Church Slavonic Fonts

Aleksandr Andreev Yuri Shardt Nikita Simmons*

November 4, 2016 version 1.1

(PDF file generated on January 31, 2017)

Contents

1 Introduction **3**

2 License **3**
2.1 The Legal Text . . . 3

3 Introduction **4**

4 Installation and Usage **4**
4.1 Font Formats . . . 4
4.2 Source Packages . . . 6

5 System Requirements **6**
5.1 Microsoft Windows . . . 6
5.2 GNU/Linux . . . 6
5.3 OS X . . . 7

6 Private Use Area **7**

7 OpenType Technology **8**
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.

7.3.2 Glyph Composition and Decomposition 9
7.3.3 Language-based Features 10
7.3.4 Stylistic Alternatives and Stylistic Sets 10

8 SIL Graphite Technology 11
8.1 Graphite in LibreOffice 12
8.2 Graphite in XeTeX 12

9 Ponomar Unicode 13
9.1 Synodal Church Slavonic 13
9.2 Kievan Church Slavonic 13
9.3 Other Languages 14
9.4 Font Features 15
9.5 SIL Graphite Features 18

10 Fedorovsk Unicode 19
10.1 Sample Texts 19
10.1.1 Apostol of Ivan Fedorov 19
10.1.2 Flowery Triodion 20
10.2 OpenType Features 20
10.3 Graphite Features 22

11 Menaion Unicode 22
11.1 Sample Texts 23
11.2 Provided Ligatures 23

12 Pomorsky Unicode 23
12.1 Sample Texts 28

13 Monomakh Unicode 28
13.1 Sample Bilingual Text 28
13.2 OpenType and SIL Graphite features 29

14 Indiction Unicode 30
14.1 Sample Texts 31

15 Known Issues 31

16 Credits 33

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 CreativeCommons 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 TEX, you will need a Unicode-aware TEX engine, such as XETEX or 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 TEX, you will need a Unicode-aware TEX engine, such as XETEX or 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 XeTeX 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

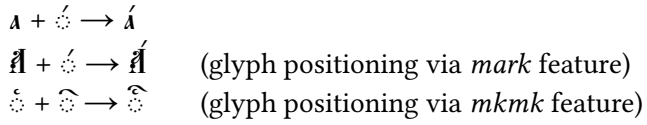
OpenType support is provided by the HarfBuzz shaping library, which is accessible through FreeType, part of most standard distributions of the X Window System. Thus, OpenType will be available in any application that uses FreeType, though many applications lack an interface to turn on and off optional features. Generally speaking, you will get best results in XeTeX 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

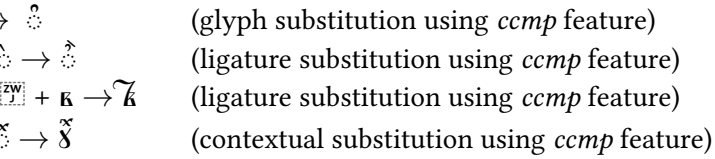
OpenType allows smart diacritic positioning: if you type a letter followed by a diacritic, the diacritic will be placed exactly above or below the letter; this is provided by the `mark` feature. In addition, the `mkmk` feature is used to position two marks with respect to each other, so that an additional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



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 XeTeX).

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:



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 `locl` (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:

ı (Church Slavonic text)
i (Ukrainian text)

To make use of these features, you need an OpenType-aware application that supports specifying the language of text, for example XeTeX or LuaTeX 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.[1]

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:

[1]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.

U+1F545	Alternative Glyphs			

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

Graphite support is available in XeTeX, 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 XeTeX 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 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

Бл҃же́нъ мꙋ́жъ, и҆́же не и҆́де на совѣ́тъ нечести́выхъ, и҆ на пꙋтѝ грѣ́шныхъ не ста̀, и҆ на сѣда́лищи гꙋби́телей не сѣ́де: но въ зако́нѣ гдⷭ҇ни во́лѧ є҆гѡ̀, и҆ въ зако́нѣ є҆гѡ̀ поꙋчи́тсѧ де́нь и҆ но́щь. И҆ бꙋ́детъ ꙗ҆́кѡ дре́во насажде́ное при и҆схо́дищихъ во́дъ, є҆́же пло́дъ сво́й да́стъ во вре́мѧ своѐ, и҆ ли́стъ є҆гѡ̀ не ѿпаде́тъ: и҆ всѧ̑, є҆ли̑ка а҆́ще твори́тъ, ᲂу҆спѣ́етъ. Не та́кѡ нечести́вїи, не та́кѡ: но ꙗ҆́кѡ пра́хъ, є҆го́же возмета́етъ вѣ́тръ ѿ лица̀ землѝ. Сегѡ̀ ра́ди не воскре́снꙋтъ нечести́вїи на сꙋ́дъ, нижѐ грѣ̑шницы въ совѣ́тъ прⷣвныхъ. Ꙗ҆́кѡ вѣ́сть гдⷭ҇ь пꙋ́ть прⷣвныхъ, и҆ пꙋ́ть нечести́выхъ поги́бнетъ.

9.2 Kievan Church Slavonic

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

Бл҃же́нъ мꙋ́жъ, и҆́же не и҆́де на совѣ́тъ нечести́выхъ, и҆ на пꙋтѝ грѣ́шныхъ не ста̀, и҆ на сѣда́лищи гꙋби́телей не сѣ́де: но въ зако́нѣ гдⷭ҇ни во́лѧ є҆гѡ̀, и҆ въ зако́нѣ є҆гѡ̀ поꙋчи́тсѧ де́нь и҆ но́щь. И҆ бꙋ́детъ ꙗ҆́кѡ дре́во насажде́ное при и҆схо́дищихъ во́дъ, є҆́же плодъ сво́й да́стъ во вре́мѧ своѐ, и҆ ли́стъ є҆гѡ̀ не ѿпаде́тъ: и҆ всѧ̑, є҆ли̑ка а҆́ще твори́тъ, ᲂу҆спѣ́етъ. Не та́кѡ нечести́вїи, не та́кѡ: но ꙗ҆́кѡ пра́хъ, є҆го́же возмета́етъ вѣ́тръ ѿ лица̀ землѝ. Сегѡ̀ ра́ди не воскре́снꙋтъ нечести́вїи на сꙋ́дъ, нижѐ грѣ̑шницы въ совѣ́тъ прв҃ныхъ. Ꙗ҆́кѡ вѣ́сть гдⷭ҇ь пꙋ́ть прв҃ныхъ, и҆ пꙋ́ть нечести́выхъ поги́бнетъ.

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 fully supported by the font: Romanian (Moldovan) in its Cyrillic alphabet, Aleut (Fox Island or Eastern dialect) in its Cyrillic alphabet, and Yakut (Sakha) as written in the alphabet created by Bishop Dionysius (Khitrov).
Here is an example of the Lord's Prayer in Romanian (Moldovan) Cyrillic:

Та́тъль но́стрꙋ ка́реле є҆́щй ѧ҆ Че́рюрй: сфн҃цѣ́скъсе Нꙋ́меле тъ́ꙋ: ві́е ѧ҆пъръці́ѧ та̀: фі́е во́ѧ та̀, прекꙋ́мь ѧ҆ Че́рю̆ шѝ пре пъмѫ́нть. Пѫ́йнѣ но́астръ чѣ̀ депꙋ́рꙋрѣ дъ́не но́аѡ а҆́стъзй. Шѝ не ꙗ҆́ртъ но́аѡ греша́леле но́астре, прекꙋ́мь шѝ но́й є҆ртъ́мь греши́цилѡрь но́щри. Ши́ нꙋ́не дꙋ́че пре но́й ѧ҆ и҆спи́тъ. Чѝ не и҆збъвѣ́ще де че́ль ръ́ꙋ.

And here is an example of the Lord's Prayer in Aleut Cyrillic:

Тꙋмани́нъ А҆́дакъ! А҆́манъ акꙋ́х҆тхинъ и́нинъ кꙋ́ѓинъ, А҆са́нъ амчꙋг҆а́сѧ́да́г҆та, А҆ѓали́нъ а҆ка́г҆та, А҆нꙋхтана́тхинъ малга́г҆танъ и́нимъ кꙋ́ганъ ка́юхъ та́намъ кꙋ́ганъ. Калга́дамъ анꙋхтана̀ ѓи҆̂нъ акача́ ꙋ҆а҆ѧ́мъ: ка́юхъ тꙋма́нинъ а́д҆ꙋнъ ѓи҆̂нъ игни́да, а҆ма́кꙋнъ тꙋ́манъ ка́юхъ малгалиги́нъ ѓи҆̂нъ ад҆ꙋг҆и́нанъ игнида́кꙋнъ: ка́юхъ тꙋ́манъ сꙋглатачх҆и́г҆анах҆тхинъ, та́г҆а ад҆алю́дамъ илѧ́нъ тꙋ́манъ аг҆г҆ича.

And here is an example of the Lord's Prayer in Yakut (Sakha):

Халланнаръ юрдюлѧригѧрь баръ агабытъ бисенѧ! Свѧтейдѧннинь а҃тыѓъ ѯенѧ; кѧллинь царствaѓъ ѯенѧ; сирь юрдюгѧрь кёѓюлюѓь ѯенѧ, халланъ юрдюгѧрь курдукъ боллунъ; бюгюѓю кюннѧги асыръ аспытынъ бисенинь кулу бисеха бюгюнь; бисиги да естѧрбитинь халларъ бисеха, хайтахъ бисиги да халларабытъ беѯбить естѧхтѧрбитигѧрь; килѧримѧ да бисигини альджархайга; хата быса бисигини албынтанъ.

Here is an example using the Typicon symbols from Nikita Syrnikov's book Клю́чъ къ церко́вномꙋ ᲂу҆ста́вꙋ:

і҆ ѿ іѡа́ннꙋ ї҃ на лⷮ бл҃же҃, па́влꙋ пⷭ҇ г҃. а і҆оа́ннꙋ ѕ҃.
и҆і і҆ ѿ кири́лꙋ ї҃ ко҃ и҆ и҆́ко҃ ѻ҆́бщїй и҆ ко҃ а҆фана́сїю.
д҃ ꙋ҆ ш҃ и҆си́дорꙋ ᲂ геѻ́ргїю кири́лѣ ;

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	
Ligature El-U	U+043B U+200D U+0443	
Ligature Te-Ve	U+0442 U+200D U+0432	

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					
U+0423					
U+040E					
U+0404					

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 TEX, though this is difficult to accomplish.

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

	Base Form	Truncated Forms		
U+0440				
U+0443				
U+0444				
U+0445				
U+0446				
U+0449				
U+0471				
U+A641				
U+A64B				

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. а҆ їѣ по словéнски.
Blackletter	The quick brown fox. 1234567890. а҆ їѣ по словéнски.

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

Пе́рвое оу҆́бо сло́во сотвори́хъ ѻ҆ всѣ́хъ , ѿ , ѳеѻ҆́филе , ѻ҆ ни́хже начѧ́тъ і҃с , твори́тиже и҆ оу҆чи́ти . до него́же д҃не , запо-вѣ́давъ а҆п҃ломъ д҃хомъ с҃ты́мъ , и҆́хже и҆збра̀ вознесе́сѧ . пре́дни́миже и҆ поста́ви себѐ жи́ва по страда́нїи свое́мъ . во мно́зехъ и҆́стинныхъ зна́менїи҆хъ . д҃ньми четы́ридесѧтьми ꙗ҆влѧ́ꙗсѧ и҆́мъ и҆ гл҃ѧ ꙗ҆́же ѻ҆ цр҃твїи б҃жїи . сни́миже и҆ ꙗ҆ды̀ , повелѣва́ше и҆́мъ ѿ і҆ерусали́ма не ѿлꙋча́тисѧ . но̑ жда́ти ѻ҆бѣтова́нїе ѿ ч҃ее , е҆́же слы́шасте ѿ менѐ . ꙗ҆́кѡ і҆ѡ҃а́ннъ оу҆́бо кр҃ти́лъ е҆́сть водо́ю . вы́же и҆́мате крести́тисѧ д҃хомъ с҃ты́мъ , не по мно́зѣхъ си́хъ д҃не .

10.1.2 Flowery Triodion

ст҃ры па́сцѣ . гла́съ , є҃ . Да воскр҃нетъ бг҃ъ ,* и҆ разы́дꙋтсѧ врази̑ є҆гѡ̀ . Па́сха сщ҃е́ннаѧ на́мъ дне́сь показа́сѧ , па́сха но́ва ст҃а́ѧ , па́сха таи́нственнаѧ , па́сха всечестна́ѧ , па́сха хр҃та̀ и҆зба́вителѧ , па́сха непоро́чнаѧ , па́сха вели́каѧ , па́сха вѣ́рнымъ , па́сха двѣ́ри ра́йскїѧ на́мъ ѿверза́ющаѧ , па́сха всѣ́хъ ѡ҆сщ҃а́ющаѧ 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	ꙋ
Ligature El-U	U+043B U+200D U+0443	ꙋ
Ligature A-Izhitsa	U+0430 U+200D U+0475	ѵ
Ligature El-Izhitsa	U+043B U+200D U+075	ѵ
Ligature Te-Ve	U+0442 U+200D U+0432	ѣ
Ligature Er-Yat	U+0440 U+200D U+0463	ѣ

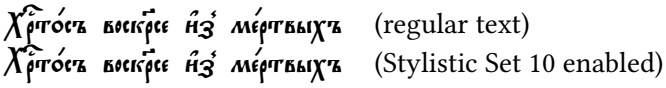
In OpenType, a number of Stylistic Alternatives are defined. They are listed in Table 6. In addition to providing alternative glyph shapes for U+1F545 Symbol for Mark's Chapter, they allow you to control the positioning of diacritical marks over certain letters.

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

Table 6: Stylistic Alternatives in Fedorovsk Unicode

	Base Form	Alternate Forms						
U+0404	Є	Є						
U+0426	Ц	Ц						
U+0491	ґ	ґ						
U+A64C	Ѡ	Ѡ						
U+047C	Ѽ	Ѽ						
U+047E	Ѿ	Ѿ						
U+047F	Ѿ	Ѿ						
U+1F545	Ѿ	Ѿ	Ѿ	Ѿ	Ѿ	Ѿ	Ѿ	Ѿ
U+0463 U+0486	ѣ҆	ѣ҆						
U+0463 U+0300	ѣ̀	ѣ̀	ѣ̀					
U+0463 U+0301	ѣ́	ѣ́	ѣ́					
U+0463 U+0311	ѣ̑	ѣ̑	ѣ̑					
U+0463 U+0486 U+0301	ѣ҆́	ѣ҆́						
U+A64B U+0486	ꙋ҆	ꙋ҆						
U+A64B U+0300	ꙋ̀	ꙋ̀	ꙋ̀					
U+A64B U+0301	ꙋ́	ꙋ́	ꙋ́					
U+A64B U+0311	ꙋ̑	ꙋ̑	ꙋ̑	ꙋ̑				
U+A64B U+0486 U+0301	ꙋ҆́	ꙋ҆́						

Хр҃то́съ воскр҃се и҆з̾ ме́ртвыхъ (regular text)
Хр҃то́съ воскр҃се и҆з̾ ме́ртвыхъ (Stylistic Set 10 enabled)



10.3 Graphite Features

The stylistic alternatives of the Mark's Chapter symbol, the Letter Ge with Upturn, 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	Ц	Ц		
U+0491	ґ	Ґ		
U+A64C	ꙍ	Ꙍ		
U+047C	Ѽ	Ѽ		
U+047E	Ѿ	Ѿ		
U+047F	ѿ	Ѿ		
U+1F545	Ѿ҃	Ѿ҃	Ѿ҃к	Ѿ҃
		Alternative 4	Alternative 7	Alternative 8
		Ѿ҃	Ѿ҃	Ѿ҃
		Alternative 9		
		Ѿ҃		

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, XꚖTEX, 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.

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

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

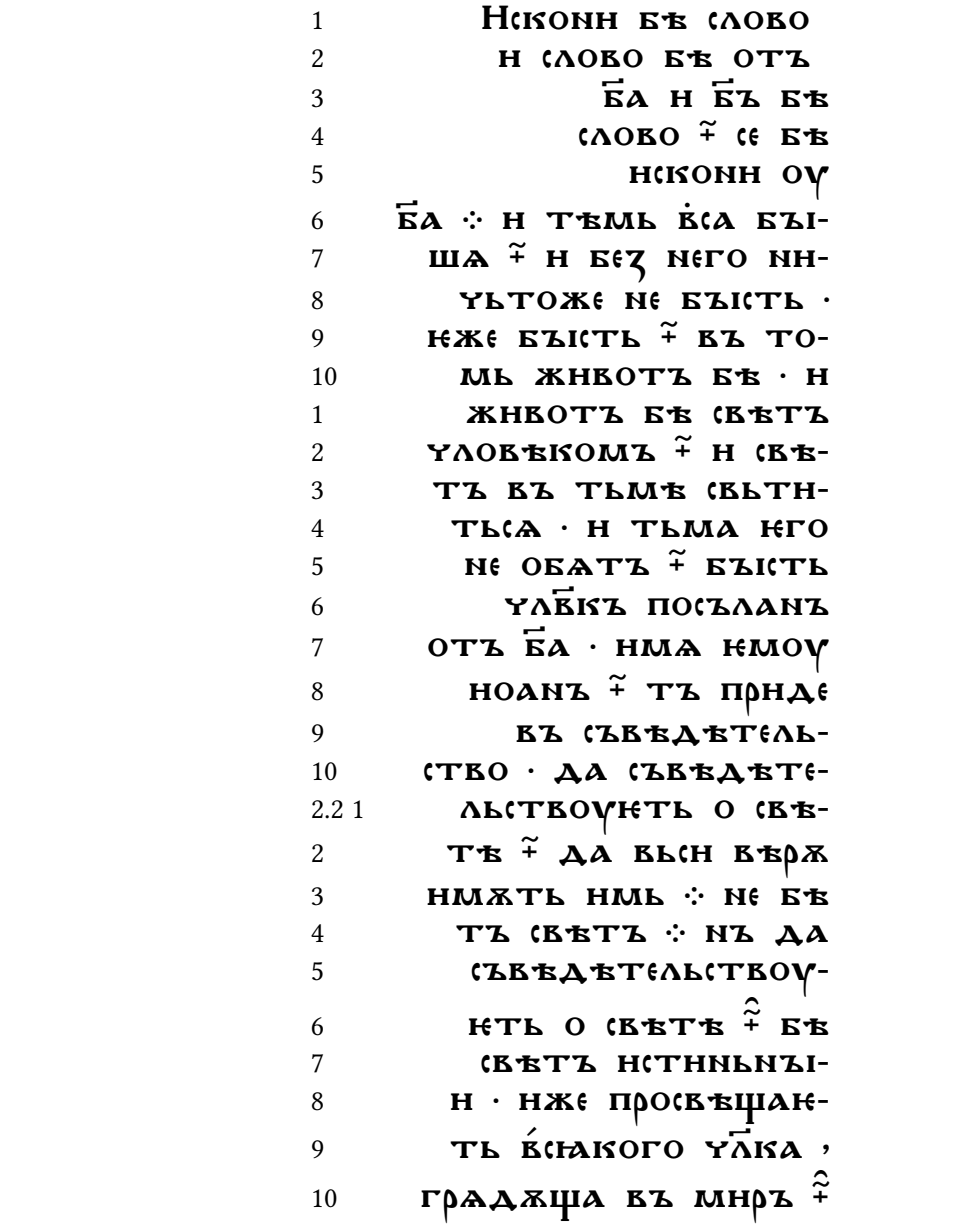


Figure 2: Glagolitic text from Codex Assemanius (11th century)

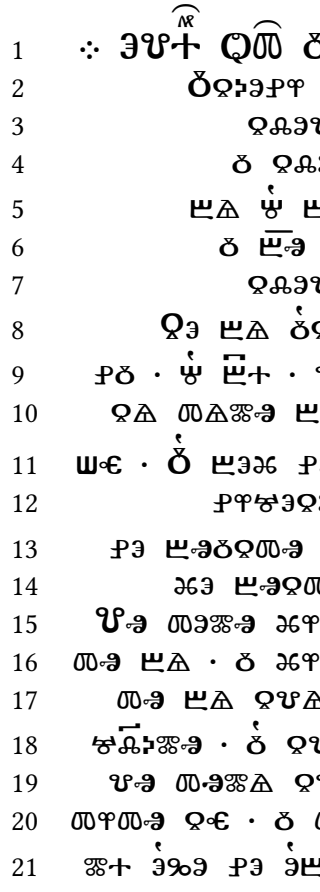


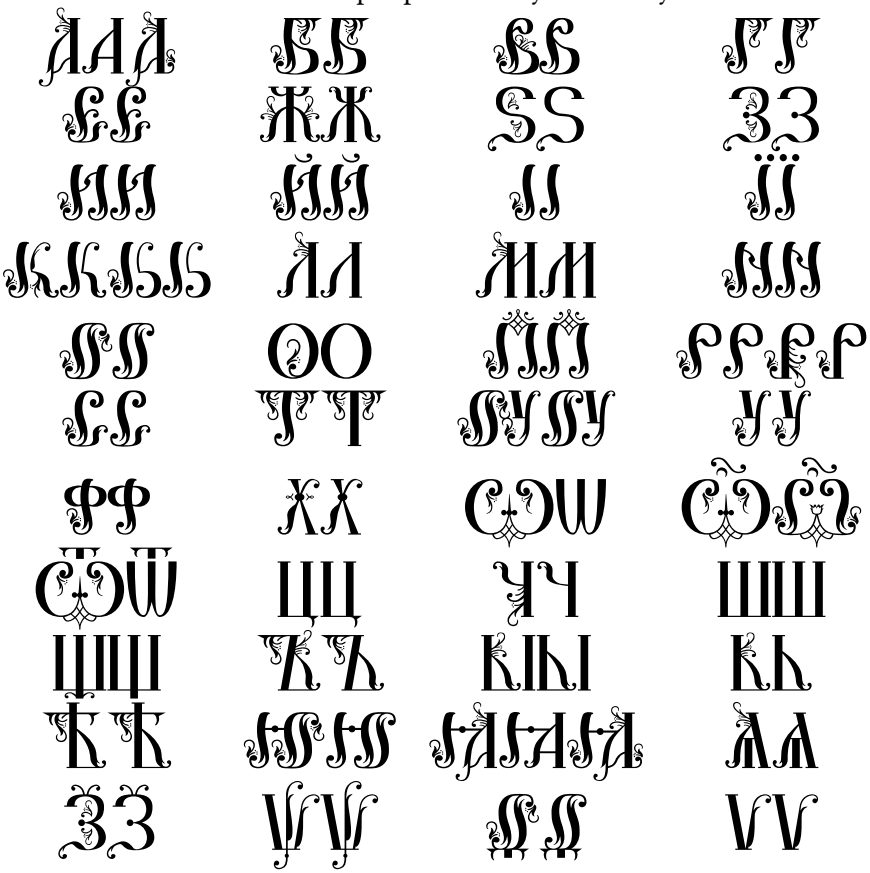
Table 8: Ligatures available in the Menaion Unicode font

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

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

ЧИ́НЪ ВЕЧЕ́РНИ.
ЧИ́НЪ ВЕЧЕ́РНИ.
СѴ̈НСѺДИКЪ.
СѴ̈НѠ́ДИКЪ.

13 Monomakh Unicode

Monomakh Unicode is based on the Monomachus font designed by Alexey Kryukov. It has been modified with permission. Monomakh Unicode is a Cyrillic font implemented in a mixed ustav/poluustav style and intended to cover needs of researches dealing with Slavic history and philology. It includes all historical Cyrillic characters currently defined in Unicode font also includes a set of Latin letters designed to be stylistically compatible with the Cyrillic part. This may be useful for typesetting bilingual editions in Church Slavonic and languages written in the Latin script, especially those that use many diacritical marks, as in the Romanian example below.

13.1 Sample Bilingual Text

Бл҃же́нъ мꙋ́жъ, и҆́же не и҆́де на совѣ́тъ нечести́выхъ, и҆ на пꙋтѝ грѣ́шныхъ не ста̀, и҆ на сѣда́лищи гꙋби́телей не сѣ́де: но въ зако́нѣ гдⷭ҇ни во́лѧ є҆гѡ̀, и҆ въ зако́нѣ є҆гѡ̀ поꙋчи́тсѧ де́нь и҆ но́щь. И҆ бꙋ́детъ ꙗ҆́кѡ дре́во насажде́ное при и҆схо́дищихъ во́дъ, є҆́же пло́дъ сво́й да́стъ во вре́мѧ своѐ.

Fericit bărbatul, care n-a umblat în sfatul necredincioșilor și în calea păcătoșilor nu a stat și pe scaunul hulitorilor n-a șezut; ci în legea Domnului e voia lui și la legea Lui va cugeta ziua și noaptea. și va fi ca un pom răsădit lângă izvoarele apelor, care rodul său va da la vremea sa.

13.2 OpenType and SIL Graphite features

The Monomakh font offers a number of optional OpenType features that may be turned on or off by the user, coupled with analagous features in SIL Graphite. These are:

- Stylistic Set 6 (*ss06*) displays U+0456 Cyrillic Small Letter Ukrainian / Belorussian I with one dot above and Stylistic Set 7 (*ss07*) displays the same character with two dots above. By default, U+0456 is displayed with no dots. The Graphite analog is provided by the feature Cyrillic Decimal I (*deci*), which takes values 0 (default), 1 or 2, for the number of dots on U+0456.

- Stylistic Set 8 (*ss08*) displays the characters U+0417 Cyrillic Capital Letter Ze and U+0437 Cyrillic Small Letter Ze as a "sharp zemlya", i.e., like the characters U+A640 Cyrillic Capital Letter Zemlya and U+A641 Cyrillic Small Letter Zemlya, respectively. In Graphite the same functionality is provided by the feature Cyrillic Zemlya (*zeml*), which takes values *Round* (0) and *Sharp* (1). Generally, this change should be handled at the codepoint level, so the use of this feature is discouraged.

- Stylistic Set 9 (*ss09*) displays the characters U+0427 Cyrillic Capital Letter Che and U+0447 Cyrillic Small Letter Che in their archaic form, with the descender in the middle (e.g., ҵ instead of ч). The same functionality is provided by the Graphite feature Cyrillic Cherv (*chrv*), which takes values *Modern (Single Sided)* (0) and *Old (Double Sided)* (1).

- Stylistic Set 10 (*ss10*) displays the characters U+0429 Cyrillic Capital Letter Shcha and U+0449 Cyrillic Small Letter Shcha in their modern form, with the descender on the right (e.g., щ instead of щ). The same functionality is provided by the Graphite feature Cyrillic Shcha (*shch*), which takes values *Modern (Descender Right)* (0) and *Old (Descender Centered)* (1).

- Stylistic Set 11 (*ss11*) displays the characters U+044B Cyrillic Small Letter Yeru and U+A651 Cyrillic Small Letter Yeru with Back Yer with the two glyphs connected (e.g., ы instead of ы). The same functionality is provided by the Graphite feature Cyrillic Yery (*yery*), which takes on values *Without a Connecting Line* (0) and *With a Connecting Line* (1).

- Stylistic Set 13 (*ss13*) displays the character U+0463 Cyrillic Small Letter Yat with the left stem extended to the baseline (e.g., as ѣ). The same func-

tionality is provided by the Graphite feature Cyrillic Yat (*cyat*), which takes values *With a Back Beak* (0) and *With an Additional Vertical Stem* (1). Please note that this is not the same as U+A653 Cyrillic Small Letter Iotified Yat.

- The same functionality of these Stylistic Sets is provided in OpenType also by the Stylistic Alternatives (*salt*) feature.
- Previous versions of the font provided Stylistic Set 1 (*ss01*), duplicated as Graphite feature Localized Forms for Romanian (*rold*), which displayed U+015E Latin Capital Letter S with Cedilla, U+0162 Latin Capital Letter T with Cedilla, and their lowercase analogs, as U+0218 Latin Capital Letter S with Comma Below, U+021A Latin Capital Letter T with Comma Below, and their lowercase analogs. However, since the use of U+015E, U+0162 and their lowercase analogs for the encoding of Romanian text is considered erroneous, this feature is deprecated. Users are strongly encouraged to convert their text at the codepoint level to use the correct characters for Romanian orthography. However, for the sake of compatibility with text that has been erroneously encoded, this feature is still available.
- Stylistic Set 15 (*ss15*), which provides combining Cyrillic letters with an automatic *pokrytie* where warranted by Synodal orthography is also deprecated and may be removed. Users should explicitly encode the *pokrytie* as U+0487 Combining Cyrillic Pokrytie. See UTN 41: Church Slavonic Typography in Unicode for more information.

Two additional features are available in SIL Graphite only:

- The Graphite feature Convert Arabic Digits to Church Slavonic (*cnum*), when turned on, will automatically display Western Digits ("Arabic numerals") as Cyrillic numerals. This is helpful, for example, for page numbering in software that does not support Cyrillic numerals.
- The Graphite feature Convert HIP-6B Keystrokes to Church Slavonic Characters (*hipb*), when turned on, will display text encoded in the legacy HIP codepage as Church Slavonic. The use of this feature is discouraged and users are encouraged instead to convert HIP-encoded text to Unicode.

14 Indiction Unicode

The Indiction Unicode font reproduces the decorative style of drop caps used in Synodal Slavonic editions since the late 1800's.

The original Indyction font was developed by Vladislav V. Dorosh and was distributed as Indyction UCS as part of CSLTeX, licensed under the LaTeX Project Public License. The font was reencoded for Unicode and edited by Aleksandr Andreev, and is now distributed as Indiction Unicode under the SIL Open Font License. It is intended for use with *bukvitsi* (drop caps) in modern Church Slavonic editions. The character shapes are demonstrated in Table 10.

14.1 Sample Texts

Сл҃же́нъ мꙋ́жъ, и҆́же не и҆́де на совѣ́тъ нечести́выхъ, и҆ на пꙋтѝ грѣ́шныхъ не ста̀, и҆ на сѣда́лищи гꙋби́телей не сѣ́де: но въ зако́нѣ гдⷭ҇ни во́лѧ є҆гѡ̀, и҆ въ зако́нѣ є҆гѡ̀ поꙋчи́тсѧ де́нь и҆ но́щь. И҆ бꙋ́детъ ꙗ҆́кѡ дре́во насажде́ное при и҆схо́дищихъ во́дъ, є҆́же пло́дъ сво́й да́стъ во вре́мѧ своѐ, и҆ ли́стъ є҆гѡ̀ не ѿпаде́тъ: и҆ всѧ̑, є҆ли̑ка а҆́ще твори́тъ, ᲂу҆спѣ́етъ.

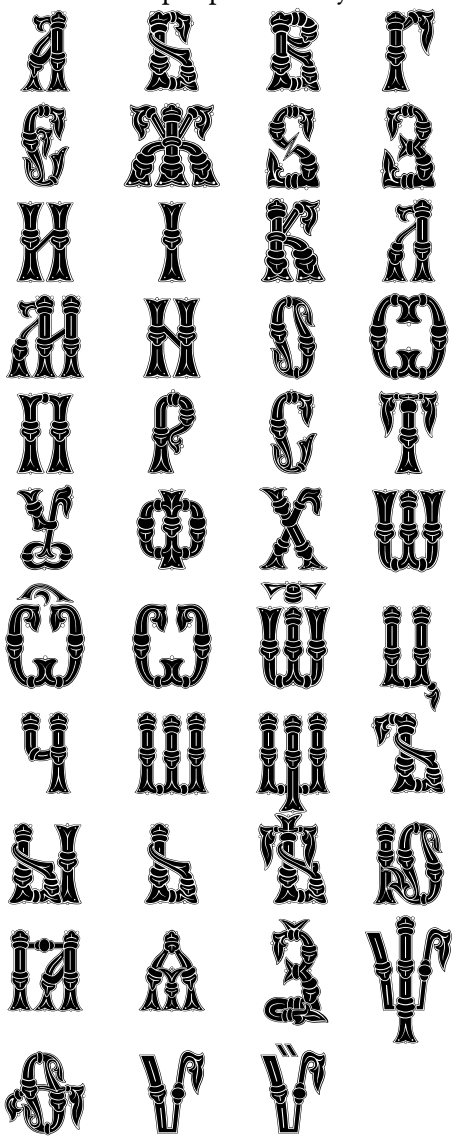
15 Known Issues

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 XeTeX or LuaTeX.

Table 10: Character shapes provided by Indiction Unicode



16 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, allowed his Monomachus font to be modified and repackaged, and whose extensive documentation for the Old Standard font was consulted and partially reused.
- Mike Kroutikov, who put together the TEX package of the fonts.