

Church Slavonic Fonts

Aleksandr Andreev Yuri Shardt Nikita Simmons*

April 17, 2017 version 1.2

(PDF file generated on January 5, 2018)

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	11
8.2	Graphite in X _Y T _E X	12
9	Support of Advanced Features in LibreOffice	12
10	Ponomar Unicode	13
10.1	Synodal Church Slavonic	13
10.2	Kievan Church Slavonic	14
10.3	Other Languages	14
10.4	Font Features	16
10.5	SIL Graphite Features	18
11	Fedorovsk Unicode	19
11.1	Sample Texts	20
11.1.1	Apostol of Ivan Fedorov	20
11.1.2	Flowery Triodion	20
11.2	OpenType Features	20
11.3	Graphite Features	22
12	Menaion Unicode	22
12.1	Sample Texts	23
12.2	Provided Ligatures	23
13	Pomorsky Unicode	23
13.1	Sample Texts	27
14	Monomakh Unicode	27
14.1	Sample Bilingual Text	29
14.2	OpenType and SIL Graphite features	29
15	Indiction Unicode	31
15.1	Sample Texts	33
16	Known Issues	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 [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:

- OpenOffice.org and older versions of LibreOffice had no mechanism to turn off and on advanced OpenType features, so if you needed to use optional typographic features in these applications, you had to use [SIL Graphite](#), which is only available in the TTF version. This limitation has been fixed as of LibreOffice 5.3, which now has [full OpenType support](#).
- OpenType-CFF fonts were poorly supported in Java, and so we had recommended use of the TTF versions in Java programming situations. This problem appears to have been resolved in Oracle Java SE 7.
- In Microsoft products, OpenType glyph positioning is not supported for glyphs in the Private Use Area or characters outside of the Unicode 7.0 range. You should use LibreOffice if you need positioning of combining Glagolitic characters.

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 $X_{\text{Y}}\TeX$ or $\text{Lua}\TeX$.

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 $X_{\text{E}}\TeX$ or $\text{Lua}\TeX$.

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_YTeX 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. LibreOffice also supports OpenType features starting with version 4.1, and support for turning off and on optional features was added in version 5.3.

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 some applications lack an interface to turn on and off optional features. Generally speaking, you will get best results in X_YTeX or LuaTeX using the fontspec package. LibreOffice also supports OpenType features starting with version 4.1, and support for turning off and on optional features was added in version 5.3. Please see the section [Support of Advanced Features in LibreOffice](#), 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:

$$\begin{aligned} \text{A} + \acute{\text{}} &\rightarrow \acute{\text{A}} \\ \text{A} + \acute{\text{}} + \acute{\text{}} &\rightarrow \acute{\acute{\text{A}}} \quad (\text{glyph positioning via } \textit{mark} \text{ feature}) \\ \text{A} + \acute{\text{}} + \hat{\text{}} &\rightarrow \hat{\acute{\text{A}}} \quad (\text{glyph positioning via } \textit{mkmk} \text{ feature}) \end{aligned}$$

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 X₂L^AT_EX).

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:

$$\begin{aligned} \text{A} &\rightarrow \text{A} \quad (\text{glyph substitution using } \textit{ccmp} \text{ feature}) \\ \text{A} + \acute{\text{}} &\rightarrow \acute{\text{A}} \quad (\text{ligature substitution using } \textit{ccmp} \text{ feature}) \\ \text{A} + \text{W} + \text{K} &\rightarrow \text{A} \quad (\text{ligature substitution using } \textit{ccmp} \text{ feature}) \\ \text{A} + \acute{\text{}} &\rightarrow \text{A} \quad (\text{contextual substitution using } \textit{ccmp} \text{ feature}) \end{aligned}$$

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:

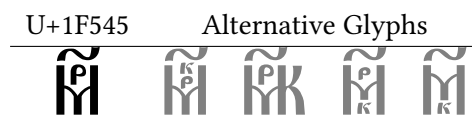
- ı (Church Slavonic text)
- і (Ukrainian text)

To make use of these features, you need an OpenType-aware application that supports specifying the language of text, for example X_YTeX 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:



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

¹Please 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 X_YTeX or LuaTeX using the `fontspec` package and in LibreOffice (starting with version 5.3) by use of a special syntax that appends the needed option to the font name. See the section [Support of Advanced Features in LibreOffice](#), below. In Microsoft Office 2010 and later, Stylistic Sets may be turned off and on under OpenType features on the Advanced tab of the Font dialog. However, Microsoft Office does not allow you to select multiple Stylistic Sets simultaneously or to access the `salt` feature. If necessary, you may access alternate glyphs by codepoint from the Private Use Area (PUA). However, relying on the PUA as a data exchange mechanism is discouraged.

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 (on GNU/Linux and Windows, starting with OpenOffice.org version 3.2; on OS X, starting with LibreOffice version 5.3), Mozilla Firefox (starting with version 11), and X_YTeX (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.

8.1 Graphite in LibreOffice

See the section [Support of Advanced Features in LibreOffice](#), below.

8.2 Graphite in X_YTeX

Graphite support is available in X_YTeX, 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_YTeX 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 Support of Advanced Features in LibreOffice

Support for OpenType features is available in LibreOffice and all OpenOffice.org derivatives starting with version 3.2 of OpenOffice.org. In addition, LibreOffice (as well as other OpenOffice.org derivatives) automatically recognizes fonts which contain Graphite tables. For such fonts, Graphite rendering is enabled by default and the OpenType layout tables will be ignored. OpenType and SIL Graphite cannot be used simultaneously in LibreOffice, so if you do not want SIL Graphite layout, you should only use the OpenType-CFF versions of the fonts.

While correct positioning, attachment and substitutions will work automatically both in the OpenType and SIL Graphite fonts, earlier versions of LibreOffice had no mechanism to turn off and on optional features. Support for turning off and on Graphite features is available starting with LibreOffice version 4.1. However, there is no graphical interface that can be used to turn these 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 enable the mark SIL Graphite feature and display 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.

Beginning with LibreOffice version 5.3, the same syntax may be used also to turn off and on optional OpenType features, such as Stylistic Alternatives (salt) and Stylistic Sets (ss01, ...) in the OpenType-CFF versions of the fonts. For example, to turn on Stylistic Set 1 (ss01) in the font Ponomar Unicode, change the font name to the following:

```
Ponomar Unicode:ss01=1
```

This feature is not available in Apache OpenOffice; since Apache OpenOffice is not well maintained, we suggest users migrate to LibreOffice.

This functionality will be useful for LibreOffice users relying on automatic hyphenation. Since LibreOffice has [no mechanism to set the hyphenation character](#), the Ponomar Unicode and Monomakh Unicode fonts provide the underscore as a hyphenation character via Stylistic Set 1 in OpenType and via the hypf feature in SIL Graphite.

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. Unfortunately, there is presently no graphical interface to support turning off and on OpenType and SIL Graphite features.

10 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.

10.1 Synodal Church Slavonic

Бл҃жѣнѣ мѡжѣ, ѣже не ѣде на совѣтѣ нечестѣвѣхѣ, ѡ на
пѡтѣ грѣшнѣхѣ не стѣ, ѡ на сѣдѣлицѣ гѡбѣителѣхѣ не сѣде:

но въ законѣ гдѣни вола ѡгвѣ, ѡ въ законѣ ѡгвѣ под-
 чѣтца дѣнь ѡ нощь. ѡ вѣдетъ ѡкв дрѣво насаждѣное при
 ѡсходницнхъ вола, ѡже плодъ своѡ дѣтъ во врѣма своѡ,
 ѡ листъ ѡгвѣ не ѡпадѣтъ: ѡ всѡ, ѡлика ѡце творѣтъ,
 ѡспѣетъ. Не тѡкв нечестѣвѣн, не тѡкв: но ѡкв прѡхъ,
 ѡгѡже возметѣтъ вѣтръ ѡ лица земли. Сегѡ рѡди не
 воскреснѣтъ нечестѣвѣн на сѡдъ, ниже грѣшницы въ со-
 вѣтъ прѣвнхъ. ѡкв вѣстъ гдѣ пѣтъ прѣвнхъ, ѡ пѣтъ
 нечестѣвхъ погѣнетъ.

10.2 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 (ѣ):

Блѣженъ мѡжъ, ѡже не ѡде на совѣтъ нечестѣвхъ, ѡ
 на пѣтѣ грѣшнхъ не стѡ, ѡ на сѣдѡлицѣ гвѣтѣлей не
 сѣде: но въ законѣ гдѣни вола ѡгвѣ, ѡ въ законѣ ѡгвѣ
 подчѣтца дѣнь ѡ нощь. ѡ вѣдетъ ѡкв дрѣво насаждѣное
 при ѡсходницнхъ вола, ѡже плодъ своѡ дѣтъ во врѣма
 своѡ, ѡ листъ ѡгвѣ не ѡпадѣтъ: ѡ всѡ, ѡлика ѡце твѡ-
 рѣтъ, ѡспѣетъ. Не тѡкв нечестѣвѣн, не тѡкв: но ѡкв
 прѡхъ, ѡгѡже возметѣтъ вѣтръ ѡ лица земли. Сегѡ
 рѡди не воскреснѣтъ нечестѣвѣн на сѡдъ, ниже грѣшницы
 въ совѣтъ прѣвнхъ. ѡкв вѣстъ гдѣ пѣтъ прѣвнхъ, ѡ
 пѣтъ нечестѣвхъ погѣнетъ.

10.3 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:

ТѦТЪЛЪ НѦСТРЪ КѦРЕЛЕ ЁЩИ ѿ ЧЕРЮРЪ: СФѦЦѦКЪСЕ НЪ-
МЕЛЕ ТЪЪ: ВІЕ ѿПЪРЪЦІА ТѦ: ФІЕ ВѦА ТѦ, ПРЕЪМЪ ѿ ЧЕРЮ
ШИ ПРЕ ПЪМЪНТЬ. ПЪНѦ НѦСТРЪ ЧѦ ДЕРЪРѦ ДЪНЕ НѦА
ЪСТЪЪ. ШИ НЕ ЁРТЪ НѦА ГРЕШАЛЕЛЕ НѦСТРЕ, ПРЕЪМЪ ШИ
НОИ ЁРТЪМЪ ГРЕШИЦАВЪРЪ НѦЦРИ. ШИ НЪНЕ ДЪЧЕ ПРЕ НѦИ
ѿ НСПИТЪ. ЧИ НЕ НЪВЪЦЕ ДЕ ЧЕЛЪ РЪЪ.

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

ТЪМАНІНЪ ЁДАКЪ! ЁМАНЪ АКЪХЪТЪХІНЪ ІНІНЪ КЪГІНЪ,
ЁСАНЪ АМЪГІСАДАГЪТА, ЁСАЛІНЪ АКАГЪТА, АНЪХЪТАНАТЪХІНЪ
МАЛГАГЪТАНЪ ІНІМЪ КЪГАНЪ КАЮХЪ ТАНАМЪ КЪГАНЪ. ВАЛ-
ГАДАМЪ АНЪХЪТАНА ІНІНЪ АКАЧА ЁААМЪ: КАЮХЪ ТЪМАНІНЪ
АДЪНЪ ІНІНЪ ІГНІДА, АМАКЪНЪ ТЪМАНЪ КАЮХЪ МАЛГАЛІГІНЪ
ІНІНЪ АДЪГІНАНЪ ІГНІДАКЪНЪ: КАЮХЪ ТЪМАНЪ СЪГЛАТАЧЪНІГА-
НАХЪТЪХІНЪ, ТАГА АДАЮДАМЪ ІЛАНЪ ТЪМАНЪ АГІНА.

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

Халланнаръ юрдюлоригаръ баръ агыбытъ бисена! СвѦ-
тейданинъ атынъ эена; каллинъ царстванъ эена; сиръ
юрдюгаръ кѦнгюлюгъ эена, халланъ юрдюгаръ курдүкъ бол-
лунъ; бюгюгю кюннѦги асыръ аспытынъ бисенинъ күлү
бисеѦа бюгюгю; бисиги да естѦрѦвѦтинъ халларъ бисеѦа, хай-
таѦхъ бисиги да халларѦбытъ бѦэвѦтѦ естѦхътѦрѦвѦтигаръ;

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

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 for Ukrainian text stylized in a Church Slavonic font).

Table 2: Stylistic Alternatives in Ponomar Unicode

	Base Form	Alternate Forms
U+1F545	Ѡ	Ѡ̅, Ѡ̅̅, Ѡ̅̅̅, Ѡ̅̅̅̅
		Ѡ̅̅̅̅̅, Ѡ̅̅̅̅̅̅, Ѡ̅̅̅̅̅̅̅, Ѡ̅̅̅̅̅̅̅̅
U+0423	Ѳ	Ѳ̅
U+040E	Ѳ̅	Ѳ̅̅
U+0404	Ѣ	Ѣ̅

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.

Stylistic Set 1 (ss01) is provided as a temporary workaround to [LibreOffice Bug 85731](#), which does not allow you to specify the hyphenation character in LibreOffice. When turned on, it replaces all instances of U+002D Hyphen-Minus and U+2010 Hyphen with U+005F Low Line (underscore) for use as a hyphenation character. Please note that this feature will be deprecated once the neces-

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

	Base Form	Truncated Forms		
U+0440	Ѳ	Ѳ		
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	ѹ	ѹ	ѹ	
U+A641	Ѻ	Ѻ		
U+A64B	ѻ	ѻ	ѻ	ѻ

sary functionality is added to LibreOffice.

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. Ѡ ѡѣ по словѣнскѣ.
 Blackletter **The quick brown fox. 1234567890. Ѡ ѡѣ 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.

10.5 SIL Graphite Features

The SIL Graphite features in the font provide the same functionality as the OpenType features. 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.

The following additional Graphite features are provided (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

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			

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”) provides the same functionality as Stylistic Set 1. It replaces all instances of U+002D Hyphen-Minus and U+2010 Hyphen with U+005F Low Line (underscore) for use as a hyphenation character. Possible values are 0 (no) and 1 (yes). Please note that this feature will be deprecated once the necessary functionality is added to LibreOffice.

11 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.

11.1 Sample Texts

11.1.1 Apostol of Ivan Fedorov

Первое оубо слово сотвори́хъ о всѣхъ , ѿ , дреѡфиле , о ни́же
начѣтъ ꙗко , творѣни́же и оуби́ти . до него́же днѣ , заповѣдавъ
аплѡмъ дхѡмъ стѣи , и́хже и́збра̀ вознесѣса . прѣни́мниже и́ поствѣи себѣ
жѣва по страданіи своѣмъ . во мно́зехъ и́стинныхъ знаменіи́хъ .
днѣми четы́рнадцатми іавлѣиася и́мъ и́глѣ іа́же оцрѣвѣи бжѣи .
снѣмниже и́ іады̀ , повелѣвѣаше и́мъ ѡ іеросали́ма не ѡлѡчѣтиса . но
ждѣти обѣтовѣніе ѡчѣе , ѣже слышасте ѡ менѣ . іако ѡ іѡаннѣ оубо
во крѣпѣи́хъ е́сть водою . вы́же и́мате крестѣи́тиса дхѡмъ стѣимъ ,
не по мно́зѣхъ сѣхъ днѣ .

11.1.2 Flowery Triodion

сѣры пѣщѣ . глѣзъ , ѿ . **Да воскреснетъ вѣтъ , * и разы́дѣтсѣ вразнѣ**
ѣгѡ . Пѣсѣха сѣи́ннаѣ на́мъ днѣсь показѣса , пѣсѣха но́ва сѣла ,
пѣсѣха члѣнствѣннаѣ , пѣсѣха всечестнаѣ , пѣсѣха хрѣта́ и́збѣвѣи́телеѣ ,
пѣсѣха непорѡчнаѣ , пѣсѣха вели́каѣ , пѣсѣха вѣрны́мъ , пѣсѣха двѣри
ра́йскѣѣ на́мъ ѡверзѣи́цаѣ , пѣсѣха всѣхъ ѡсѣи́и́цаѣ вѣрныхъ .

11.2 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.

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.

Table 5: Ligatures available in Fedorovsk Unicode

Name	Sequence	Appearance
Ligature A-U	U+0430 U+200D U+0443	АУ
Ligature El-U	U+043B U+200D U+0443	АУ
Ligature A-Izhitsa	U+0430 U+200D U+0475	АУ
Ligature El-Izhitsa	U+043B U+200D U+0475	АУ
Ligature Te-Ve	U+0442 U+200D U+0432	ѢѦ
Ligature Er-Yat	U+0440 U+200D U+0463	ѢѦ

Table 6: Stylistic Alternatives in Fedorovsk Unicode

	Base Form	Alternate Forms
U+0404	Ѧ	Ѧ
U+0426	Ѧ	Ѧ
U+0491	Ѧ	Ѧ
U+A64C	Ѧ	Ѧ
U+047C	Ѧ	Ѧ
U+047E	Ѧ	Ѧ
U+047F	Ѧ	Ѧ
U+1F545	Ѧ	Ѧ Ѧ Ѧ Ѧ Ѧ Ѧ Ѧ Ѧ
U+0463 U+0486	Ѧ	Ѧ
U+0463 U+0300	Ѧ	Ѧ Ѧ
U+0463 U+0301	Ѧ	Ѧ Ѧ
U+0463 U+0311	Ѧ	Ѧ Ѧ
U+0463 U+0486 U+0301	Ѧ	Ѧ
U+A64B U+0486	Ѧ	Ѧ
U+A64B U+0300	Ѧ	Ѧ Ѧ
U+A64B U+0301	Ѧ	Ѧ Ѧ
U+A64B U+0311	Ѧ	Ѧ Ѧ Ѧ
U+A64B U+0486 U+0301	Ѧ	Ѧ

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:

Χ̂ρ̂ι̂σ̂τ̂ο̂ς̂ β̂ο̂σ̂κ̂ι̂ρ̂σ̂ε̂ ἡ̂ς̂ μ̂ε̂ρ̂τ̂β̂υ̂χ̂ς̂ (regular text)
Χ̂ρ̂ι̂σ̂τ̂ο̂ς̂ β̂ο̂σ̂κ̂ι̂ρ̂σ̂ε̂ ἡ̂ς̂ μ̂ε̂ρ̂τ̂β̂υ̂χ̂ς̂ (Stylistic Set 10 enabled)

11.3 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.

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.

12 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.

Table 7: Alternatives via Graphite Features in Fedorovsk Unicode

	Base form	Alternative 1	Alternative 2	Alternative 3
U+0404	Ѧ	Ѧ		
U+0426	Ѧ	Ѧ		
U+0491	Ѧ	Ѧ		
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		

12.1 Sample Texts

Samples of text in Menaion Unicode are presented in Figures 1 and 2. Please note that combining Glagolitic letters (Glagolitic Supplement) became available in Unicode 9.0. In older versions of Microsoft software, correct glyph positioning for these characters using OpenType features may not be possible. To achieve the desired output, we recommend you use LibreOffice, X₂TeX, LuaTeX, or advanced desktop publishing software such as Adobe InDesign.

12.2 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.

13 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,

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 ꙗ҃҃ СЕ БѢ
5 НѢКОНН ОУ
6 БѢ ∷ Н ТѢМЬ ВСА БЫ
7 ША ꙗ҃҃ Н БЕЗ НЕГО НН
8 УЪТОЖЕ НЕ БЫСТЬ ·
9 КЖЕ БЫСТЬ ꙗ҃҃ ВЪ ТО
10 МЬ ЖНВОТЪ БѢ · Н
1 ЖНВОТЪ БѢ СВѢТЪ
2 УЛОВѢКОМЪ ꙗ҃҃ Н СВѢ
3 ТЪ ВЪ ТѢМѢ СВѢТН
4 ТЪСА · Н ТѢМА КГО
5 НЕ ОБАТЪ ꙗ҃҃ БЫСТЬ
6 УЛѢКЪ ПОСЪЛАНЪ
7 ОТЪ БѢ · НМА КМОУ
8 НОАНЪ ꙗ҃҃ ТЪ ПРНДЕ
9 ВЪ СВѢДѢТЕЛЬ
10 СТВО · ДА СВѢДѢТЕ
2.2 1 ЛСТВОУКЪТЪ О СВѢ
2 ТѢ ꙗ҃҃ ДА ВЪСН ВЕРЖ
3 НМЖТЪ НМЬ ∷ НЕ БѢ
4 ТЪ СВѢТЪ ∷ НЪ ДА
5 СВѢДѢТЕЛЬСТВОУ
6 КЪТЪ О СВѢТѢ ꙗ҃҃ БѢ
7 СВѢТЪ НСТННЪНЪ
8 Н · НЖЕ ПРОСВѢЩАН
9 ТЪ ВСАКОГО УЛКА ,
10 ГРАДЖЩА ВЪ МНРЪ ꙗ҃҃

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

1 ✠ Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 2 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 3 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 4 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 5 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 6 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 7 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 8 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 9 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 10 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 11 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 12 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 13 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 14 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 15 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 16 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 17 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 18 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 19 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 20 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇
 21 Ǫ̇ Ǫ̇ Ǫ̇ Ǫ̇

Table 8: Ligatures available in the Menaion Unicode font

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

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 ornamentation of another); these simple forms are available as Stylistic Set 1 or as the Graphite feature “Use simple forms” (`sml`). There are a few additional characters that are stylistic variants, which are provided as Stylistic Alternatives (`sal`) or as the Graphite feature “Alternates” (`sal`). 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.

13.1 Sample Texts



14 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

Table 9: Character shapes provided by Pomorsky Unicode

AAA	BBB	SSS	TTT
EEE	KKK	SSS	ZZZ
HHH	HHH	JJJ	III
KKKSSS	AAA	AAA	HHH
SSS	OOO	HHH	RRR
SSS	TTT	SSS	YYY
PPP	XXX	OWW	OWW
OWW	UUU	YYY	UUU
UUU	KKK	KKK	KKK
KKK	SSS	AAA	AAA
ZZZ	VVV	SSS	VVV

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.

14.1 Sample Bilingual Text

Бл҃жєнѣ мѡжѣ, ѡже не ѡде на
совѣтѣ нечестѣвыхѣ, ѡ на пѡтѣ
грѣшныхѣ не стѡ, ѡ на сѣдѡлицѣ
гѡвѣтелей не сѣде: но въ законѣ
г҃днѣ вола ѡг҃у, ѡ въ законѣ ѡг҃у
пѡдѡйтса дѣнь ѡ нѡчь. ѡ бѡдетѣ
ѡкѡ дрѣво насаждѣное при ѡсѡди-
цихѣ водѣ, ѡже плѡдѣ свѡѡ дѡстѣ
во врѣмѡ свѡѡ.

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

14.2 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 analogous features in SIL Graphite. These are:

- Stylistic Set 1 (*ss01*) is provided as a temporary workaround to [LibreOffice Bug 85731](#), which does not allow you to specify the hyphenation character in LibreOffice. When turned on, it replaces all instances of U+002D Hyphen-Minus and U+2010 Hyphen with U+005F Low Line (underscore) for use as a hyphenation character. The Graphite analog is provided by the feature “Use underscore for hyphenation” feature (*hyph*), which takes values 0 (no) and 1 (yes). Please note that this feature will be deprecated once the necessary functionality is added to LibreOffice.
- 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 (*zempl*), which takes values *Round* (0) and *Sharp* (1). Generally, this change should be handled at the code-point 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., **ѵ** 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., **ш** 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., **ѣ** instead of **ѣ**). 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 functionality 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:

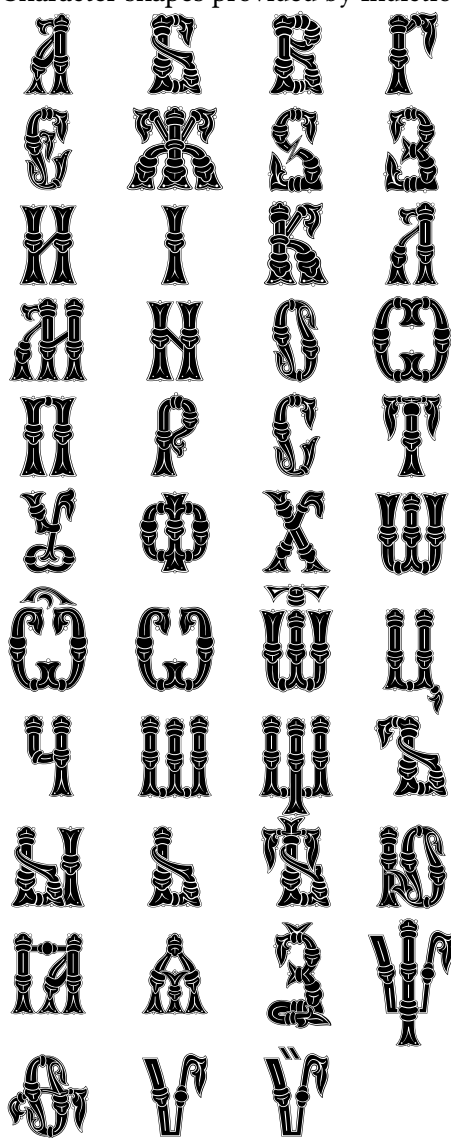
- The Graphite feature Convert Arabic Digits to Church Slavonic (*cnum*), when turned on, will automatically display Western Digits (“Arabic numerals”) as Cyrillic numerals. This is helpful, for example, for page numbering in software that does not support Cyrillic numerals.
- The Graphite feature Convert 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.

15 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 Indiction font was developed by Vladislav V. Dorosh and was distributed as Indiction UCS as part of CSLTeX, licensed under the L^AT_EX 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.

Table 10: Character shapes provided by Indiction Unicode



15.1 Sample Texts

 ѿжѣнѣ мѡжѣ, ѿже не ѿде на совѣтѣ нечестѣвыхѣ,
и на пѡтѣ грѣшныхѣ не стѣ, и на сѣдѣлицѣ гдѣ-
бѣителѣи не сѣде: но въ законѣ гдѣни вѡлѣ ѣгѡ, и
въ законѣ ѣгѡ подчѣтѣмъ дѣнь и нѡщѣ. И вѣдетѣ ѿкѡ
дрѣво насажденое при исхѡдѣиныхѣ вѡдѣ, ѣже плѡдѣ свѡй
дѣстѣ во врѣмѣ своѣ, и листѣ ѣгѡ не ѡпадѣтѣ: и вѣлѣ,
ѣлика ѿце творѣтѣ, оѡспѣетѣ.

16 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_{La}TeX or Lua_{TeX}.

17 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.