

# Church Slavonic Fonts

Aleksandr Andreev      Yuri Shardt      Nikita Simmons\*

April 26, 2016      version 1.0

(PDF file generated on May 2, 2016)

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>License</b>	<b>3</b>
2.1	The Legal Text . . . . .	3
<b>3</b>	<b>Introduction</b>	<b>4</b>
<b>4</b>	<b>Installation and Usage</b>	<b>4</b>
4.1	Font Formats . . . . .	4
4.2	Source Packages . . . . .	6
<b>5</b>	<b>System Requirements</b>	<b>6</b>
5.1	Microsoft Windows . . . . .	6
5.2	GNU/Linux . . . . .	6
5.3	OS X . . . . .	7
<b>6</b>	<b>Private Use Area</b>	<b>7</b>
<b>7</b>	<b>OpenType Technology</b>	<b>8</b>
7.1	On Microsoft Windows . . . . .	8
7.2	On GNU/Linux . . . . .	8
7.3	OpenType Features . . . . .	9
7.3.1	Combining Mark Positioning . . . . .	9

---

\*Comments may be directed to [aleksandr.andreev@gmail.com](mailto:aleksandr.andreev@gmail.com).

7.3.2	Glyph Composition and Decomposition . . . . .	9
7.3.3	Language-based Features . . . . .	10
7.3.4	Stylistic Alternatives and Stylistic Sets . . . . .	10
<b>8</b>	<b>SIL Graphite Technology</b>	<b>11</b>
8.1	Graphite in LibreOffice . . . . .	11
8.2	Graphite in X <sub>Y</sub> TeX . . . . .	12
<b>9</b>	<b>Ponomar Unicode</b>	<b>13</b>
9.1	Synodal Church Slavonic . . . . .	13
9.2	Kievan Church Slavonic . . . . .	13
9.3	Other Languages . . . . .	13
9.4	Font Features . . . . .	15
9.5	SIL Graphite Features . . . . .	17
<b>10</b>	<b>Fedorovsk Unicode</b>	<b>18</b>
10.1	Sample Texts . . . . .	18
10.1.1	Apostol of Ivan Fedorov . . . . .	18
10.1.2	Flowery Triodion . . . . .	19
10.2	OpenType Features . . . . .	19
10.3	Graphite Features . . . . .	21
<b>11</b>	<b>Menaion Unicode</b>	<b>22</b>
11.1	Sample Texts . . . . .	22
11.2	Provided Ligatures . . . . .	22
<b>12</b>	<b>Pomorsky Unicode</b>	<b>22</b>
12.1	Sample Texts . . . . .	27
<b>13</b>	<b>Known Issues</b>	<b>27</b>
<b>14</b>	<b>Credits</b>	<b>27</b>

## 1 Introduction

Church Slavonic (also called Church Slavic, Old Church Slavonic or Old Slavonic; ISO 639-2 code cu) is a literary language used by the Slavic peoples; presently it is used as a liturgical language by the Russian Orthodox Church, other local Orthodox Churches, as well as various Byzantine-Rite Catholic and Old Ritualist

communities. The package `fonts-churchslavonic` provides fonts for representing Church Slavonic text.

The fonts are designed to work with Unicode text encoded in UTF-8. Texts encoded in legacy codepages (such as HIP and UCS) may be converted to Unicode using a separate bundle of utilities. See the [Slavonic Computing Initiative website](#) for more information.

## 2 License

The fonts distributed in this package are dual-licensed under the GNU General Public License (version 3 or later) and the SIL Open Font License (version 1.1 or later). The SIL Open Font License is preferred, since this is a FLOSS license intended for fonts. Dual licensing under GNU GPL is maintained to allow embedding of these fonts into GPL-licensed applications and for compatibility with other projects.

### 2.1 The Legal Text

The fonts distributed in this package are free software: you can redistribute them and/or modify them, in whole or in part, EITHER under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version OR under the terms of the SIL Open Font License, version 1.1, or (at your option) any later version, without reserved font names.

As a special exception, if you create a document which uses any of these fonts, and embed the font or unaltered portions of the font into the document, the font does not by itself cause the resulting document to be covered by the GNU General Public License. This exception does not however invalidate any other reasons why the document might be covered by the GNU General Public License. If you modify any of these fonts, you may extend this exception to your version of the fonts, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

As free software, these fonts are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License or the SIL Open Font License for more details.

This document is licensed under the Creative Commons Attribution-ShareAlike

4.0 International License. To view a copy of this license, visit the [Creative Commons website](#).

### 3 Introduction

The package provides several fonts that are intended for working with Church Slavonic text of various recensions and other texts related to Church Slavonic: modern Church Slavonic text (“Synodal Slavonic”), historical printed Church Slavonic text and manuscript uncial (ustav) Church Slavonic text (in either Cyrillic or Glagolitic), as well as text in Sakha (Yakut), Aleut (Fox Island dialect), and Romanian (Moldovan) Cyrillic, all written in the ecclesiastical script. The coverage of the various fonts agrees with the guidelines for font coverage specified in [Unicode Technical Note #41: Church Slavonic Typography in Unicode](#). Generally speaking, it includes most (but not all) characters in the Cyrillic, Cyrillic Supplement, Cyrillic Extended-A, Cyrillic Extended-B, Cyrillic Extended-C (as of Unicode 9.0), Glagolitic, and Glagolitic Supplement blocks of Unicode. Characters not used in Church Slavonic, however, are not included (except for some characters used in modern Russian, Ukrainian, Belorussian, Serbian and Macedonian for purposes of compatibility with some applications).

### 4 Installation and Usage

If you are reading this document, then you probably have already downloaded the font package. You may check if you have the most recent version by visiting the [Slavonic Computing Initiative website](#).

#### 4.1 Font Formats

All fonts are currently available in two formats:



TrueType fonts, or, more precisely, [OpenType](#) fonts with TrueType outlines;



[OpenType](#) fonts with PostScript outlines (also called OpenType-CFF fonts).

Note that fonts in these two formats have different file extensions: \*.ttf for TrueType, \*.otf for OpenType-CFF fonts. Both the TrueType version and the OpenType-CFF version support the same set of advanced [OpenType](#) features.

The OpenType-CFF fonts use PostScript outlines, based on third-order (cubic) Bézier curves, while the TrueType fonts use second-order (quadratic) curves.

There is also a significant difference in hinting (grid fitting): TrueType instructions theoretically allow to achieve much better quality of screen rendering than PostScript hinting. However, since quality hinting is a very difficult and time-consuming process, both the PostScript hinting and TrueType instructing of the fonts has been done automatically, so high quality grid fitting is not available.

Note that it is possible to install both the TrueType and OpenType-CFF versions simultaneously. For this purpose, the TrueType fonts contain a “TT” suffix in their font name/family name fields. Since all of the fonts have been drawn in cubic splines (and then converted to quadratic for the TTF version), and since the TrueType instructions have been automatically generated, the OpenType-CFF format may theoretically give you better screen rendering quality, though in most situations this will not be noticeable. Furthermore, note that only the TTF version supports [SIL Graphite](#), so you will need to use the TrueType fonts if Graphite support is desired. The following considerations are also in order:

- In older versions of OpenOffice.org, OpenType-CFF fonts were not properly embedded into PDF files. Moreover, under Unix-based systems, OpenOffice.org could not access such fonts at all, so using TTF versions was the only option. This was fixed in OpenOffice.org 3.2 and LibreOffice.
- OpenOffice.org and LibreOffice, however, still have no mechanism to turn off and on advanced OpenType features, so if you plan to use optional typographic features, you will need to use SIL Graphite, which is only available in the TTF version.
- OpenType-CFF fonts are poorly supported in the Sun Java Development Kit, so we recommend use of the TTF versions in Java programming situations and on Android devices.
- On Microsoft Windows, OpenType glyph positioning is not supported for glyphs in the Private Use Area or characters outside of the Unicode 7.0 range. You will need to rely on [SIL Graphite](#) (only available in the TTF versions) for positioning of combining Glagolitic characters and various glyphs in the PUA.

Note that Microsoft Windows checks the presence of a digital signature in a TrueType font, considering this would allow to distinguish “old” TrueType fonts from “modern” OpenType fonts with TrueType outlines. The fonts in this package contain a dummy digital signature in order to fool Microsoft products into allowing use of additional TrueType features.

## 4.2 Source Packages

You can also download the FontForge sources for all of the fonts from the [GitHub repository](#). This is only useful if you are planning on editing the fonts in the [FontForge](#) font editor. In general, you will not gain any productivity improvements from rebuilding the font files, so rebuilding from source is not recommended, unless you have a real need to modify the fonts, for example, to add your own additional glyphs to the Private Use Area.

## 5 System Requirements

All of these fonts are large Unicode fonts and require a Unicode-aware operating system and software environment. Outside of a Unicode-aware environment, you will only be able, at most, to access the first 256 glyphs of a font.

### 5.1 Microsoft Windows

Unicode has been supported since Windows 95, however to use the OpenType-CFF version of the fonts, you need at least Windows 2000. You will need a word processor that can handle Unicode-based documents, such as Microsoft Word 97 and above, or [LibreOffice](#). Please note that maintenance of OpenOffice.org has been discontinued, so we recommend using LibreOffice instead. If using  $\text{\TeX}$ , you will need a Unicode-aware  $\text{\TeX}$  engine, such as  $\text{\XeTeX}$  or  $\text{\LuaTeX}$ .

You will also need a way to enter the Unicode characters that are not directly accessible from standard keyboards. We recommend installing a Church Slavonic or Russian-Extended keyboard layout, available from the [Slavonic Computing Initiative website](#). It is also possible to enter characters using the Windows Character Map utility or by codepoint, but this is not recommended.

### 5.2 GNU/Linux

In order to be able to handle TrueType or OpenType fonts, your system should have the [freetype](#) library installed and enabled; this is normally done by default in all modern distributions. You will need a Unicode-aware word processor, such as [LibreOffice](#). Please note that OpenOffice.org is no longer maintained, so we recommend using LibreOffice instead. If using  $\text{\TeX}$ , you will need a Unicode-aware  $\text{\TeX}$  engine, such as  $\text{\XeTeX}$  or  $\text{\LuaTeX}$ .

You will need a keyboard driver to input Unicode characters. Under GNU/Linux,

this is handled by the m17n library and database. See the [Slavonic Computing Initiative website](#) for more details.

### 5.3 OS X

Not sure.

## 6 Private Use Area

The Unicode Private Use Area (PUA) is a set of three ranges of codepoints (U+E000 to U+F8FF, Plane 15 and Plane 16) that are guaranteed to never be assigned to characters by the Unicode Consortium and can be used by third parties to allocate their own characters. The Slavonic Computing Initiative has established an industry standard for character allocation in the PUA, which is described in full in the [PUA Allocation Policy](#).

The PUA in these fonts contains various additional glyphs: contextual alternatives, stylistic alternatives, ligatures, hypothetical and nonce glyphs, various glyphs not yet encoded in Unicode and various technical symbols. Most of these glyphs (the alternative glyphs and ligatures) are accessible via [OpenType](#) and [SIL Graphite](#) features. Thus, you generally do not need to access glyphs in the PUA directly. There may be some exceptions:

- If you need to access characters not yet encoded in Unicode and nonce glyphs.
- If you need to access alternative glyphs and ligatures on legacy systems that do not support OpenType or Graphite features.
- If you are a computer programmer and need to work with glyphs on a low level without relying on OpenType: having all alternatives mapped to the PUA allows for a simple way to access glyphs by codepoint instead of working with glyph indexes, which can change between versions of a font.

For the characters mapped in the PUA and other technical considerations, please see the [PUA Allocation Policy](#).

## 7 OpenType Technology

OpenType is a “smart font” technology for advanced typography developed by Microsoft Corporation and Adobe Systems and based on the TrueType font format. It allows for correct typography in complex scripts as well as providing advanced typographic effects. This is achieved by applying various *features*, or *tags*, described in the OpenType specification. Some of these features are supposed to be enabled by default, while others are considered optional, and may be turned on and off by the user when desired.

### 7.1 On Microsoft Windows

In order to use these advanced typographic features, in addition to a “smart” font (like the fonts in this package), you need an OpenType-aware application. Not all applications currently support OpenType, and not all applications that claim to support OpenType actually support all features or provide an interface to access features. Older versions of Microsoft’s Uniscribe library did not support OpenType features for Cyrillic and Glagolitic, but beginning with Windows 7, this has been resolved.

Generally speaking, you will get best results in X<sub>Y</sub>TeX or LuaTeX using the fontspec package or using advanced desktop publishing software such as Adobe InDesign. Most OpenType features are also accessible in Microsoft Office 2010 and later (but see below for details). LibreOffice also supports OpenType features starting with version 4.1, however provides no method to turn on optional features. Please see the section on [SIL Graphite](#), below.

### 7.2 On GNU/Linux

OpenType support is provided by the HarfBuzz shaping library, which is accessible through FreeType, part of most standard distributions of the X Window System. Thus, OpenType will be available in any application that uses FreeType, though many applications lack an interface to turn on and off optional features. Generally speaking, you will get best results in X<sub>Y</sub>TeX or LuaTeX using the fontspec package. LibreOffice also supports OpenType features starting with version 4.1, however provides no method to turn on optional features. Please see the section on [SIL Graphite](#), below.



## 7.3 OpenType Features

### 7.3.1 Combining Mark Positioning

OpenType allows smart diacritic positioning: if you type a letter followed by a diacritic, the diacritic will be placed exactly above or below the letter; this is provided by the *mark* feature. In addition, the *mkmk* feature is used to position two marks with respect to each other, so that an additional diacritic can be stacked properly above the first. This behavior is demonstrated below:

$$\begin{array}{ll} \text{А} + \text{◌}^{\text{́}} \rightarrow \text{А}^{\text{́}} & \\ \text{Ѧ} + \text{◌}^{\text{́}} \rightarrow \text{Ѧ}^{\text{́}} & \text{(glyph positioning via } \textit{mark} \text{ feature)} \\ \text{◌}^{\text{́}} + \text{◌}^{\text{̂}} \rightarrow \text{◌}^{\text{̂́}} & \text{(glyph positioning via } \textit{mkmk} \text{ feature)} \end{array}$$

The fonts provide proper mark and mkmk anchor points for all Cyrillic and Glagolitic letters and combining marks, allowing you to enter them in almost any combination (even those that are implausible). Most OpenType renderers (except older versions of Adobe’s Cooltype library) support these features, so you should be able to achieve correct positioning in most OpenType-aware applications (for example, in MS Word 2010 or newer, LibreOffice 4.1 or newer, and X<sub>2</sub>TeX).

### 7.3.2 Glyph Composition and Decomposition

The Glyph Composition / Decomposition (*ccmp*) feature is used to compose two characters into a single glyph for better glyph processing. This feature is also used to create precomposed forms of a base glyph with diacritical marks when use of only *mark* and *mkmk* cannot achieve the necessary positioning. It is also used to create alternative glyph shapes, such as the alternative version of the Psili used over capital letters and the truncated forms of the letter Uk used with accent marks, as is demonstrated in the examples below:

$$\begin{array}{ll} \text{◌}^{\text{́}} \rightarrow \text{◌}^{\text{́}} & \text{(glyph substitution using } \textit{ccmp} \text{ feature)} \\ \text{◌}^{\text{́}} + \text{◌}^{\text{̂}} \rightarrow \text{◌}^{\text{̂́}} & \text{(ligature substitution using } \textit{ccmp} \text{ feature)} \\ \text{Ѧ} + \text{Ѧ}^{\text{́}} + \text{Ѧ}^{\text{̂}} \rightarrow \text{Ѧ}^{\text{̂́}} & \text{(ligature substitution using } \textit{ccmp} \text{ feature)} \\ \text{Ѧ} + \text{Ѧ}^{\text{́}} \rightarrow \text{Ѧ}^{\text{́}} & \text{(contextual substitution using } \textit{ccmp} \text{ feature)} \end{array}$$

Generally speaking, the *ccmp* feature is not supposed to (and often just cannot) be turned off, and thus this functionality should work properly in any OpenType-aware application. For more details on ligatures, see [Unicode Technical Note #41: Church Slavonic Typography in Unicode](#).

### 7.3.3 Language-based Features

Language-based features such as the `loca` (localized forms) feature provide access to language-specific alternate glyph forms, such as the alternate forms of the Cyrillic Letter I used in Ukrainian and Belorussian:

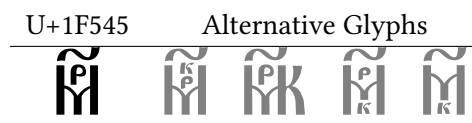
І	(Church Slavonic text)
і	(Ukrainian text)

To make use of these features, you need an OpenType-aware application that supports specifying the language of text, for example  $\text{X}\text{\TeX}$  or  $\text{L}\text{\TeX}$  using the `fontspec` or `polyglossia` packages. Since many software applications do not allow you to specify Church Slavonic as a language of text, it is assumed by default that the font is being used to represent Church Slavonic text, and thus all glyphs have Church Slavonic appearances unless another language is specified.

LibreOffice allows you to specify that text is in Church Slavonic starting with version 5.0. This will allow you to take advantage of other features, such as Church Slavonic hyphenation (see the [Slavonic Computing Initiative website](#) for more information). Microsoft Corporation does not recognize Church Slavonic as a valid language, so you will not be able to set the language of text to Church Slavonic in any Microsoft product.<sup>1</sup>

### 7.3.4 Stylistic Alternatives and Stylistic Sets

Stylistic Alternatives (`salt` feature) provide variant glyph shapes that may be selected by the user at will. Typically, these are glyphs that differ from the base glyph only in graphical appearance where the use of these glyphs does not follow any language-based or typography-based rules, but rather is just an embellishment. For example, the following variant forms of U+1F545 Symbol for Marks Chapter are provided:



Stylistic sets are used to enable a group of stylistic variant glyphs, designed to harmonize visually, and make them automatically substituted instead of the

<sup>1</sup>Please do not contact the font maintainers about this issue. Instead, complain to Microsoft Customer Service in the USA at 1-800-642-7676 or in Canada at +1 (877) 568-2495.

default forms. OpenType allows to specify up to 20 stylistic sets, marking them features `ss01`, `ss02`, ... `ss20`.

Use of Stylistic Alternatives and Stylistic Sets requires an OpenType-aware application that provides an interface to turn off and on advanced features (since by default these features are turned off). This is possible in  $\text{\XeTeX}$  or  $\text{\LuaTeX}$  using the `fontspec` package and in advanced desktop publishing software such as Adobe InDesign. Microsoft Office and LibreOffice currently have no way to access Stylistic Alternatives and Stylistic Sets. If necessary, you may access these glyphs by codepoint from the Private Use Area, though this is not recommended.

## 8 SIL Graphite Technology

[Graphite](#) is a “smart font” technology developed by [SIL International](#). Since, unlike OpenType, Graphite does not have predefined features, it provides the developer with an ability to control subtle typographic features that may be difficult or impossible to handle with OpenType. In fact, Graphite is in some respects more powerful than OpenType, though this additional power is not necessary for standard Church Slavonic typography. In addition, while support of OpenType features often varies from application to application, Graphite relies on a single engine, and thus all Graphite features are supported whenever an application supports Graphite. However, Graphite is not supported widely: in addition to SIL’s own [WorldPad](#) editor (a Windows-only application that requires a .NET runtime), Graphite is supported in LibreOffice (starting with OpenOffice.org version 3.2), Mozilla Firefox (starting with version 11), and  $\text{\XeTeX}$  (starting with version 0.997). Graphite support is not available in Microsoft Office.

Note that it is currently not possible to add Graphite tables to OpenType-CFF fonts. Therefore Graphite is only supported in the TrueType versions of fonts. Mostly Graphite will be of interest to LibreOffice users, since LibreOffice does not provide any interface to turn off and on OpenType features.

### 8.1 Graphite in LibreOffice

LibreOffice (and all OpenOffice.org derivatives starting with version 3.2) automatically recognizes fonts which contain Graphite tables. For such fonts Graphite rendering is enabled by default. Correct positioning, attachment and substitutions will work automatically. However, there is currently no graphical interface that can be used to turn additional features on and off. Instead, a special extended font name syntax has been developed: in order to activate an optional

feature, its ID, followed by an equals sign and the ID of the desired setting, are appended directly to the font name string. An ampersand is used to separate different feature/settings pairs.

For example, the following “font” should be used in order to get alternative glyphs for U+1F545 Symbol for Marks Chapter:

```
Ponomar Unicode TT:mark=1
```

for the first alternate glyph, mark=2 for the second alternate glyph, and so forth.

Of course modifying the font name directly is very inconvenient, since it is difficult to remember short tags and numerical values used for feature/setting IDs in different fonts. You may try to install the [Graphite Font Extension](#), which provides a dialog for easier feature selection. However, this extension has not been maintained since the passing of its developer in 2011, and so may not work correctly in later versions of LibreOffice. If you experience problems with Graphite features, you may get better results accessing glyphs directly by codepoint from the Private Use Area, though this is not recommended.

## 8.2 Graphite in X<sub>Y</sub>TeX

Graphite support is available in X<sub>Y</sub>TeX, which means Graphite features are now accessible from TeX documents. Moreover, it is possible to enable the Graphite font renderer with the `fontspec` package, which greatly simplifies selecting system-installed fonts in X<sub>Y</sub>TeX and LuaTeX. This functionality requires at least TeX Live 2010 or MikTeX2.9.

You can activate the Graphite rendering mode for a particular font via the `Renderer` option (its value should be set to `Graphite`) in the argument list of a font selection command. Since there are no standard feature tags in Graphite, the feature identifiers and their settings are passed to the `RawFeature` option as follows:

```
\newfontfamily{\graphA}  
  [Renderer=Graphite, RawFeature=  
    {Symbol for Mark's Chapter=Alternative 1}]  
{Ponomar Unicode TT}
```

Please consult the `fontspec` documentation for more information.

## 9 Ponomar Unicode

Ponomar Unicode is a font that reproduces the typeface of Synodal Church Slavonic editions from the beginning of the 20th Century. It is intended for working with modern Church Slavonic texts (Synodal Slavonic). Ponomar Unicode is based on the Hirmos UCS font designed by Vlad Dorosh, but has been modified by the authors of this package. Examples of text set in Ponomar Unicode are presented below.

## 9.1 Synodal Church Slavonic

Бл҃женъ мѡжъ, ѿже не ѿде на совѣтъ нечестѣвыхъ, ѿ на пѣти грѣшныхъ не стѣ, ѿ на сѣдѣльщи гл҃вителей не сѣде: но въ законѣ гд҃ни бола ѿгнѣ, ѿ въ законѣ ѿгнѣ поучителѣ дѣнь ѿ нощь. ѿ вѣдетъ ѿакъ дрѣво насажденое при ѿходниихъ кода, ѿже плодъ скѡй дѣстъ во время скѡе, ѿ листъ ѿгнѣ не ѡпадѣтъ: ѿ вѣа, ѿлика ѿце творитъ, ѡупѣетъ. Не тѣакъ нечестѣвѣн, не тѣакъ: но ѿакъ прѣхъ, ѿгѡже возметѣетъ вѣтръ ѡ лица земли. Се гд҃ь радн не воскрѣснѣтъ нечестѣвѣн на сѣдъ, ниже грѣшницы въ совѣтъ прѣвыхъ. ѿакъ вѣстъ гд҃ь пѣтъ прѣвыхъ, ѿ пѣтъ нечестѣвыхъ погнѣнетъ.

## 9.2 Kievan Church Slavonic

Kievan Church Slavonic uses a number of variant glyph forms, such as U+1C81 Long-Legged De (Ѡ) and U+A641 Variant Ze (Ѷ):

Бл҃женѣхъ мѡѡѣ, ѥже не ѥде на совѣтѣхъ нечестѣвыхъ, ѥ на пѡтѣи грѣшныхъ не стѣ, ѥ на сѣдѣлании гѡвѣителей не сѣде: но въз законѣ гдѣи вола ѣгѡ, ѥ въз законѣ ѣгѡ пѡчнѣла день ѥ нѡщѣ. ѥ вѣдетѣ ѥакѡ древо насажденое при исхѡдннхъ воѡхъ, ѣже плодѣ свой дѣетѣ во время своеѣ, ѥ листѣ ѣгѡ не ѡпадѣтъ: ѥ всѣа, ѣлика ѥще тѡорнѣтъ, ѡсѡѣетѣ. Не тѣакѡ нечестѣвѣи, не тѣакѡ: но ѥакѡ прѣхъ, ѣгѡже возмечѣетѣ вѣтрѣ ѡ лица земли. Се гѡ радѣи не воскреснѣтъ нечестѣвѣи на сѣдѣ, ниже грѣшницы въз совѣтѣхъ прѣвнхъ. ѥакѡ вѣстѣ гдѣ пѡтѣ прѣвнхъ, ѥ пѡтѣ нечестѣвыхъ погнѣнетѣ.

### 9.3 Other Languages

The Ponomar Unicode font may also be used to typeset liturgical texts in other languages that use the ecclesiastic Cyrillic alphabet. Three such examples are



## 9.4 Font Features

Ponomar Unicode places some characters in the Private Use Area (PUA). For the general PUA mappings, please see the [PUA Allocation Policy](#).

In addition to the general PUA mappings, some characters have been allocated the open range section of the PUA. These are:

- U+F400 – Alternatives of SMP glyphs: This section contains copies in the BMP of SMP glyphs for support in legacy applications. Currently, the following are available: U+F400 - U+F405 – Typicon symbols (copies of U+1F540 through U+1F545)
- U+F410 – Presentation forms: Contains various presentation forms and ligatures used internally by the font. Generally, these are not intended to be called by users or external applications.
- U+F420 – Linguistic alternatives: Contains alternative shapes of glyphs that are language specific. Right now, these are modern punctuation shapes for use with Latin characters. These are not intended to be called externally.
- U+F441 and on – stylistic alternatives of Latin characters (Blackletter forms). These can be called via the Stylistic Set 2, but, if necessary, they may be called from the PUA directly. They are mapped to in the same order as in the Basic Latin block, beginning with U+F441 (for U+0041 Latin Capital Letter A). In addition to the Basic Latin repertoire, we also have: U+F4DE – Blackletter Thorn; U+F4FE – Lowercase Blackletter Thorn; and U+F575 – Blackletter Long S

The font provides a number of ligatures, which are made by inserting the Zero Width Joiner (U+200D) between two characters. A list of ligatures is provided in Table 1.

Table 1: Ligatures available in Ponomar Unicode

Name	Sequence	Appearance
Ligature A-U	U+0430 U+200D U+0443	Ŧ
Ligature El-U	U+043B U+200D U+0443	Ŧ
Ligature Te-Ve	U+0442 U+200D U+0432	Ŧ

In OpenType, a number of Stylistic Alternatives are defined. They are listed in Table 2. In addition to additional decorative glyphs for the Symbol for Mark’s Chapter, the feature provides the alternate forms of the letter U+0423 U that look exactly like U+A64A Uk (this usage is found in some publications), and an alternative form for the U+0404 Wide Ye for use in contexts where it needs to be distinguished from U+0415 Ye (mostly in Ukrainian text stylized in a Church Slavonic font).

Table 2: Stylistic Alternatives in Ponomar Unicode

	Base Form	Alternate Forms
U+1F545	Ɑ	Ɑ Ɑ Ɑ Ɑ
U+0423	У	Ɑ Ɑ Ɑ Ɑ
U+040E	Ѹ	Ѹ
U+0404	Ѧ	Ѧ

For the Cyrillic letters, the stylistic alternatives feature also allows access to truncated letter forms; the order of the alternate forms is always: lower truncation, upper truncation, left truncation, right truncation. Table 3 demonstrates which truncated forms are available. Generally speaking, truncation should be handled automatically by desktop publishing software and T<sub>E</sub>X, though this is difficult to accomplish.

Table 3: Truncated Forms Accessible via Stylistic Alternatives Feature in Ponomar Unicode

	Base Form	Truncated Forms
U+0440	Ѱ	Ѱ
U+0443	ѳ	ѳ
U+0444	Ѵ	Ѵ Ѵ
U+0445	Ѷ	Ѷ Ѷ Ѷ
U+0446	ѷ	ѷ
U+0449	ѹ	ѹ
U+0471	ѻ	ѻ ѻ
U+A641	Ѹ	Ѹ
U+A64B	Ѹ	Ѹ Ѹ Ѹ

There is also defined Stylistic Set 2 (“ss02”), Blackletter forms. When this stylistic



set is turned on, Latin letters appear in blackletter as opposed to their modern forms. This is useful for setting Latin text side-by-side with Slavonic in some contexts. See the following example:

Regular      The quick brown fox. 1234567890. ѿ ѿѿ по словѣнскѣ.  
 Blackletter   **The quick brown fox. 1234567890. ѿ ѿѿ по словѣнскѣ.**

Note that as of version 2.0 of the font, the ASCII digits (commonly called “Arabic numerals”) are provided in roman form. Use Stylistic Set 2 to access the blackletter forms, if necessary.

## 9.5 SIL Graphite Features

The SIL Graphite features in the font provide the same functionality as the OpenType features. This is mostly of interest to users of LibreOffice, which lacks support for stylistic alternatives and stylistic sets but can access Graphite features (see above).

The “Symbol for Mark’s Chapter” (“mark”) feature provides alternatives for the U+1F545 Symbol for Marks Chapter, much like the salt feature in OpenType. The following values produce the results given in Table 4.

Table 4: Values of the Symbol for Mark’s Chapter (“mark”) Feature in Ponomar Unicode

	Base form	Alternative 1	Alternative 2	Alternative 3
U+1F545				
	Alternative 4	Alternative 5	Alternative 6	Alternative 7
	Alternative 8			

The following additional Graphite features are provided (with the exception of the “hyph” feature, they duplicate the functionality of OpenType features):

- The “Truncation” feature (“trnc”) provides the same functionality as stylistic alternatives (for truncation) above. The possible values are: 1 (lower truncation), 2 (upper truncation), 3 (left truncation) and 4 (right truncation).

- The “Use blackletter characters for Latin” feature (“blck”) provides the same functionality as Stylistic Set 2 in OpenType (see above). Possible values are 0 (no) and 1 (yes).
- The “Use alternative form of U” feature (“altu”) provides the alternative forms of the letter U+0423 U that look exactly like U+A64A Uk. Possible values are 0 (no) and 1 (yes).
- The “Cyrillic i has dot” feature (“doti”) provides a localized form of U+0456 Cyrillic I for use in Ukrainian text. Possible values are 0 (no) and 1 (yes).
- The “Use underscore for hyphenation” feature (“hyph”) replaces all instances of U+002D Hyphen-Minus and U+2010 Hyphen with U+005F Low Line (underscore) for use as a hyphenation character. This is meant as a workaround to [LibreOffice Bug 85731](#), which does not allow you to specify the hyphenation character in LibreOffice. Possible values are 0 (no) and 1 (yes). Please note that this feature will be deprecated once the necessary functionality is added to LibreOffice.

## 10 Fedorovsk Unicode

Fedorovsk Unicode is based on the Fedorovsk font designed by Nikita Simmons. It has been re-encoded for Unicode, with added OpenType and Graphite features by Aleksandr Andreev. The Fedorovsk typeface is supposed to reproduce the typeface of the printed editions of Ivan Fedorov produced in Moscow, for example, the Apostol of 1564. The font is intended primarily for typesetting pre-Nikonian (Old Rite) liturgical texts or for working with such texts in an academic context.

### 10.1 Sample Texts

#### 10.1.1 Apostol of Ivan Fedorov

**П**ѣрвое оубо слово сотвори́хъ о́всѣхъ , ѿ , дѣо́филе , о́ниже  
 на́уча́тъ ꙗ́ко , тѣвори́тиже ѿ о́учи́ти . до негѣже дѣе , заповѣда́въ  
 а́пѣломъ дѣхѣомъ стѣи , ѿхъже ѿзѣра́ възнесе́ся . прѣни́мниже ѿ поста́ви себе́  
 жѣва по страда́ниѣи своѣмъ . во мно́зехъ ѿстѣнны́хъ зна́менихъ .  
 дѣньми чѣтыри́дѣсѣтьми ѡбѣла́нѣся ѿмъ ѿгѣла́ ѿже о́црѣви́и бѣжи́и .  
 снѣи́мниже ѿѣады́ , повѣле́бѣа́ше ѿмъ ѿѣро́сала́нѣма не ѿла́дѣа́ти сѣа . нѣ  
 жѣа́дѣа́ти о́бѣтѣова́нїе ѿѣе , ѿже слы́шасѣте ѿ́мене . ѿа́кѡ ѿѡ́аннѣ

о́убо крѣ́пѣ́хъ ѣ́сть водо́ю . вы́же ѣ́мате крестѣ́хъ дѣ́ломъ сѣ́бѣ́мъ ,  
непо́мнозѣ́хъ сѣ́хъ дѣ́хъ .

### 10.1.2 Flowery Triodion

сѣ́ры пѣ́щѣ́ . глѣ́хъ , ѣ́ . Да воскрѣ́нетъ бѣ́хъ , \* ѣ́ разы́дѣ́хъ вразѣ́  
ѣ́гѣ́хъ . Пѣ́хъ сѣ́еннаѣ́ на́мъ днѣ́хъ показѣ́хъ , пѣ́хъ но́ва сѣ́ла ,  
пѣ́хъ тѣ́нствѣ́ннаѣ́ , пѣ́хъ всечѣ́стнаѣ́ , пѣ́хъ хрѣ́ста ѣ́зѣ́вѣ́телеѣ́ ,  
пѣ́хъ непоро́чнаѣ́ , пѣ́хъ вѣ́ликаѣ́ , пѣ́хъ вѣ́рныхъ , пѣ́хъ двѣ́ри  
ра́нскѣ́ на́мъ ѡ́верза́ющѣ́хъ , пѣ́хъ вѣ́хъ ѡ́сѣ́ляющѣ́хъ вѣ́рныхъ .

## 10.2 OpenType Features

The font provides a number of ligatures, which are made by inserting the Zero Width Joiner (U+200D) between two characters. A list of ligatures is provided in Table 5.

Table 5: Ligatures available in Fedorovsk Unicode

Name	Sequence	Appearance
Ligature A-U	U+0430 U+200D U+0443	Ѡ
Ligature El-U	U+043B U+200D U+0443	ѡ
Ligature A-Izhitsa	U+0430 U+200D U+0475	Ѣ
Ligature El-Izhitsa	U+043B U+200D U+0475	ѣ
Ligature Te-Ve	U+0442 U+200D U+0432	Ѥ
Ligature Er-Yat	U+0440 U+200D U+0463	Ѧ

In OpenType, a number of Stylistic Alternatives are defined. They are listed in Table 6. In addition to providing alternative glyph shapes for U+1F545 Symbol for Mark’s Chapter, they allow you to control the positioning of diacritical marks over certain letters.

Additionally, three stylistic sets have been defined in the font. Stylistic set 1 (“Right-side accents”) positions the accents over the Yat and the Uk on the right side and Stylistic set 2 (“Left-side accents”) positions the accents over the Yat and the Uk on the left side. These stylistic sets are useful when a text uses one of these positionings throughout. Stylistic set 10 (“Equal Baseline Variants”) sets the capital letters on the same baseline as the lowercase letters (useful for working with the font in an academic context where the traditionally lowered baseline of

Table 6: Stylistic Alternatives in Fedorovsk Unicode

	Base Form	Alternate Forms					
U+0404	€	€					
U+0426	ll	ll					
U+0491	r	r					
U+A64C	∞	∞					
U+047C	∞	∞					
U+047E	∞	∞					
U+047F	∞	∞					
U+1F545	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ
U+0463 U+0486	ℳ	ℳ					
U+0463 U+0300	ℳ	ℳ	ℳ				
U+0463 U+0301	ℳ	ℳ	ℳ				
U+0463 U+0311	ℳ	ℳ	ℳ				
U+0463 U+0486 U+0301	ℳ	ℳ					
U+A64B U+0486	ℳ	ℳ					
U+A64B U+0300	ℳ	ℳ	ℳ				
U+A64B U+0301	ℳ	ℳ	ℳ				
U+A64B U+0311	ℳ	ℳ	ℳ	ℳ			
U+A64B U+0486 U+0301	ℳ	ℳ					

uppercase letters can cause vertical spacing issues when working with text that is both in Latin and Cyrillic scripts). Here is an example:

Χρῖστος βοικρεῖ ἡς μέγυλχῃ (regular text)  
 Χρῖστος βοικρεῖ ἡς μέγυλχῃ (Stylistic Set 10 enabled)

### 10.3 Graphite Features

The stylistic alternatives of the Mark’s Chapter symbol, the Letter Ge with Up-turn, and the letters Ye, Tse, and Omega have been duplicated as Graphite features in the TTF version of the font, with names “Symbol for Mark’s Chapter”, “Ye”, “Tse”, “Ghe”, and “Omega” respectively. For the alternatives of the Mark’s Chapter symbol, the values of the property are assigned to correspond with the [Private Use Area Allocation Policy](#) and other fonts. The Graphite features are demonstrated in Table 7.

Table 7: Alternatives via Graphite Features in Fedorovsk Unicode

	Base form	Alternative 1	Alternative 2	Alternative 3
U+0404	Є	Є		
U+0426	И	И		
U+0491	г	г		
U+A64C	Ѧ	Ѧ		
U+047C	Ѧ	Ѧ		
U+047E	Ѧ	Ѧ		
U+047F	Ѧ	Ѧ		
U+1F545	Ѧ	Ѧ	Ѧ	Ѧ
		Alternative 4	Alternative 7	Alternative 8
		Ѧ	Ѧ	Ѧ
		Alternative 9		
		Ѧ		

Two additional Graphite features are defined: “Accent Positions”, with values “left” and “right”, which mimics the behavior of stylistic sets 1 and 2; and “Equal Baseline” (with value “yes”), which mimics the behavior of Stylistic Set 10.

## 11 Menaion Unicode

The Menaion typeface is supposed to be used for working with text of Ustav-era manuscripts. It contains the full repertoire of necessary Cyrillic and Glagolitic glyphs as well as glyphs of Byzantine Ecphonetic notation of the kind used in Cyrillic or Glagolitic manuscripts.

The Menaion font was originally designed by Victor A. Baranov at [the Manuscript Project](#). It was re-encoded for Unicode by Aleksandr Andreev with permission of the original author.

### 11.1 Sample Texts

Samples of text in Menaion Unicode are presented in Figures 1 and 2. Please note that combining Glagolitic letters (Glagolitic Supplement) will become available in Unicode 9.0. In Microsoft products, correct glyph positioning for these characters using OpenType features is currently not possible. To achieve the desired output, you will need to use LibreOffice, X<sub>Y</sub>TeX, LuaTeX, or advanced desktop publishing software such as Adobe InDesign.

### 11.2 Provided Ligatures

The font provides a number of ligatures, which are made by inserting the Zero Width Joiner (U+200D) between two characters. The list of ligatures is provided in Table 8. The ligatures may be processed using either OpenType or SIL Graphite.

## 12 Pomorsky Unicode

The Pomorsky Unicode font is a close (idealized) reproduction of the decorative calligraphic style of book and chapter titles, which was most likely developed in the 1700's by the scribes of the Old Ritualist Vyg River Hermitage (Выговская пустынь). It is seen extensively in the chant manuscripts, liturgical manuscripts, hagiographic and polemical works of the Pomortsy and Fedoseyevtsy communities, and is a traditional and “organic” style of lettering lacking any obvious influence from western European and Latin typography. The Pomorsky typeface was originally designed by Nikita Simmons. It was edited and re-encoded for Unicode by Aleksandr Andreev. It is intended for use with *bukvitsi* (drop caps) and decorative titling.

Figure 1: Cyrillic text from the Ostromir Gospels (11th century)

1           НѢКОНН БѢ СЛОВО  
 2           Н СЛОВО БѢ ОТЪ  
 3           БѢ Н БѢ БѢ  
 4           СЛОВО ꙗко бѢ  
 5           НѢКОНН ОУ  
 6   БѢ ꙗко н тѣмъ вса бѣи-  
 7   ша ꙗко н бѣзъ него нн-  
 8   ѣтоже нѣ бѣистъ .  
 9   нѣ бѣистъ ꙗко въ то-  
 10   мъ животъ бѣ . н  
 1   животъ бѣ свѣтъ  
 2   ѣлобѣкомъ ꙗко н свѣ-  
 3   тъ въ тѣмъ свѣти-  
 4   тѣса . н тѣма нго  
 5   нѣ обѣтъ ꙗко бѣистъ  
 6   ѣлѣкъ послѣланъ  
 7   отъ бѣ . нма нмоу  
 8   ноанъ ꙗко тѣ приде  
 9   въ свѣдѣтель-  
 10   ство . да свѣдѣте-  
 2.2 1   льствоуѣтъ о свѣ-  
 2   тѣ ꙗко да всн вѣрж  
 3   нмѣтъ нмѣ ꙗко нѣ бѣ  
 4   тъ свѣтъ ꙗко нѣ да  
 5   свѣдѣтельствоу-  
 6   ѣтъ о свѣтѣ ꙗко бѣ  
 7   свѣтъ истинный  
 8   н . нѣ просвѣщаѣ-  
 9   тъ всакого ѣлка ,  
 10   граждѣща въ миръ ꙗко

Figure 2: Glagolitic text from Codex Assemanius (11th century)

1    ✠ Ǫꙋⱁ ⱁⱏ ⱄⱏ  
 2        ⱄⱏⱏⱏⱏⱏ ⱏⱏ  
 3        ⱏⱏⱏⱏⱏ .  
 4        ⱄ ⱏⱏⱏⱏⱏ  
 5        ⱏⱏ ⱏⱏ ⱏⱏ .  
 6        ⱄ ⱏⱏ ⱏⱏ  
 7        ⱏⱏⱏⱏⱏ .  
 8        ⱏⱏ ⱏⱏ ⱄⱏⱏ-  
 9    ⱏⱄ . ⱏⱏ ⱏⱏ . ⱏⱏ-  
 10    ⱏⱏ ⱏⱏⱏⱏ ⱏⱏⱄ-  
 11    ⱏⱏ . ⱄ ⱏⱏⱏ ⱏⱏⱏ  
 12        ⱏⱏⱏⱏⱏⱏⱏ  
 13    ⱏⱏ ⱏⱏⱄⱏⱏⱏ . ⱏ-  
 14        ⱏⱏ ⱏⱏⱏⱏ .  
 15    ⱏⱏ ⱏⱏⱏⱏ ⱏⱏⱏ-  
 16    ⱏⱏ ⱏⱏ . ⱄ ⱏⱏⱏ-  
 17        ⱏⱏ ⱏⱏ ⱏⱏⱏⱏ  
 18    ⱏⱏⱏⱏ . ⱄ ⱏⱏⱏⱏ  
 19    ⱏⱏ ⱏⱏⱏⱏ ⱏⱏⱏ-  
 20    ⱏⱏⱏⱏ ⱏⱏ . ⱄ ⱏⱏ-  
 21    ⱏⱏ ⱏⱏ ⱏⱏ ⱏⱏⱏ



Table 8: Ligatures available in the Menaion Unicode font

Name	Sequence	Appearance
Small Ligature I-Ye	U+0438 U+200D U+0435	<b>ИѢ</b>
Small Ligature En-I	U+043d U+200D U+0438	<b>ИИ</b>
Small Ligature En-Small Yus	U+043d U+200D U+0467	<b>ИѦ</b>
Small Ligature Es-Ve	U+0441 U+200D U+0432	<b>ѢѢ</b>
Small Ligature Te-Er	U+0442 U+200D U+0440	<b>ѢѢ</b>
Capital Litagure A-U	U+0410 U+200D U+0423	<b>ѦѢ</b>
Small Ligature A-U	U+0430 U+200D U+0443	<b>ѢѢ</b>
Small Ligature A-Te	U+0430 U+200D U+0442	<b>ѢѢ</b>
Capital Ligature I-Ye	U+0418 U+200D U+0415	<b>ИѢ</b>
Capital Ligature El-Ge	U+041b U+200D U+0413	<b>ѢѢ</b>
Small Ligature El-Ge	U+043b U+200D U+0433	<b>ѢѢ</b>
Capital Ligature En-I	U+041d U+200D U+0418	<b>ИИ</b>
Capital Ligature En-Small Yus	U+041d U+200D U+0466	<b>ИѦ</b>
Capital Ligature Es-Ve	U+0421 U+200D U+0412	<b>ѢѢ</b>
Small Ligature Te-Yat	U+0442 U+200D U+0463	<b>ѢѢ</b>
Capital Ligature Te-Ve	U+0422 U+200D U+0412	<b>ѢѢ</b>
Small Ligature Te-Ve	U+0442 U+200D U+0432	<b>ѢѢ</b>
Capital Ligature Te-I	U+0422 U+200D U+0418	<b>ѢИ</b>
Small Ligature Te-I	U+0442 U+200D U+0438	<b>ѢИ</b>
Capital Ligature Te-Er	U+0422 U+200D U+0420	<b>ѢѢ</b>
Ligature Capital A-Small Te	U+0410 U+200D U+0442	<b>ѢѢ</b>
Capital Ligature Te-Soft Sign	U+0422 U+200D U+042c	<b>ѢѢ</b>
Small Ligature Te-Soft Sign	U+0442 U+200D U+044c	<b>ѢѢ</b>
Small Ligature Te-A	U+0442 U+200D U+0430	<b>ѢѢ</b>

Several versions of many glyphs are provided in the font. The ornate forms of the letters are default and provided at the uppercase Cyrillic codepoints; they should be used as much as possible. Simpler forms can be used whenever the letters need a less ornate appearance, or when diacritics might conflict with the ornamentation (or when the ornamentation of one character will conflict with the ornamentation of another); these simple forms are available as Stylistic Set 1 or as the Graphite feature “Use simple forms” (smpl). There are a few additional characters that are stylistic variants, which are provided as Stylistic Alternatives (salt) or as the Graphite feature “Alternates” (salt). Since the font is intended for drop caps and titling, lowercase characters are not available.

The base form, the “simple” form, and any stylistic alternatives of a character are demonstrated in Table 9.

Table 9: Character shapes provided by Pomorsky Unicode



### 12.1 Sample Texts

УЎЎЎЎ ЁЁУЁРММ.  
УЎЎЎЎ ЁЁУЁРММ.  
ЅЎЎЎЎЎЎЎ.  
ЅЎЎЎЎЎЎЎ.  
ЅЎЎЎЎЎЎЎ.

## 13 Known Issues

Here are some known issues:

- Kerning is not available for Latin characters in any of the fonts. Since it is not expected for these fonts to be heavily used to typeset Latin text, this issue is not a very high priority for resolution.
- Ponomar Unicode has Graphite-based kerning for Cyrillic characters starting in version 2.0, but it is defective. In particular, inserting a diacritical mark will break the kerning between two letters. This will be fixed in version 2.1.
- Kerning is not available in the Graphite version of Pomorsky Unicode.

There may be other issues, but before reporting such issues, please check that your software properly supports OpenType and / or SIL Graphite. We suggest checking for expected behavior in Xe<sub>La</sub>TeX or Lua<sub>La</sub>TeX.

## 14 Credits

The authors would like to thank the following people:

- Vladislav Dorosh, who allowed his [Hirmos](#) font to be re-encoded in Unicode and modified, leading to the creation of the Ponomar font.
- Viktor Baranov of the [Manuscripts](#) project, who allowed the re-encoding and modification of his Menaion font.
- Michael Ivanovich for help in designing the characters for Sakha (Yakut), partially taken from his Sakha UCS font.
- Alexey Kryukov, who answered various questions about FontForge and whose extensive documentation for the [Old Standard](#) font was consulted and partially reused.
- Mike Kroutikov, who put together the  $\text{\TeX}$  package of the fonts.