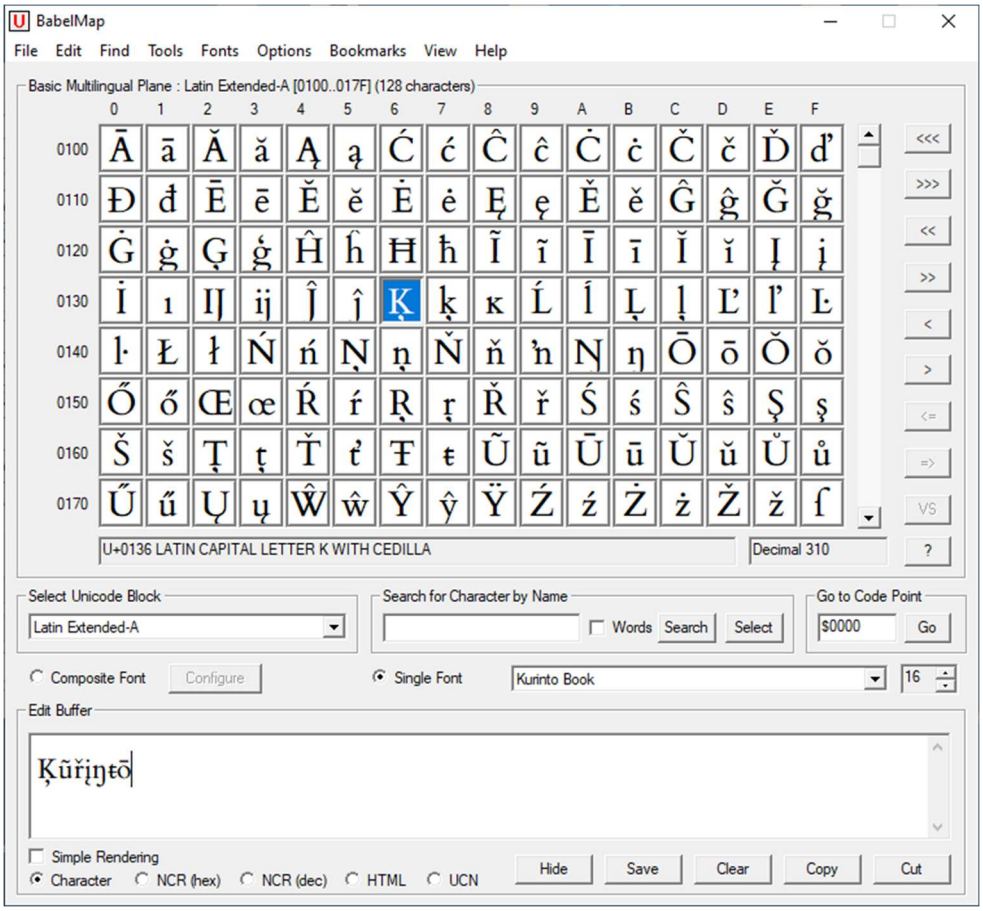
Windows Character Map

However, as of January 2020 in Windows 10, *Character Map* has numerous flaws that limit its usefulness:

- Characters are displayed in a single, tiny, fixed size.
- The application does not show any characters above **U+FFFF**.
- For reasons unexplained, it does not display certain characters in that range below **U+FFFF**.

These limitations severely limit the usefulness of *Character Map* for practical typesetting.

An excellent alternative is Andrew West's *BabelMap*, freely available at



Andrew West's BabelMap

http://www.Babelstone.co.uk/. It is similar in intent to Windows *Character Map*, but has vastly more functionality, features, and support.

The *Keyman* program (https://Keyman.com/ from SIL International) is another Windows application that I have not explored, but appears to be useful. It supports over 1,000 keyboard layouts.

Linux and MacOS

I do not currently used Linux systems for font development, so I cannot offer advice. However I have heard good reports on these options:

- On Linux systems such as Ubuntu, *KMFL* (http://kmfl.sourceforge.net/) is available.
- *Ukelele* (http://scripts.sil.org/ukelele) is available for Mac OS X versions 10.2 and later.
- *XKB* (http://www.x.org/wiki/XKB) may be useful.
- The utilities *gucharmap* and *kcharselect* let you access the full Unicode range on Ubuntu systems.

Revision History

Version 1.100 updates — Jan 7, 2026

Initial release.

Version 1.110 updates — Jan 9, 2026

Raise the Figure Dash **U+2012** in all 16 fonts. It was noticeably too low.

Version 1.120 updates — Jan 23, 2026

Repair the advance width of 7 finger diagrams. The advance widths of those diagrams were zero in some applications (e.g. Corel X8) – those that assume that a glyph with OpenType class **[Mark]** has a zero advance width.

Version 1.200 updates — Feb 13, 2026

Major upgrade to add additional “closure” symbols for finger holes in addition to “open” and “closed”: half-hole, two trill styles, “X”, don’t care open, don’t care closed, and a “ + ” symbol near the head of the finger diagram to indicate an overblow. This is implemented using zero-width OTF Mark characters that add the new symbols to a preceding OTF Base character that (typically) provides the flute outline.

In addition to .ttf files, .woff and .woff2 format files are now provided for use on Web pages.

About these Fonts

"With twenty-six soldiers of lead
I have conquered the world"
– James Mosley[2]

This project grew out of the intersection of:

- My need for a finger diagram font so I could easily document custom fingerings I developed for my Warbl2 to control various software synthesizers, samplers, sequencers, and other digital music goodies, and
- My experience developing Kurinto – a very large set of pan-Unicode fonts (see www.Kurinto.com – the image at the right was while researching historical writing systems in Myanmar).

Contemporary Myanmar and Historic Pyu writing systems at the Myanmar National Museum, Yangon, December 2018. Photo: Vera Shanov

The fonts comprise Latin and Cyrillic scripts from **Charis** v7.00 (June 2, 2025), general music-related characters from **Bravura Text** v1.393 (February 9, 2021) and **Cardo** v1.045 (October 24, 2005), and finger diagrams I developed for this project.

Some Details

Using the **Warbl** fonts should be straightforward, but there are a few things you should note:

Private Use Area

The finger diagrams all reside in the "Private Use Area" of the fonts – an area of code points set aside for characters that do not warrant inclusion in the Unicode standard because the size of their audience is limited. This Private Use Area uses code points in the range **U+E000** to **U+F8FF**. All the finger diagrams reside in this range.

What the Symbols Mean

Each of the finger diagrams – eg.  – have symbols that indicate how each of the finger holes on the Warbl is covered:

[2] James Mosley. *The Caslon foundry in 1902: selections from an album*, Matrix 13 (1993), pp. 34–42.

- The triangle at the top (or left) represents the thumb hole.
- The main six finger holes are represented by round circles.
- The smaller oval at the bottom represents the pinky hole.

Each finger hole typically shows as open or closed by these symbols:

 represents an open finger hole;

 represents a closed finger hole.

In addition, as of version 1.200 of this font package, these symbols are available:

 for a “half-hole” – a partially closed hole;

 for one style of showing a trill. Specifically, this style has a complete border circle with a small trill symbol in the center;

 for an alternate style of showing a trill: a trill symbol that extends entirely through the circle. You might prefer this style of trill, or you might think it looks like a “broken hole”;

 an “X” symbol that is one way to indicate “don’t care whether the hole is open or closed”;

 an open hole with a black dot in the center, intended to indicate “don’t care whether the hole is open or close, but where the normal or most convenient fingering is open”;

 a closed hole with a white dot in the center, intended to indicate “don’t care whether the hole is open or closed, but where the normal or most convenient fingering closed”.

There is also a ✚ symbol that can be added at the head of any finger diagram – for example:  – that is typically used to indicate an overblow.

Music Characters

All characters specified in the Standard Music Font Layout (SMuFL) are in each of the **Warbl** fonts. They were sourced from Open-Source Bravura font, which is the reference implementation of SMuFL. However, one block of characters from **U+EE00 – U+EFFF** was relocated to the range **U+FEE00 – U+FEFFF** in the Supplementary Private Use Area-A. You may access those (rarely used) musical characters from those code points.

Outlines

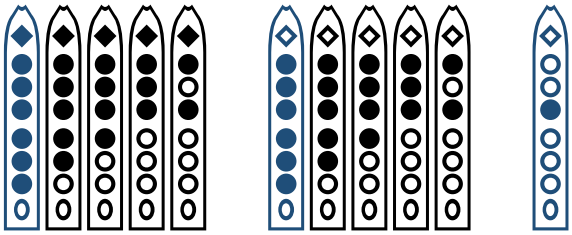
There are two types of fonts in this package: **Warbl** and **WarblC** fonts.

The **Warbl** fonts have outlines ( ) around the finger diagram that represent the shape of the flute.

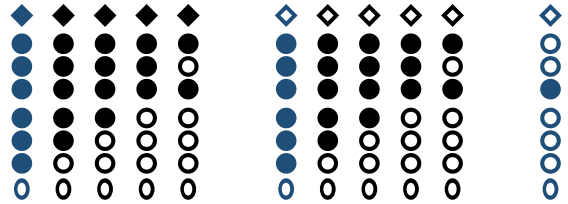
The **WarblC** fonts have no outlines ( ). The **WarblC** fonts have the same finger diagrams in the same code points and have the same spacing as **Warbl** fonts. The "C" at the end of the name stands for "Clean".

Personally, I use the **Warbl** fonts for all text with inline finger diagrams and tend to use **WarblC** for melody lines I write out in documents (my way to quickly write down a melody). Here are examples using 72 point text:

Warbl



WarblC



Orientation

All the fonts have both vertical finger diagrams ( ) and horizontal finger diagrams ( ). The vertical finger diagrams are great for scoring music, but are too small for use in-line in text. If you boost the point size of the vertical finger diagrams to something more readable, they wreak havoc with your page layout, because you have to make them in a much larger point size for them to be seen. For that reason, I typically use the horizontal finger diagram characters for in-line text.

The vertical finger diagrams are at Unicode code points **U+EE00 – U+EEFF** and the corresponding horizontal finger diagrams are at code points **U+EF00 – U+EFFF**.

Styles

Each of the **Warbl** and **WarblC** fonts come in 8 styles:

- Regular weight upright and *Italic characters*.

- Medium weight upright and *Italic characters*, which are slightly bolder than Regular.
- **SemiBold weight upright and *Italic characters*, which have even more boldness.**
- **Bold weight upright and *Italic characters*, which are the heaviest.**

NOTE that the musical symbols and finger diagrams in the fonts do **not** change style – just the characters used for natural language text.

TrueType fonts have traditionally been organized into families of four fonts: Regular, *Italic*, **Bold**, and ***Bold Italic***. These share a family name and work as expected with applications that have B and I buttons to switch between the styles. These are commonly referred to as 'RIBBI' families.

WARBL is an "axis-based" typeface, with four weights: Regular, Medium, SemiBold, and Bold. This gives you finer control for "how bold" you want the text to appear in a given situation.

Some applications, such as *InDesign*, *Google Docs*, and most macOS apps, list the single family name and then offer the full range of weight options.

Microsoft Word and some other Windows apps list weights (except Bold) as if they were separate families. This happens with Microsoft Word and some other applications on Windows. It can be confusing and cause problems. This can cause some significant heartburn:

However, there are some significant difficulties:

- The non-Bold weights are listed alphabetically rather than in weight order.
- It is often unclear what styles are available "natively" (i.e. from the font designer) and which styles will be "auto-generated" by the application.

The auto-generated styles tend to look terrible! They are sometimes call "Fake Bold" and "Fake Italic". For example:

This sentence is typeset in Warbl SemiBold. ***(Good!)***

This sentence is typeset in Warbl Bold. ***(Good!)***

This sentence is typeset in Warbl SemiBold, with Microsoft Word's [Bold] button selected. ***Notice how the characters have uneven spacing and appear "muddy".***

Some guidelines for applications such as *Microsoft Word*:

- If you want Regular, *Regular-Italic*, **Bold**, or ***Bold-Italic***, select **WARBL** or **WARBLC** and use the B and I buttons.
- If you want Medium or *Medium-Italic* (slightly heavier than Regular), then select **WARBL MEDIUM** or **WARBLC MEDIUM** and (optionally) use the I button. Avoid B.
- **If you want SemiBold or *SemiBold-Italic* (slightly heavier than Medium), then select WARBL SEMIBOLD or WARBLC SEMIBOLD and (optionally) use the I button.** Avoid B.

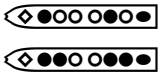
Finding the Fingering

The easiest way to get started entering finger diagrams into your document is to open `WarblFont_AllFingerDiagrams.docx`, supplied in the release package along with the fonts. You can then browse around for the character you want and then Copy/Paste it into your document. However, you do need to be able to run Microsoft Word for this to work.

With practice, you can type the code points directly into your document and then convert them (Alt-X in Microsoft Word) into a Unicode character. This takes a bit of practice with Hexadecimal (base 16 or "hex") numbers. Here is the general scheme:

- All the Unicode code points are four hex digits.
- The first two hex digits are **`EE`** for vertical finger diagrams and **`EF`** for horizontal finger diagrams.
- The last two hex digits represent 8 binary bits (or binary digits), which are 0 for "hole open" and 1 for "hole closed":
    - The first binary digit represents the thumb hole
    - Binary digits 2 through 4 represent the top 3 finger holes on the front of the instrument – typically fingered by three fingers of the player's upper hand.
    - Those 4 binary digits are combined into the third hex digit of the Unicode code point.
    - Binary digits 5 through 7 represent the next 3 finger holes on the front of the instrument – typically fingered by three fingers of the player's lower hand.
    - The last binary digit represents the bottom finger hole, typically covered by the player's pinky.
    - Binary digits 5 through 8 are combined into the last hex digit of the Unicode code point.

Binary	Hex	Binary	Hex
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

- Some examples:
    - **`EF01`**: 
    - **`EF23`**: 
    - **`EF45`**: 
    - **`EF67`**: 
    - **`EF89`**:

- **EFAB:**
- **EFCD:**
- **EFEF:**

More Exotic Fingerings

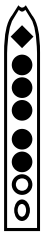
As of version 1.20 (Feb 13, 2026), this font package has characters for less-used finger hole symbols:

Each finger hole typically shows as open or closed by these symbols: , , , , , , and (see the section *What the Symbols Mean* above for more details). However, there are two different systems for including these symbols in a font diagram.

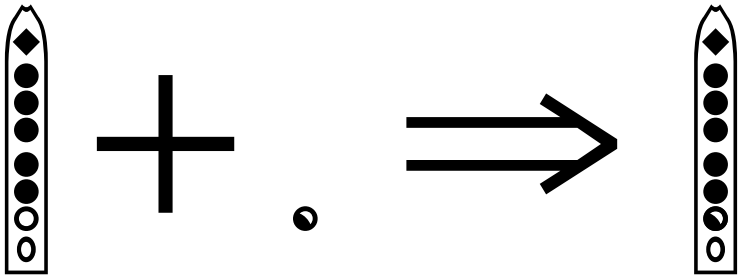
1. Modifying a Finger Diagram

The first system takes a finger diagram that is close to what you want and modifies it by overlaying a second character. For example, you could start with the base diagram and modify the second hole from the bottom with a to indicate a half-hole. The general approach is:

Get a Base character that has the open and closed fingerings you want, and has an open hole in the location you want the special symbol. In the example above that would be **U+EEFC**:



You then follow that Base character with a second “Add-On” character that modifies the open hole:

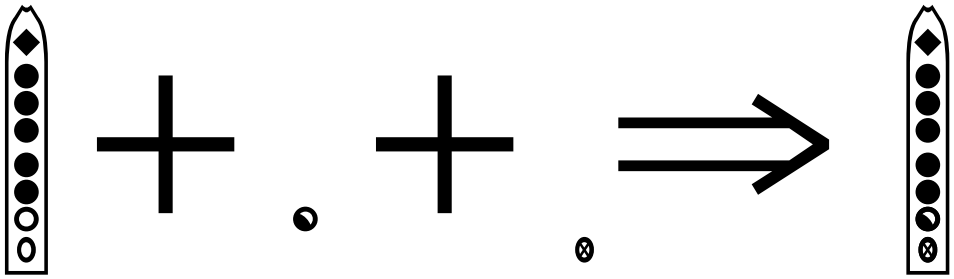


The Add-On characters are new characters that were added to these fonts in version 1.20 that let you add any of the symbols shown above to any finger hole. They have special versions of the symbols appropriate to the diamond-shaped thumb hole and oval pinky hole.

These Add-On characters ae special in that they modify the character that comes before them. Normally when you type text, the cursor location advances by some width (typically a different

width for wide characters such as “m” versus a narrow “i”). The Add-On characters do not advance the cursor at all and are called “Zero-width” or “Zero Advance Width” characters.

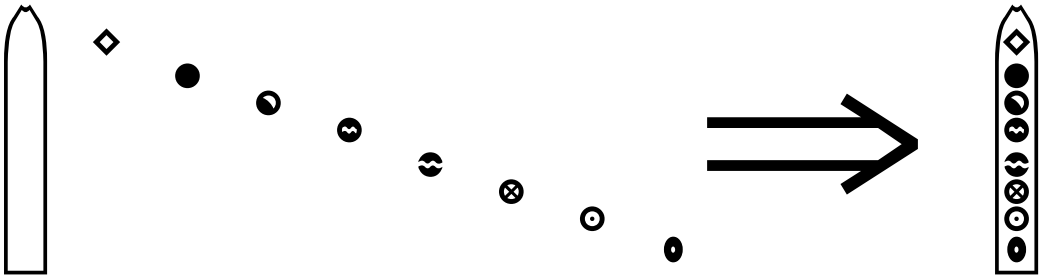
It is perfectly fine to add multiple Add-On characters. For example:



… However, there is an issue with this approach: Since you are laying the Add-On character over a 𝅝 open-hole symbol, it will always have the circular border of the open hole. Alternate finger holes that look like this: 𝅗𝅥 are not possible.

2. Composing a Finger Diagram from the Ground Up

The second system begins with a “blank” finger diagram (either one that just has the flute-shaped outline or one that is entirely blank) and adds all the 8 finger holes to compose a complete finger diagram. For example (the plus signs have been removed to save space):



While this example is extreme (and absurd), you can see that all combinations of finger diagrams are possible. However, typing such composite finger diagrams can be frustrating, because dealing with zero-width characters can be frustrating and counter-intuitive. However, with practice and experimentation (and a willingness to delete and re-try a failed attempt from scratch), you can get reliable and fast results.

Generating PDF Files

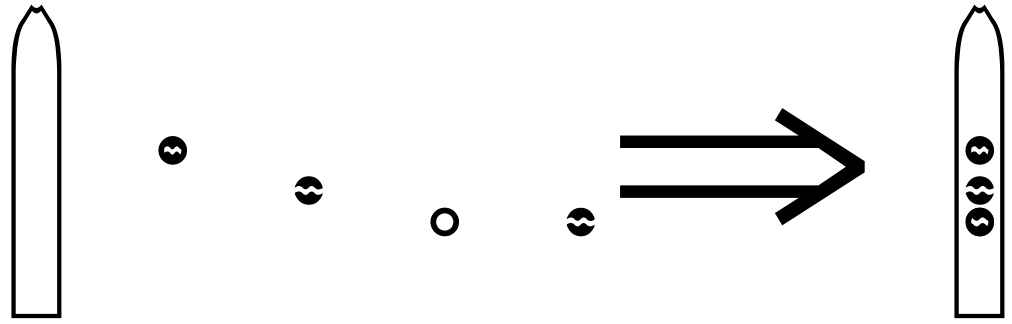
The use of zero-width characters, although common in digital typography for a ***long*** time, is not always handled well by mainstream software. Saving Microsoft Word files to PDF is one area of long-standing issues.

Rather than executing **`Save As`** from Microsoft Word when generating a PDF, I use the **`Print`** command and then use the **`Microsoft Print to PDF`** driver rather than the **`Adobe PDF`** printer

driver. The **Microsoft Print to PDF** files are noticeably larger than the **Adobe PDF** files, but the rendering engine appears far more precise and stable.

Trill Symbols

The symbols that denote trills deserve special note: There are actually three different possibilities, demonstrated by this composed finger diagram with just three finger holes:



- The top finger hole is an Add-On character.
- The second finger hole is another Add-on Character.
- The third finger hole is actually the same Add-on character from the second finger hole, laid on top of an open-hole diagram. It is similar to the first example, but somewhat different if you look at it up close.

The Add-On Roadmap

Here is a roadmap of the code points in all the **Warbl** fonts for the add-on characters:

- $F000–$F0FF: Support characters for **vertical** finger diagrams, including characters that indicate special closures for individual finger holes:
    - $F000–$F07F: Vertical finger hole closure characters. They can convert an open finger hole into another kind of hole (half-hole, trill, etc.), or they can provide the entire image for that finger hole. These characters are designed to either:
        1. follow a regular finger diagram (in the range $EE00 – $EEFF) and replace one or more open finger holes with other kinds of finger hole closures, or
        2. follow a bare flute outline (e.g. $F0A0) or empty character (e.g. $F0A1) and supply **all** 8 finger hole closures. This method allows an arbitrary mix of finger hole closures and has more flexibility.

      The general format of the code point is $F0**H****T**, where **H** gives the hole number and **T** gives the style of hole coverage. More specifically:

      - **H** is 0 for the thumb hole and 1 through 7 are for the seven finger holes on the front side of the instrument.
      - **T** is:

- 0 for an open finger hole.
- 1 for a closed finger hole.
- 2 for half-hole.
- 3 for one style of showing a trill. Specifically, this style has a complete border circle with a small trill symbol in the center.
- 4 for an alternate style of showing a trill: a trill symbol that extends entirely through the circle. This style may be preferably graphically, but cannot be printed over an open-hole symbol, so it requires that the finger diagram be composed of a flute outline and all eight finger hole symbols.
- 5 for an "X" symbol that is one way to indicate "don't care".
- 6 for an open hole with a black dot in the center, intended to indicate "don't care with the normal or most convenient fingering open".
- 7 for a closed hole with a white dot in the center, intended to indicate "don't care with the normal or most convenient fingering closed".
- 8 through F are unused.

Note that **T** = 0 and **T** = 1 (open hole and closed hole) may seem redundant since there are code points for finger diagrams with all the open hole and closed hole possibilities. However, these glyphs allow construction of a finger diagram on a hole-by-hole basis.

- $F080–$F088: Large, display versions of each flavor of finger hole covered (open, closed, half-hole, two trill styles, "X", don't care open, don't care closed, and "+") suitable for use in documentation about this font. These show the predominant round finger hole style (as opposed to the triangular thumb hole or oval pinky hole).

- $F090–$F098: Vertical finger diagrams with flute outlines where all the finger holes in each character have the same closure style (open, closed, half-hole, two trill styles, "X", don't care open, and don't care closed). The last character is an outline with a "+".

- $F0A0: A character with a vertical flute outline and no finger holes. This character has a standard advance width. This can be used to create a finger diagram with a flute outline "from scratch" by supplying following characters for each of the finger holes.

- $F0A1: Vertical add-on "+". They have no outline or finger holes. These code points have zero advance width. The "+" sits at the top of the finger diagram and indicates an overblow.

- $F0B0–$F0B7: Vertical finger diagrams with no flute outline where all the finger holes have the same closure style (open, closed, half-hole, two trill styles, “X”, don’t care open, and don’t care closed).

- $F0C0: A character with a no flute outline and no finger holes. This is similar to $F0A0, but is designed for building a “clean” finger diagram from scratch with no flute outline. This character appears like a space character, but has the standard advance width of a finger diagram.

- $F0C1: Vertical add-on “+”. They have no outline or finger holes. These code points have zero advance width. The “+” sits at the top of the finger diagram and indicates an overblow. This is a copy of the character at $F0A1.

- $F0F0–$F0F3: Utility glyphs from the Charis fonts (formerly $F000–$F003).

- $F100–$F1FF: Support characters for **horizontal** finger diagrams. These are analogous to the characters for vertical finger diagrams, but are larger, horizontal, and designed for in-line use in text.

- $F200–$F3FF: Unused.

- $F400–$F8FF: Music symbols copied from Bravura.

Frequently Asked Questions

Licensing

Is it free?

Yes. The **WARBL** fonts are free like beer and free like liberty. Enjoy!

I want to use Warbl fonts in my publication – can I?

Yes. **WARBL** is released under the *SIL Open Font License Version 1.1 (“the OFL”)*, which permits use in any publication, whether electronic or printed. For more answers question about using the fonts, see `FAQ-OFL.txt` or visit https://scripts.sil.org/ofl-faq_web. The text of the *OFL* is distributed in each release package in the `OFL.txt` file.

I would like to bundle Warbl fonts with my application – can I?

Yes. The *SIL Open Font License Version 1.1* allows bundling of the licensed fonts with applications, subject to some restrictions. This permission to bundle applies even for commercial, closed-source applications. See the `OFL.txt` and the `FAQ-OFL.txt` files.

Can I use the font on my web site?

Yes. You can create web pages that use **Warbl** fonts to display text. This works both if that font is already available on the user's system or if it is delivered via CSS directives such as `@font-face`.

According to the *SIL Open Font License Version 1.1*, you may also place the **Warbl** fonts on your server for people to download. There is further discussion of web-font issues in the `FAQ-OFL.txt` file.

It is best to use the `.woff2` or (if necessary) the `.woff` versions of the font files, which are designed for use on the Web.

Is Warbl going to stay unrestricted and available at no cost?

There is no intention to ever charge users for using **Warbl** and its variants. The current version is licensed under a free/open license and the intent is that future versions will be similarly unencumbered.

Can I send you money?

No, but thank you for the thought. Please consider supporting the Unicode Consortium at http://unicode.org/consortium/donations.html.

Why does the free, open-source license for Warbl have a copyright statement?

A central concept of the Open Source movement is how it uses existing copyright law to guarantee the free and open access to and redistribution of software. This may seem counterintuitive, since copyrights have traditionally been used to restrict the rights of access and redistribution.

Warbl fonts are accessible to anyone, in exchange for abiding by the terms of the *OFL*. The *OFL* uses copyright law to guarantee that no party may restrict or curtail anyone else's right to copy, use, modify, and redistribute the font software. A party that violates the terms of the *OFL* (for example, by restricting the rights of others) loses their rights under the License, placing them in violation of copyright law, and opens the possibility of prosecution.

To further promote open collaboration and development, the guarantees of the *OFL* extend to the source code of the software.

Why isn't this software simply released into the Public Domain?

While it might appear that releasing software into the public domain would be a more noble approach, this path tends not to serve the community as well as the *OFL*.

- The concept of public domain is not universally recognized.
- The mechanism of making a public domain declaration is problematic in many situations, and even illegal in some countries.
- The authors cannot apply a warranty clause and open themselves to litigation by making the code available under a Public Domain declaration.

- Public domain places no restrictions on use. This freedom allows anyone to apply legal restrictions that prevent others (even the original authors) from using the software. Anyone may even claim authorship of the software, making the history of its development unclear. There are many notable cases of these perverse situations involving public domain content.
- Many commercial enterprises avoid using public domain content, because it opens **them** to subsequent litigation by others who have subsequently applied copyright and trademark restrictions.

For these reasons, I am using the well-established tradition of the *OFL* to promote continued, open development of **Warbl**.

Modification

I would like to modify Warbl to add a couple of characters I need. Can I?

Yes. Modifications are allowed, as long as you abide by the conditions of the *SIL Open Font License Version 1.1*.

Technical

Can you help me get Warbl fonts working on my system?

I cannot afford to offer individual technical support. However, we do want to hear of any problems you encounter, so that I can add them to the list of bugs to fix in later releases. Our contact address is clint@goss.com. Please understand that I cannot guarantee a personal response.

How do I report issues with these fonts?

If you have issues, suggestions, or ideas about the **Warbl** Font Folio, head over to the GitHub repository for reporting and discussing issues at https://GitHub.com/ClintGoss/Warbl.

Windows Font Installation

The font installation instructions for Windows 10 will work in most cases. However, users who repeatedly install and uninstall fonts may encounter frustrating situations:

- Fonts files are loaded into `/Windows/Fonts`, but do not show up in an application's font list.
- Font installation says that a font is already installed when it is not, and clicking `Yes` to replace the font fails to install the font.
- You find many duplicated fonts in `/Windows/Fonts` with extensions such as `_0`, `_1`, …
- *Windows* shows only some of the installed fonts in a typeface.
- And (**particularly** frustrating) your application winds up accessing the **wrong version** of an installed font.

There are many Web-based sources of information on these situations which provide more detailed and timely information than I can cover in this document. However, here are some basic things to note:

- Current versions of *Windows* use a completely different directory structure for fonts that are installed for a single user versus for all users.
- When fonts are installed, *Windows* copies the font file into a system folder and also updates a list of available fonts in the *Windows* registry. Many bizarre issues arise if the list of fonts in the registry gets out of step with the font files in the system folder.
- Windows does not delete superseded fonts – it renames the font files. It can also restore previously "replaced" fonts, making the job of actually deleting a font difficult.
- Fonts listed in the registry with incorrect information (such as referring to a `.ttf` font when the actual file name is `.otf`) can render the font inaccessible.

Here are some pointers to resources that I have come across:

- *"Fonts Seem to Install, but Don't"* - https://forums.adobe.com/thread/951333
- *How to Delete Fonts using the Windows Registry* – https://www.maketecheasier.com/delete-fonts-using-windows-registry/
- *What is the purpose of the Fonts key in the Registry?* – https://superuser.com/questions/813039/what-is-the-purpose-of-the-fonts-key-in-the-registry

In addition, the font installation instructions for Windows 10 will **only** work if the default viewer for fonts is set to *Windows Font Viewer*. If the default view has been changed to another application, the context menu items Install and Install for all users will not be available.

References

[CDS 2018] Microsoft Typography, *Character Design Standards*, retrieved February 2, 2019 from https://docs.microsoft.com/en-us/typography/develop/character-design-standards/.

[Evans 2013] Lorna Evans, *Recommended Characters for Non-Roman Fonts*, retrieved January 21, 2018 from https://scriptsource.org/entry/gg5wm9hhd3. See also subsequent comments and updates on the cited web page that modify the recommendations of the posted document.

[FF 2017] FontForge Community, *Design with FontForge – A Book About How to Create New Typefaces Using FongForge*, 205 pages, 2017, retrieved May 19, 2020 from http://designwithfontforge.com/ebook/design-with-fontforge_en-US.pdf.

[GPO 2016] U.S. Government Publishing Office, *Style Manual: An Official Guide to the Form and Style of Federal Government Publications*, 2016, 461 + xiii pages. ISBN 978-0-16-093601-2, retrieved from

https://www.govinfo.gov/content/pkg/GPO-STYLEMANUAL-2016/pdf/GPO-STYLEMANUAL-2016.pdf on April 9, 2020.

[Kellman 2016] Steven G. Kellman, *Omnilingual Aspirations: The Case of the Universal Declaration of Human Rights*, Critical Multilingualism Studies, Volume 4, Number 1 (2016), pages 5–24.

[PRC-Edu 2013] Ministry of Education of the People's Republic of China, 通用规范汉字表 «Table of General Standard Chinese Characters», published June 18, 2013, http://www.gov.cn/gzdt/att/att/site1/20130819/tygfhzb.pdf. See also https://en.wikipedia.org/wiki/Table_of_General_Standard_Chinese_Characters.

[SIL 2020] SIL / NRSI, *Font Development Best Practices, Technical Guidance Regarding Font Development And Production*, https://silnrsi.github.io/FDBP/en-US/, retrieved March 15, 2020.

[Unicode 2019] The Unicode Consortium, *The Unicode® Standard, Version 12.0 – Core Specification*, March 2019, 1018 + xxviii pages, ISBN 978-1-936213-22-1, http://www.unicode.org/versions/Unicode12.0.0/.